INVESTIGATIONS INTO CONTROLLERS FOR ADAPTIVE AUTONOMOUS AGENTS BASED ON ARTIFICIAL NEURAL NETWORKS

R. Mark Rylatt, B.Sc. (Hons.), M.Sc.

Submitted in partial fulfilment of

the requirements for the degree of

DOCTOR OF PHILOSOPHY

De Montfort University

June, 2001

ABSTRACT

R. MARK RYLATT: INVESTIGATIONS INTO CONTROLLERS FOR ADAPTIVE AUTONOMOUS AGENTS BASED ON ARTIFICIAL NEURAL NETWORKS (2001).

This thesis reports the development and study of novel architectures for the simulation of adaptive behaviour based on artificial neural networks. There are two distinct levels of enquiry. At the primary level, the initial aim was to design and implement a unified architecture integrating sensorimotor learning and overall control. This was intended to overcome shortcomings of typical behaviour-based approaches in reactive control settings. It was achieved in two stages. Initially, feedforward neural networks were used at the sensorimotor level of a modular architecture and overall control was provided by an algorithm. The algorithm was then replaced by a recurrent neural network. For training, a form of reinforcement learning was used. This posed an intriguing composite of the well-known action selection and credit assignment problems. The solution was demonstrated in two sets of simulation studies involving variants of each architecture. These studies also showed: firstly that the expected advantages over the standard behaviour-based approach were realised, and secondly that the new integrated architecture preserved these advantages, with the added value of a unified control approach. The secondary level of enquiry addressed the more foundational question of whether the choice of processing mechanism is critical if the simulation of adaptive behaviour is to progress much beyond the reactive stage in more than a trivial sense. It proceeded by way of a critique of the standard behaviour-based approach to make a positive assessment of the potential for recurrent neural networks to fill such a role. The findings were used to inform further investigations at the primary level of enquiry. These were based on a framework for the simulation of delayed response learning using supervised learning techniques. A further new architecture, based on a second-order recurrent neural network, was designed for this set of studies. It was then compared with existing architectures. Some interesting results are presented to indicate the appropriateness of the design and the potential of the approach, though limitations in the long run are not discounted.

LIST OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

PAGE
MISSING
IN
ORIGINAL

AUTHOR DECLARATIONS

During the period of registered study in which this thesis was prepared, the author has not been registered for any other academic award or qualification.

The material included in this thesis has not been submitted wholly or in part for any academic award or qualification other than for that for which it is now submitted.

R. Mark Rylatt
March, 2001.

ACKNOWLEDGEMENTS

Thanks are due to my second supervisor Dr. Chris Czarnecki for giving me the chance to undertake these studies, for his continued faith throughout some difficult times and for his constant encouragement to publish my work; to Dr.Tom Routen for awakening my interest in the more philosophical implications of my work before his departure; to Professor Paul Luker for valuable supervisory meetings, general encouragement and understanding of personal problems; and to my wife Yvonne who despite serious health problems has continued to give me her full support; and to my thirteen year old son Matthew for his patience and offers of help!

CHAPTER 1

OUTLINE OF THE THESIS

The structure of this thesis reflects both the nature of the research field to which it belongs and the two distinct levels of enquiry in the doctoral studies on which it is based. As to the first of these observations, the simulation of adaptive behaviour (SAB, see section 2.3) is a relatively young and multidisciplinary field of research. It is therefore not surprising to encounter differing positions on foundational issues as well as disagreements about practical aims and approaches. Accordingly, it is the aim in the earlier chapters to prepare the reader with critically assessed background material and technical information on issues and mechanisms. The second observation concerns the contribution to knowledge offered by the thesis, mainly contained in the later chapters. On one level – and this may be considered the primary level - the thesis reports the development and study of control architectures for the simulation of adaptive behaviour. At another level, a position is established on a foundational issue associated with the chosen mechanisms. This constitutes an essential link between two clearly distinguishable phases at the primary level, a relationship reflected in the order of presentation. A summary of the chapter contents will make all this clear.

Chapter 2 begins with a brief treatment of foundational problems in traditional AI and a critical examination of one of the solutions offered by behaviour-based robotics. Following this, the argument is begun that an approach based on artificial neural networks (ANNs) can at least begin to address some of these problems. The general importance of learning in simulating adaptive behaviour is discussed in order to

suggest why neural network-based approaches may have advantages over behaviour-based control architectures; different broad approaches and standpoints are identified and critiqued. Chapter 3 contains some background to the mechanisms chosen as the substrate for this investigation and a brief outline of the general ANN class of interest is given there. This is followed by a more detailed discussion of the less well-known subclass of recurrent neural networks that become the focus of the second half of the thesis. A review of research work concerning ANN control architectures and approaches is the subject of Chapter 4. It contains references to, and descriptions of, some quite specific antecedents to this research and highlights research issues for investigation. In Chapter 5 there is a discussion of aspects of the approach used for the studies of simulated adaptive behaviour that follow. In particular, an outline is provided of the integrated mobile robot and neural network simulator developed by the author to support them.

The theme of how an architecture may be developed with a unified approach to learning and control begins properly in Chapter 6. It concerns a novel modular architecture based on ANNs and some architectural ideas from one of the antecedents discussed in Chapter 4. The approach represents a first step, as it combines reinforcement learning at the sensorimotor level with an overall control algorithm designed to address an intriguing composite of the action selection and credit assignment problem. Some studies are described which show that it does not share a well-known shortcoming of the subsumption architecture. Chapter 7 continues the theme of unifying control and learning in a modular architecture. Here attention is turned to finding an ANN-based solution to the problem of action selection and credit assignment at the control level. It contains an account of how a connectionist

architecture intended for static domains, and based on supervised learning, can be adapted to the temporally extended domain of interest using, instead, reinforcement learning. Studies are presented that indicate how the new unified architecture can perform in a similar manner to its predecessor.

Chapter 8 represents a step back from the studies undertaken thus far, taken in order to reflect more deeply on the foundational issues introduced in Chapter 2. It documents a point at which insights gained from the unfolding studies could support a more fundamental critique of the behaviour-based approach and an explanation of the need for an approach permitting seamless temporal processing such as recurrent neural networks. Chapter 9 concerns an attempt to show how these extended temporal aspects can begin to be explored. A new architecture is introduced, together with a framework for studying the phenomenon of delayed response learning. Some interesting results are presented based on the comparison of the new architecture with several other ANN-based approaches.

The tenth chapter contains concise conclusions concerning the significance of the main research findings in relation to the position set forth, together with some reflections on the strengths and weaknesses of the approaches used. Finally, in the Chapter 11 the implications for future research are considered.

CHAPTER 2

SHIFTING TOWARDS A NEW AI

COMPETING PARADIGMS

2.1 Introduction

In a research field that is both inchoate and multidisciplinary, it is necessary to establish a position at the outset. Indeed, it is integral to an explanation of the purpose and scope of the research. In this case, it also serves to justify the choice of mechanism underlying the practical investigations and later provides a platform for foundational deliberations arising from them. Hence, this chapter deals firstly - and necessarily briefly - with relatively well-known, foundational problems in traditional Artificial Intelligence (AI). A more detailed critique of an alternative paradigm known as *behaviour-based AI* follows. The argument is then begun, *per contra,* that an approach based on artificial neural networks (ANNs) can at least begin to address some of the problems at the project level[1]. The general importance of learning for achieving adaptive behaviour is discussed and, in this respect, it is argued that neural network-based approaches have advantages over behaviour-based control architectures. Different broad approaches to learning are identified, particularly in relation to the point at which learning should start, and some alternatives to the one adopted here are critically assessed.

[1] In this thesis the term *project-level* refers to work in the field of AI with mainly practical aspirations as opposed to *program-level* or *programmatic* work that primarily aims to address foundational issues.

2.2 Of gunpowder and giant birds: foundational problems in symbolic AI

The investigations in this thesis are primarily empirical in nature, but their provenance is to be found at the level of fundamental debate – the philosophy of AI - rather than of purely pragmatic concerns. Although these background issues emerge, and are to some extent developed, during later discussions of antecedents and means, a brief summary is provided in this section. Because space does not permit adequate discussion at this level, a metaphor is used in order to place the contribution of this thesis in the wider context.

Recently, and perhaps more clearly than in previous accounts, Franklin (1995) has identified three principal areas of debate in what he sees as an emerging paradigm of mind, describing intelligence primarily in terms of control mechanisms. The first of these debates needs little introduction: it is the ultimate question of whether machines will ever be able to think in the way that humans do, the ultimate goal of programmatic AI as opposed to project-level AI. It would be presumptuous to suggest that the investigations described here will make any substantial contribution to this most fundamental debate. The massive AI research programme to date has failed to alter the extreme polarity of informed opinion on the issue (see, for example, Copeland, 1993 and Winograd and Flores, 1986, pro and contra). By analogy with some ancient ideas about space exploration, we have probably not arrived at the stage

of inventing gunpowder but, maybe we are beginning to reject the notion of flying into space using the wings of giant birds[2].

This rejection of the prevailing symbolic paradigm leads to Franklin's second area of debate, which concerns the choice of an appropriate model of mind[3]. The position in this thesis can be very broadly stated at this point by extending the space exploration metaphor. As feathers may not have properties entirely suited to flight through a vacuum, we must work towards the discovery of rocket propulsion (and hope that we are not sidetracked by ideas such as being launched from a cannon). In artificial neural networks (ANNs), we may have at our disposal at least a potential component of an equivalent to gunpowder. To this debate too, however, little will be added here: Justification for this very moderate stance can be found in respected sources such as (Smolensky, 1988) and the practical benefits of neural networks will be weighed in Chapter 3.

To the third, and most recently joined of Franklin's debates the last phase of the work described in this thesis attempts to contribute. It is the foundational argument about representation. Representation is the reason why traditional AI is probably not going to get into space, the barrier to symbolic wings. After the next section, a discussion is begun of perhaps the best-known and most influential example of a non-connectionist approach that rejects traditional AI-style representation. In Chapter 8, it will be argued more precisely that this approach too can never lead to artificial creatures with human-level intelligence, originally claimed to be its ultimate goal (Brooks, 1991a).

---

[2] The means of getting to the moon suggested by Lucian of Samosata, an early Greek writer, in *Icaro Menippus.*

[3] No attempt is made in this thesis to explore the well known Mind-Body problem: Franklin's view is broadly that mind is process, the activity of the brain.

However, it was an approach giving hope to at least one leading movement in the philosophy of AI (section 8.2). That it may be viewed as "space propulsion's cannon" is explained and justified in the following sections. This is a prerequisite for the work described in the next three chapters, in which the advantages of the alternative ANN substrate as a control medium are investigated mainly empirically.

2.3 The simulation of adaptive behaviour

To facilitate understanding of terms and conventions, a brief introduction to the field is provided in this section. Deciding how to label one's research activity should be a trivial matter but as indicated earlier, the field is still in the process of cohering into a readily identifiable corpus. The *simulation of adaptive behaviour* (SAB) was adopted after some deliberation. The related field of *artificial life* (A-Life) offers too broad an alternative rubric because, as its name suggests, it admits research into the simulation of all forms of life. *Behaviour-based robotics* (sometimes *behaviour-based AI*) on the other hand has connotations that may be too restrictive – although there is no precise definition available, the term seems to be used quite commonly to exclude work based on neural networks. For example, in a recent authoritative survey of the field (Arkin, 1998) contains a chapter on different types of behaviour-based architectures, but none of these is based on ANNS, even though a later chapter does discuss neural network learning.

Recently, the Society for Simulation of Adaptive Behaviour has been formed, organising biennial international conferences referred to (semiofficially) as SABs.

Therefore, SAB has been adopted in this thesis as a convenient acronym representing an umbrella term with the right balance. Other alternatives are either unwieldy or carry connotations that make them inappropriate. For example Maes (1995), refers to *adaptive autonomous agent research, behaviour-based AI* and *animat approach* as synonyms for a new wave of AI that opposes the mainstream, symbolic AI whose deficiencies were outlined in the previous section. Even so, the terms "adaptive autonomous agent", sometimes abbreviated to "agent" are used for convenience in this thesis. According to Maes:

> An *agent* is a system that tries to fulfil a set of goals in a complex, dynamic environment. An agent is situated in the environment: It can sense the environment through its sensors and act upon its environment using its actuators. (Maes, 1995, p. 136)

Maes argues that the general approach is appropriate for the class of problems that require a system autonomously to fulfil multiple goals in a dynamic, unpredictable environment, rather than just robotic forms of intelligence. This means those agents and their sensors, actuators and environments do not have to be physical: The field includes the study of cyberspace agents occupying purely virtual worlds, and computer simulations of physical agents and environments. The commonality of such systems is perhaps to be found in general aims, rather than in any specific shared organisation or architecture. Most significantly, they seek to avoid any human agency in the loop between perception and action. In this, they may most clearly be distinguished from the well-known class of traditional AI systems known as expert

systems[4]. They are further distinguishable from traditional AI systems, including those that aspire to a more general autonomous real-time operation (for example, planner-based robots), by their eschewal of symbolic world modelling. Thus, they hope to avoid many of the representational difficulties implied by that approach. The field is too young to have evolved any real measure of agreement on fundamental issues; indeed this thesis attempts to bridge some of the most worrying explanatory lacunae.

2.4 Behaviour-based control

In this section, the most influential alternative to symbolic AI and conventional connectionism is described and examined. Other behaviour-based approaches are discussed by Arkin (1998), including his own *motor schema* approach, but he concedes that it was the *subsumption* architecture (Brooks, 1987) that "changed the direction of autonomous robotics research". It seems clear that the alternatives have been less influential both as a practical, architectural basis for robotics or as a more general alternative AI approach. For example, although the motor schema architecture has been quite successful at the project level, its acceptance of traditional AI planning as a higher (deliberative) control level underlines its essentially pragmatic nature. Here, some of the project-level problems inherent in the subsumption approach are outlined, so that the relevance of solutions discussed later can be fully appreciated. In Chapter 8, in the light of insights gained during the

[4] Expert systems that function on-line in real-time, for example, in a process control loop are a slightly awkward exception but they can be regarded as a special case of dedicated embedded systems.

development of this thesis, it will be argued that it has programme-level problems at least as worrying as those now apparent in symbolic AI.

2.4.1 Rationale: lessons from evolution

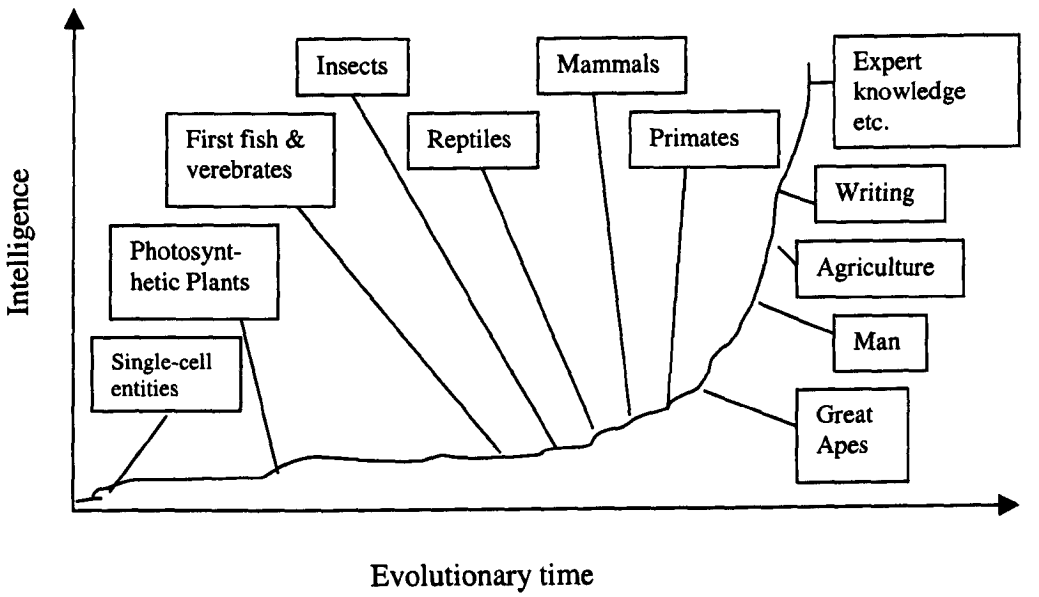

Figure 1: The evolution of "intelligence" based on Brooks' ideas (1991a) in which no particular definition of intelligence was given. This is a chart for illustrative purposes only.

The main rationale underlying the behaviour-based alternative to symbolic AI is summarised graphically in Figure 1, based on ideas expressed by Brooks (1991a). His argument was that, if time required for solving a problem is equated with its level of difficulty, then clearly Evolution found expert knowledge a relatively simple task to

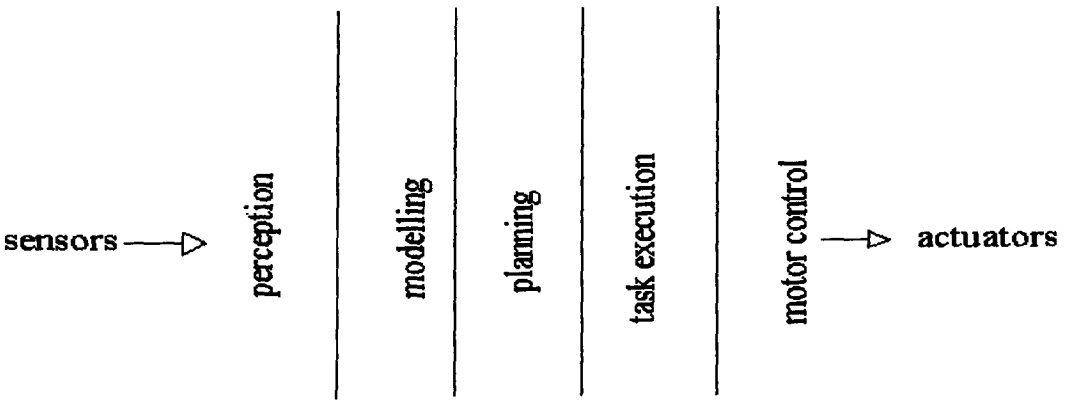

Figure 2: Functional decomposition according to Brooks (1986).

achieve, once the much harder sensorimotor problems had been solved. Accordingly (Brooks concluded), it follows that AI should itself initially address those sensorimotor problems instead of much higher-level abstractions, like planning, that had traditionally been its starting point. The idea is that, if the example of evolution can be followed, once these low-level problems are solved, all the rest should fall readily into place. At first sight, this seems a simplistic notion, and it will be part of this thesis to show that it remains so after deeper reflection. However, firstly, consider it at face value.

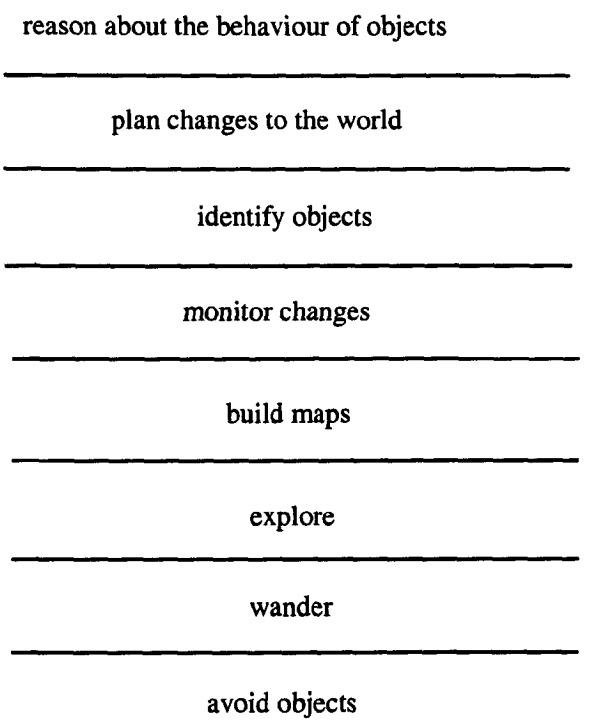

Figure 3: Behavioural decomposition, re-drawn from Brooks (1986).

Brooks (1990) set out the same long-term programme for this *nouvelle AI* as that of its traditional rival, symbolic AI - the achievement of human-like intelligence in an artificial agent. However, the implications of a re-focus on much more primitive abilities as the programmatic starting point are, of course, profound. Firstly, and most obviously, a bottom-up approach is implied. This is in contrast to traditional AI, and Brooks (1991a) made much of this distinction. He originally appears to have referred to the decomposition of problems into *functional* units (Figure 2) to characterise the standard planner-based control architecture for mobile robots (Brooks 1986). However, he later developed this into a critique of symbolic AI in general (Brooks, 1991a). Essentially, it challenged the prevailing paradigm of cognitivism which models cognition / intelligence by reference to information processing concepts.

2.4.2 Critique of cognitivism from the behaviour-based standpoint

The problems stemming from functional decomposition - and ultimately from cognitivism itself - can be viewed on at least two levels. At the lower level, there are practical consequences specifically for robotic forms of AI. In brief, the sequential processing of information, from sensory input through the various distinct stages to motor output, carries computational overheads that have to date seriously impaired the ability of such agents to perform in real time. At this level, the most serious, and (by some accounts) intractable, problem is the need to maintain consistency with the agent's changing perceptions of the world. This has to be done by continually updating the central world model (in the symbolic AI approach, usually a database of predicates). Of course, computational power seems to be ever increasing and it remains an open question whether the cognitivist view will ultimately be justified by this technological advance. Brooks (1991b) however suggests that the paradigm is more fundamentally flawed, implying that increased processing speed is not going to provide the answer. At another level, Brooks (1991a) saw AI researchers in general as subject to a *methodological solipsism*. Based on the introspection of our own human mentality, intelligence has been broken down into components (roughly corresponding to the various distinct information-processing modules required for an artificial intelligence). At the same time, perception and motor skills have mostly been abstracted away. According to this view, assumptions underlying the work on the separate components are not forced to be realistic. Brooks concluded that this lack of realism results in the production of increasingly specialised and abstracted sub-components that will never fit together to form a complete intelligence capable of interacting directly with the world as we do. His alternative way is to force realism at

every step and to develop agents incrementally so that they are competent to interact with the world at each stage of their development.

2.4.3 A methodological alternative

Brooks' alternative approach was *behavioural decomposition* (Figure 3), also referred to by Brooks as *decomposition by activity.* It was originally proposed specifically as a mobile robot control architecture, but was later presented as the methodology for a more general *nouvelle AI* (Brooks, 1990). It is interesting to note that, although proposed as a bottom-up design approach, it is still decompositional, a point that does not appear to have been commented on by its originator. Clearly therefore, it is a bottom-up approach with significant top-down constraints. It will be part of this thesis to argue that the entailments of these constraints lead to a methodological solipsism of a different kind to that suffered by symbolic AI, but one just as fatal to their common programme. For the moment however, judgement will be suspended so that the architecture associated with this design approach can be discussed.

2.4.4 The subsumption architecture

The architecture that emerged from the new design philosophy was the *subsumption architecture* (Brooks, 1986). The key idea is that control layers, corresponding to the levels of competence in the behavioural decomposition, are built as "complete" control systems from the bottom up, the entailment of completeness being that each

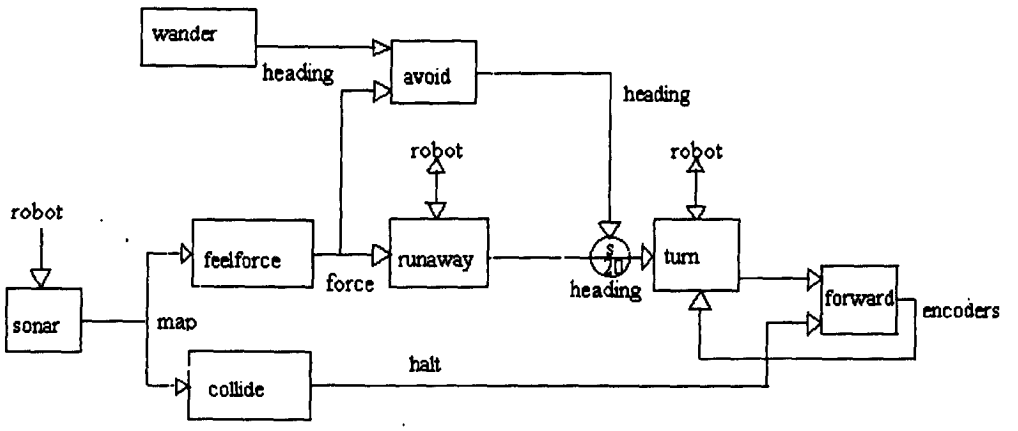

Figure 4: Augmented Finite State Machine (re-drawn from Brooks,1987)

layer has its own sensorimotor control loop though the environment. Each layer is constructed from basic building blocks, described by Brooks as augmented finite state machines (AFSM).

The issue of building blocks, or substrate, is fundamental to the critique in this thesis, but this will be developed later; for now, these AFSMs will be referred to simply as modules. Once the first layer achieves the specified level of competence - after extensive testing and debugging *in the environment for which the whole system was intended* - the second layer is developed and tested on top of the first. Subsequent layers are developed in the same way. The lower layers continue to run concurrently once the next layer has been added; they are, in a sense, unaware of the new layer. The term *subsumption* is appropriate to the design principle, because overall system behaviour results from control layers effectively including the behaviour of lower levels in their own behaviour. The mechanisms used to achieve this effect do not

involve replication of lower-level control structures in the higher levels; instead, they enable modules in the higher control layer to affect the data flow between modules in the lower control layer. The mechanisms are referred to as *inhibition* (effectively an incoming message is blocked for a fixed time-period) and *suppression* (an outgoing message is blocked, and may be replaced by an injection of data from the higher-level module). A subsumption architecture consisting of just two layers is shown in Figure 4. A robot implemented in this way would exhibit the overall behaviour of wandering in a cluttered environment while avoiding obstacles.

A series of robots, with control architectures built on these principles, demonstrated advantages over traditional control architectures and symbolic AI approaches, so that Brooks (1990) could claim: "we have the most reactive real-time robots around". Brooks went on to claim much wider significance for the new approach, and to set out further principles establishing a position essentially in opposition to the prevailing view of cognition as symbol manipulation. A critique of this *nouvelle AI* and the inadequacy of its foundational position on the issue of representation is developed in Chapter 8. However, for now, the project-level weakness of the approach will be the main concern as these were the original focus of the experimental work, and of the novel architectures described later.

2.4.5 Structural and procedural difficulties

In this last subsection of the discussion of behaviour-based control, attention is focused on mainly project-level concerns that have been raised by other researchers. Some of these are addressed either in the work described in Chapters 6 and 7 or in the examples of other ANN approaches surveyed in Chapter 4.

Firstly, it should be recognised that deliberation is no longer primarily a system function in the behaviour-based approach; instead, the designer has the responsibility to wire in the required mappings between conditions and actions. Global system behaviour, including any flexible responses that may be required in the face of unforeseen circumstances and complex behaviour that cannot be precisely engineered, is thus required to emerge from the interaction between layers. In order to ensure rationality, the designer needs to predict these interactions and at each level ensure that the new behaviour will integrate meaningfully with the whole system. Clearly, the demands made of the designer/programmer are very different to those made by more conventional programming tasks, and have implications for the viability of the approach beyond the pure research stage. These will be considered under the ensuing sub-heads.

Handcrafted architectures.

The subsumption architecture is representative of an approach to implementing autonomous agents often referred to as *handcrafted.* This cachet is intended to convey the labour-intensive nature of the process, along with the need for specialised programming skills and even a certain amount of mystique. Brooks himself did not dispel this by referring to the "Zen of robot programming" (Brooks, 1990). From the standpoint of traditional software engineering, this characterisation of the process, of course, is a source of disquiet. To allay such fears, it can be viewed as a transitional phase equivalent to the early programming days when mysterious machine code programs predominated. However, this kind of design and programming approach has other implications that make its role in SAB projects somewhat questionable. To make this objection clear, it may be helpful first to consider the functional

decomposition approach of symbolic AI from a slightly different angle. Brooks' objection was in terms of unjustified (given present knowledge) assumptions on the part of the designer. Another view is that by making the assumptions the designer is also culpable of introducing a strong bias into the way that the agent will interact with the world.

Designer bias

The design and programming process was briefly described in subsection 2.4.5. The important aspect of it for the present argument is that it takes place in "observer space". It is often described as a bottom-up approach but in fact, it is subject to significant, task-oriented top-down constraints. Essentially, a desired behaviour is identified and described qualitatively, for example "boundary-following". It is then broken down into observable discrete actions, in this case, for example, moving towards and away from the boundary, and aligning with the boundary, hence the term behavioural decomposition. The discrete actions are then implemented, essentially as reactive rules. The behaviour actually exhibited by the robot is often said to have "emerged" because there is no top-level rule that actually governs the observed overall behaviour, in this case following a boundary. However, it is probably legitimate to infer that the overall behaviour has been quite precisely engineered in the lengthy testing and debugging cycle that separates each layer of development in the subsumption methodology. Perhaps, at the lowest level of reflex behaviour, this is not a serious worry. However, the amount of "tweaking" that goes on, to ensure performance in a particular environment (as commented on by, for example, Fagg, Lottspeich and Beckey, 1994), surely guarantees severe generalisation and scaling problems. However, extrapolating from this to cognitive behaviour gives cause for

alarm. For example, Brooks' original hierarchy went as far as to specify a behavioural layer to 'reason about objects' (Brooks, 1986). It is pertinent to ask what kind of observer space specification and decomposition (not to mention tweaking) could lead to the implementation of such a layer. At this point, project-level concerns open onto a gulf where foundational demons lurk. An attempt to confront these demons is reported in Chapter 8.

Engineering solutions

It has been explained how the domain ontology of the designer is smuggled into the control mechanism of subsumption-based agents and how this diminishes their appeal to those who wish to establish a new AI predicated on self-adaptivity. A particular practical consequence, observable in such agents, is the prevalence of rigidly task-oriented engineering solutions. A good example of this is the robot Herbert (see, for example, Brooks, 1990), whose purpose is to wander around until it finds a drink can and then collect it. The control loop through the world in this case depends on:

- arm movements being triggered when the robot's wheels stop rotating (i.e. arrived at a can location);
- the robot's direction of motion being decided according to the amount of separation between the fingers on its hand (i.e. "can in hand, so return home", or "no can in hand so continue looking").

While the ingenuity of such solutions is not in question, it is clear that there is little scope for generalisation and the problem of scaling is compounded with issues of complexity. It is legitimate to wonder how many encapsulated, precisely engineered

tasks can be bolted on and persuaded to work together. Indeed, Franklin (1995) comments that Herbert was a "one trick pony" and this may not be wide of the mark.

Action Selection

A related problem is that of deciding between alternative behaviours, that is *action selection*. In the subsumption architecture, multiple behaviours run in parallel (usually simulated) and potentially can be in control at any time. In the absence of any top-level arbiter, some scheme needs to be devised to enable a choice between actions competing to control the actuators on a given step. The typical subsumption solution is a rigid arbitration regime that imposes an order on the actions according to a scale of priorities appropriate in a given context, or a precedence hierarchy. These limitations bear on a number of issues. At this level of description it can be said that the deliberation that in symbolic AI systems takes place at run time (though as has been indicated, rarely in real-time) occurs instead at design time. It has been frozen into the system. Consequently, in situations that were not precisely foreseen by the designer, the robot could be trapped in cyclical behaviour patterns. They "keep activating the same actions even though they have proven not to result in any change of state" (Maes, 1995, p.151). This is a problem specifically addressed in the initial experiments described in this thesis.

2.5 Adaptation through learning

In subsection 2.4.5, it was suggested why some of the problems faced by the behaviour-based approach arise. This was because much of the deliberation, which might be expected at run-time in a truly adaptive autonomous agent, actually occurs in

the design phase, in the mind of the designer. Metaphorically, such agents may adapt in this design space over a kind of evolutionary time span (as described in subsection 2.4.1), but they are mostly incapable of individual learning, or self-adaptation. Undoubtedly, individual hard-wired behaviours can give impressive performance of encapsulated tasks, and the emergence of complex, seemingly intelligent swarm behaviour in which no learning takes place. Even so, it is now perhaps a commonplace to state the importance of learning for the development of individual cognitive abilities. Some attempts to confer learning on these systems have been made, for example, (Maes and Brooks, 1990). Their limited extent seems to reinforce the argument in Chapter 8 that the basic building blocks are at the wrong level of abstraction. Machine learning does not of course require a connectionist substrate. Nevertheless, the kind of learning that takes place in neural networks makes a connectionist framework a much likelier host for a developmental approach to machine intelligence than an arbitrary level of abstraction such as the AFSM. (At this point in the studies, this observation can be considered as a more-or-less intuitive one, but the theme is taken up again and developed much more intensively in Chapter 8). Many examples of specifically connectionist learning will be given in Chapter 4. Here, an overview of broad approaches to learning in the adaptive autonomous agent field is presented in order both to prefigure the discussion of examples and to set the approach developed here in context.

2.5.1 What to learn

The issue of what agents should learn is closely bound with the issue of representation, and hence with questions of grounding and substrate viability (see Chapter 8). This section attempts to justify in a more pragmatic way why the

particular approach to learning followed in the practical work in this research was chosen initially, reflecting again the chronological development of insights as the investigations proceeded.

Therefore, the important question is: what should the agent or robot learn, or, putting it another way, where should learning start? From a practical perspective, it would seem that there is little to be gained by having an agent learn everything from a *tabula rasa*. If a set of competences or behaviours has been fully investigated and implemented, then learning should begin at a higher level, and its purpose should be to co-ordinate such multiple abilities already hard-wired (or pre-coded) at the lower level. Additionally, it has been found that learning complex behaviours from a *tabula rasa* is a hard problem. Therefore, learning this kind of control, sometimes termed *control composition learning,* is an alternative way for an agent to solve complex problems. However, it too smuggles in the designer's domain ontology. For example, the designer may comfortably assemble a range of controllers, each with a label for its actions, such as "aggression", "love" etc. Together, these may lead to a control solution for a problem requiring the interaction of these controls (see Araujo and Grupen, 1996). But it is not clear that a true bottom-up agent would analyse the problem in the way it has been carved up by the designer, or that the higher level solution is likely to be superior. In some ways, the approach is a "quick fix", because, again, it is notoriously hard to get bottom-up agents to learn complex behaviour.

Even if it is accepted that certain low-level competences are now sufficiently understood – so that further baseline research is unnecessary (for example, Lemon and Nehmzow, 1998) - the important issue of granularity may have consequences for

the long-term developmental perspective. This relates to how representations are learned and structured, and how they may ultimately be manipulated by deliberative and reflective agents: in other words the *grounding problem* (see for example, Harnad, 1990). The surprising abundance, even ubiquity, of spatial metaphor in language (Lakoff and Johnson, 1980) is perhaps evidence of the route such grounding has taken in human development. Leave aside, for the moment, the vexed question of *social* context, and the debate over what should be learned over somatic time, and what acquired over evolutionary time. If, ultimately, symbol-level representations are to be grounded in sensorimotor interactions with the environment, it must be desirable that the representations of these interactions are learned from the bottom. Otherwise, their structure will always be opaque to the learning agent, preventing participation in the development of language-like structures.

Superficially, this argument, for an appropriate level for grounding the learning of representations, may seem similar to the argument of Brooks (1990) based on the physical grounding hypothesis (PSH). The foundational inadequacy of the PSH is fully worked out in Chapter 8, where the representational fascicle is finally gathered up.

2.5.2 How to learn

The question of what to learn may also determine, to some extent, how it is learned, but it will be useful to consider this issue in relative isolation. The conventional broad categorisations of connectionist learning are used to demarcate approaches in Chapter 4. Here, however, the intention is to examine the implications of an approach to learning that is particularly common in adaptive autonomous agent and related

research, and to contrast it with the approach followed in this thesis. Although both approaches can broadly be described as reinforcement learning, they have radical differences.

The Q-learning approach (Watkins and Dayan, 1992) is an application of Sutton's (1988) Temporal Difference methods where parameter are adjusted using the difference between predicted and desired outputs. Most implementations of this technique rely on a formulation that tends to undermine the quest for realism in simulation and, when applied to real robots, to unrealistic control prescriptions. This kind of reinforcement learning depends on a discrete conception of state space, in which a finite number of separable states must be enumerable. Therefore, some way has to be found of quantifying both the environment faced by the agent and the actions that it performs. In simulations, this usually leads to the familiar "grid world" environments, where mobile agents move from tile to tile as on a chessboard. In experiments with real robots, it is common to find that control decisions are made at unrealistically long intervals, so that the robot may move quite a large distance between cycles. Such devices appear to have been necessary because the older formulations of Q-learning, at least, relied on a simple look-up table that had to be stored in memory. Thus, the enumerable set of state-action pairs, over which reward is estimated, needed to be kept manageably small. The limitations of this approach for ultimate scaling to real-world applications seem apparent. Even if neural networks are used to replace the look-up table approach, the technique still implies a separate net for each action, and a means of pre-processing sensory inputs to make state discrimination straightforward. Connectionist versions often resort to "high-level" sensors (that can not easily be replicated in hardware) to achieve this state

discrimination in simulation. For example, "sensors" may be dedicated to recognising a single feature in the environment, such as "food" or "a wall". It is apparent that they rely on the kind of "pre-symbolised" representation of inputs used in traditional neural network research, where, for instance a "1" in a string such as "00010000" may be designated "letter d". Another related problem with Q-learning approaches is the difficulty of getting them to generalise. This leads to the "flat policy" criticism (Maes, 1995), although some recent work has appeared that claims generalisation on typical, highly specified toy problems using Q-learning in conjunction with function approximation (Sutton, 1996).

2.5.3 Tabula rasa learning

The course taken in the practical work underlying this thesis was essentially a bottom-up one that exploits learning from a *tabula rasa*. The rationale for this has been developed in accordance with the preceding discussion of alternative approaches. For practical reasons, it can be argued that a higher level of control must be the starting point for achieving complex behaviour. However, the belief that a SAB agent must rely on raw (i.e. relatively un-interpreted) sensory inputs, and ground its problem solving representations in them, seems justifiable from the programmatic perspective of AI. Accordingly, the simulations, described later, abide by that precondition, in the expectation that they will translate successfully to real robotic domains, and ultimately scale to useful autonomous task performance. The guiding principle for the choice of approach has been to minimise the intrusion of the designer's domain ontology into the perceptions, representations and responses of the agent. Thus, the agent is not allowed to have highly abstract sensors of the kind described in section 2.5.2. Although the simulated sensors are only approximations of real sensors, the

intention is that the pre-processing of inputs should not amount to interpretations of the environment originating in the designer's domain. In this requirement, there is strong agreement with other proponents of a "new AI" or "radical connectionism" based on interactive neural networks. For example:

> "There should be sole use of sensori-motor interfaces, that is, inputs consisting of immediate sensory stimuli and outputs consisting of motor commands, instead of pre-digested representations"(Dorffner, 1997, p.97).

2.6 Summary

Some foundational problems in Artificial Intelligence (AI) underlying the research were identified. Project-level solutions offered by the behaviour-based control approach were critically examined. The important role of learning in achieving truly adaptive behaviour in autonomous agents was discussed and different, broad approaches to learning were compared in order to justify the one followed in this research. The next chapter introduces the type of neural network mechanisms in which this learning can occur and begins a discussion of how it should be interpreted.

CHAPTER 3

ARTIFICIAL NEURAL NETWORKS

MECHANISMS AND INTERPRETATIONS

3.1 Introduction

An overview of connectionism is desirable because the research field is multidisciplinary, but it will also provide a vehicle to carry forward the argument begun in section 2.5.3. So far, this points towards an AI programme predicated on building blocks that have the potential to give architectures greater run-time adaptivity, an inchoate position described in Rylatt, Czarnecki and Routen (1995). The need for an introduction will therefore be balanced by the requirement for a relatively succinct account in which to weave discursive threads to be picked up again later.

There are numerous introductory accounts commencing from descriptions of axons and neurones (for example, Haykin, 1994; Fu, 1994). Instead, the overview of feedforward neural networks in section 3.2 is pitched at a higher level of abstraction, and the language tends towards that of the dynamical systems account of the processes involved. This account is seen as a bridge between the traditional connectionist programme, whose foundational weaknesses will be outlined in section 3.4, and the one increasingly advocated here.

In section 3.3, recurrent neural networks (RNNs) are discussed in more detail. This will provide the basis for an adequate account of the rationale for the new architectures described in Chapters 7 and Chapter 9. A changing view of their essential role will be apparent as the argument is developed in Chapters 8.

3.2 Feedforward neural networks

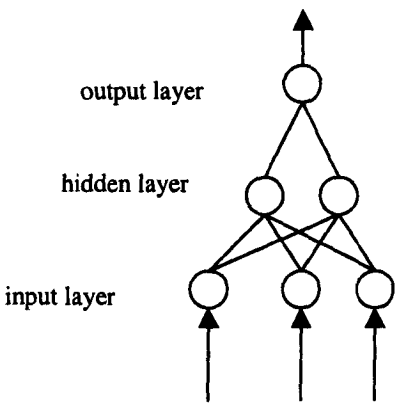

Figure 5: Simple feedforward network.

A typical connectionist system comprises variously realisable, interconnected nodes with certain properties. The connections can be abstracted as weighted directional paths, and the nodal properties as activation strengths to be transmitted along these paths. The bulk of connectionist research, to date, has concerned feed-forward networks in which nodes are conceptualised as belonging to various layers, the boundaries of the system being defined at the input layer and the output layer (Figure 5). In this conception, a stream of activations flows through the system, locally modulated by simple built-in update rules (usually uniform at least within each layer) operating at each node on the sum of activations arriving from the interconnected upstream nodes. Most commonly, nodes are fully interconnected, but also possible are

sparse interconnection, and - most significantly from the viewpoint that develops in this thesis - feedback, or recursive, connections. Connectionist systems are usually described as being connected to "the environment" via the input and output layers. By this account, the activation of an input node is simply the value of the input, and the activation of an output node is something that has informational value for the environment. More precisely, in most connectionist research, it is to the user that the output has relevance, and from the user that the so-called environmental input derives, and so the traditional connectionist system is wrapped tight in user semantics. It almost seems that because neural networks are seen as more obviously "brain-like" than symbolic systems, the language used to describe experiments using them must necessarily have a naturalistic ring, but this can be extremely deceptive. In fact, the conventions and experimental approaches underlying this usage give most connectionist research the character more of traditional AI than of *nouvelle AI* or of research inspired by neuroethology. However, before this line of argument can be developed further it is necessary to focus more closely on some fundamental characteristics of the paradigm. It will be argued that they give it the potential to escape from the constraints of the traditional outlook, in spite of the tendencies that have come to dominate its research programme.

Consider then, that an ANN has three main separable and informational aspects:

- a topology, being a digraph of nodes and arcs;
- a set of weights associated with the arcs of the digraph;
- a set of update rules determining how a node is activated.

These aspects frame both a vector space of all the possible patterns of activation strengths, and a dynamical system of possible trajectories through it. Of interest in most connectionist research is the type of activation vector space that contains sets of stable points from which trajectories do not exit. Each point has associated with it a region of attraction that determines trajectories from any vector within it to the point of stability. It is the aim of most connectionist systems to settle such a set of stable points, in response to "environmental" inputs, so that the regions of attraction will have differentiated the input space into the desired categories or distinct patterns. Usually, the topology and the update rules are fixed at design time. Only the weights change during training, being adjusted according to some measure of error[5]. The process of weight adjustment is referred to as learning, and it occurs in a second-order space of possible weight vectors, each point in this space determining a complete dynamics of the first-order activation space.

Most connectionist systems can be described as first-order, in that the weights, once trained, are frozen, and all processing then takes place within the so-determined dynamics of the first-order activation space. Additionally, the dynamical space of these systems is usually organised around local attractors. Movement into one of the alternative attractors is interpreted as a representation of the category of input patterns corresponding to it. However, interest in this thesis will later focus on networks that support the feedback of activation from downstream neurones to upstream neurones. These networks may have trajectory attractors rather than point or region attractors

[5] Such networks can be trained in various ways but the best known and most successful training or learning algorithm is *backpropagation* (Rumelhart et al., 1986).

and it is in these properties that interesting correspondences with the dynamical nature of cognitive processes may be found. These less well-understood connectionist systems will be discussed in the next section. True second-order systems have also been studied. In these, the activations of one network are interpreted as the weights of a second network, producing a second-order dynamics in the system as a whole after the weights in the first network are frozen. Systems of this kind are also explored in this project (Chapter 9).

3.3 Recurrent neural networks

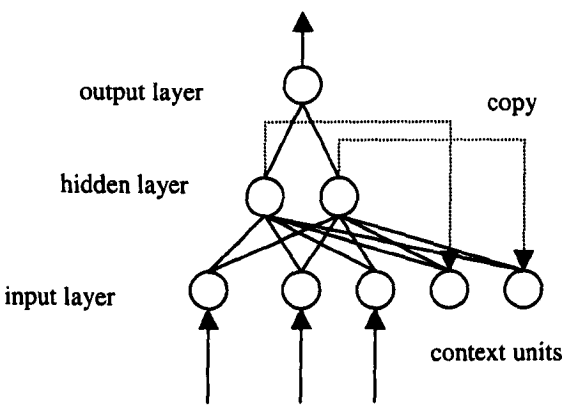

Figure 6: Elman's simple recurrent network (SRN).

The study of recurrent neural networks has become a reasonably distinct sub field within connectionism. What is actually meant by the term is not entirely agreed within the research community and mild arguments can still arise at neural network conferences on this issue. Somewhat curiously, networks that are, by their connectivity, fully recurrent (for example the Hopfield net and some competitive learning architectures) are often not recognised as recurrent networks by those working in this sub field. This seems to be because their own work is really part of a

tradition that branches off the mainstream path of feedforward neural networks trained using supervised learning techniques, rather than the former kind of settling or steady-state network. This is the reason for terminology that retains the global

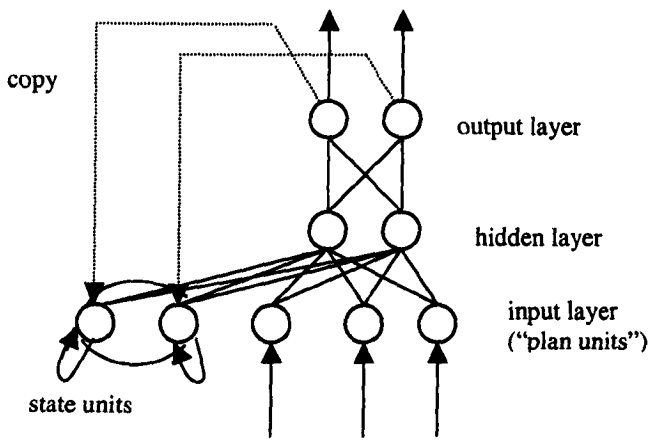

Figure 7: Simple example of a Jordan network showing self- and (optional) inter-connections between the state units. Self-connection weights are set by hand initially and then fixed rather than learned.

description of their simple forms as feedforward networks, with partially recurrent connections. In their earliest form, these networks were trained in the same manner, using the standard backpropagation algorithm. Although it is usually stated that backpropagation has been adapted in these networks by unrolling the network one step back in time, the underlying algorithm is virtually identical. Examples of these partial recurrent networks are first order networks, such as the simple recurrent network SRN in Figure 6 (Elman, 1990), the Jordan network in Figure 7 (Jordan, 1986), the simple dynamic memory (Port, Cummins and McAuley (1995). Second-order examples are the architectures developed by Pollack (1995), (Ziemke, 1996a), and related hybrids (Ziemke, 1996b). These apparently simple networks have dynamical characteristics that make their inner working harder to understand and predict than those of purely feedforward networks. For this reason, they are still the

subject of active research at a very basic level (for example, Wiles and Elman, 1995). An attempt to chart some of their general characteristics is described next.

3.3.1 Mozer's taxonomy of recurrent neural networks

Mozer (1993) presents a taxonomy of this kind of recurrent network in terms of the form, content and adaptability of their short-term memory model. An understanding of these terms will help to explain the relevance of these networks to this research.

Form

The form of a short-term memory model determines how information is stored with respect to time; two characteristics of particular interest are the *depth* and the *resolution* of the memory. One obvious possibility in approaching a temporal task is simply to collect a predetermined number of input vectors, arriving over time, and present these simultaneously to a standard feedforward network. Such a *time-delay* network represents a buffered approach. It is clearly governed by practical limitations on the size of the buffer (in other words, the depth of memory is fixed and usually quite low), but as the actual input values are "memorised", its resolution is high. The partial recurrent network approach avoids the problems raised by a fixed buffer size, but at the expense of some resolution. This is because the memory decays, usually exponentially, over time. In other words, inputs that are more recent will be memorised with greater strength, that is, more clearly, than those more distant in time. Although, in principle, the depth of memory is infinite, the rapid decay appears to limit the usefulness of this characteristic in practice. For example, Elman (1990)

warned that it should not be expected that inputs more than seven steps in the past should have any significant effect on outputs.

Even so, this approach is more promising than one that requires a fixed temporal buffer such as time-delay. Elman (1995) made the point that the dimensions (units) in the input layer of a neural network are all orthogonal to each other in the input vector space. Time-delay networks, and similar networks, simply represent a series of inputs in a convenient and conventional left-to-right spatialisation of time, but neither this arrangement nor the proximity of one unit to another has any intrinsic significance for a feedforward network. Thus, such a network cannot capture essential notions of relationships between, for example, elements in a sequence apparent to human observers. Moreover, it will never be capable of abstracting over the physical order of percepts to form novel temporal structures. In contrast, Elman categorises the recurrent network approach as one that represents time implicitly, that is, through the effects of time on processing. He argues that the dynamical characteristics of such networks enable them to capture some of the subtle temporal relationships that, for example, are intrinsic to human language processing.

Content

The content of a short-term memory describes what is remembered, in terms of the input to the memory. For example, the time-delay approach clearly memorises only the raw inputs to the network. Recurrent networks can be categorised according to the nature of their content. For example, the SRN (Figure 6) is essentially a standard feedforward network with a non-linear activation function, and linear feedback connections from the hidden layer units to the context units in the input layer. It therefore memorises a convolution of the network inputs, transformed by this

function, together with the last memory state (that is the values in the hidden layer, representing the transformed input and context vectors from the last time step). For this reason, it belongs to the class Mozer called transformed input and state memories (TIS). Other possibilities are to remember the transformed inputs only, or to remember the just the outputs. Mozer calls these, respectively, transformed input memories (TI) and output memories (O). The Jordan network is an example of the latter.

Adaptability

Adaptability is a characteristic of short-term memory more difficult to understyand intuitively; it relates to the memory parameters and the degree to which they can be modified during learning. Thus, non-recurrent networks (such as time delay) are not adaptive, as their parameters are fixed and for this reason, they are sometimes referred to as static memories. All recurrent feedforward networks with hidden layers are to some extent adaptive, but require specialised learning algorithms (see the next section) to be fully adaptive.

3.3.2 Recurrent learning algorithms

As well as the aspects discussed in the previous subsection, the learning algorithm (which Mozer did not discuss) is relevant to the choice of approach for a particular problem. Special-purpose training algorithms for recurrent models, such as backpropagation through time (BPTT) and real-time recurrent learning (RTRL) have serious drawbacks. For example, according to Doya (1996), RTRL (despite its name)

requires $O(n^3)$ memories $O(n^4)$ computations and BPTT requires $O(nT)$ memories. Though used to some extent in conventional control applications, they are not likely to be so useful in SAB problems because, typically, they cannot be highly constrained in time. For example, the expanding memory requirements of BPTT would be intractable for arbitrarily long training-sequences. Additionally, according to Pollack (1995), BPTT is unstable. With all recurrent networks trained by algorithms based on gradient descent, there is a risk of poor performance, as Lin, Horne, Tino and Giles (1996) have recently confirmed. The implications of this for SAB will be apparent when a problem sensitive to these limitations is addressed in Chapter 9.

3.3.3 Performance issues

In this section, some other research findings of particular interest to these investigations are briefly mentioned. According to Lin (1994), SRNs are able to discover and exploit task-relevant features in a system's history. Ludik, Prins, Meert and Catfolis (1997) favourably compared the SRN with other recurrent architectures in controlled tests. They reported the Jordan network performed badly. This may confirm the view of Cottrell and Sung (1991) that networks relying on feedback from the output layer only (like the Jordan network) cannot remember input characteristics not directly exploited in their output. The improvements reported by Lin, et al. (1996) using a NARX recurrent network (similar to the Jordan network but with multiple output feedback delay lines) appear to have depended heavily on their use of the BPTT algorithm for training. It may also be possible that, for the same reason suggested in the case of the Jordan network above, the NARX would not perform so well on a wider range of tasks as it similarly relies only on output feedback. Additionally, the problem on which it was tested was particularly toy-like. The

computational power of the SRN was investigated recently by Kremer (1995). Kremer considered only a subset of SRNs with binary inputs and step activation functions. Nevertheless, he concluded these networks were in principle capable of emulating any FSM (hence they have power equivalent to any digital computer with finite memory); only wiring difficulties, problem representation or training techniques limit this power in practice.

3.4 Interpretations

At this point, it is appropriate to begin a discussion – one of the discursive threads woven into this thesis – about the conceptual interpretation of the usage of connectionist techniques. It is started here in order to suggest how, for the most part, the work reviewed in the following chapter is fundamentally and radically different from traditional connectionism. To appreciate this fully, observe that the "environment" often referred to in connectionist research is not the "naturalistic" environment of real robots, or even the simulated closed loop environment of computational neuroethology. Traditional connectionist research instead takes place in the kind of microworld characteristic of symbolic AI. Inputs are usually encoded examples of the categories or patterns that are required to be distinguished by the network. Mostly, outputs do not affect subsequent inputs. As an example of this it is instructive to consider one of the best-known connectionist systems, NETtalk - a system intended to model mental processes that transform (English) text into speech. For NETtalk the environment is merely a set of letters: clearly far from a physical environment sensed directly by the system or influenced by its effectors. At this most obvious level, it seems apparent that traditional connectionist systems have more in

common with symbolic systems than with *nouvelle AI* systems or with experiments inspired by neuroethology.

Verschure (1997) discussed the issue of the designer's domain ontology, referred to in the discussion of behaviour-based control. A strong hint of how this domain exclusively provides the environment for NETtalk is given by considering the system's immediate antecedent, DECtalk, which was a (commercial) system based on symbolic computation dedicated to the same task. In order to achieve this task, DECtalk required input to be encoded by mapping each character onto a description in terms of phonemes, stresses and syntax. This scheme of encoding articulatory features was used virtually unchanged in the NETtalk experiments. Verschure further analysed this encoding scheme to demonstrate how a designer-dependent description of a task, in symbolic terms, is effectively compiled into the network model. He argued that this invalidates claims made by their designers, and other commentators, for the emergence of some kind of symbol level understanding in such networks. The specific claim for NETtalk was that it "understood" the difference between vowels and consonants, as it was able to distinguish them as part of the task, although such distinctions were not programmed into the network. Verschure however argued, quite convincingly, that the distinctions were implicit in the encoding of the inputs, whether known or unknown to the designers - the network only had to rediscover these regularities. He went further, discerning in the example of NETtalk, a procedural commitment to the programme of symbolic AI implicit in designer-dependent symbolic task descriptions.

Clark (1993) takes a less extreme view of this famous connectionist system that is more mainstream but still committed to a revision of connectionism aimed at enabling it to model what he calls *contentful thought*. With such a view, the position in this thesis essentially coincides: even though the strength of Verschure's arguments is recognised, the entailment that standard connectionist frameworks are necessarily inadequate as at least an initial basis for inquiry is not accepted. For example, although the backpropagation algorithm is generally held to be implausible at the neurobiological level (e.g. Churchland, 1992), it can be accepted as an enabling mechanism for achieving error minimisation by gradient descent. This more general procedure is more attractive as a possible brain-like mechanism, even though its low-level implementation is yet to be understood (Churchland and Sejnowski, 1996). This position is essentially the foundation for the studies in the simulation of adaptive behaviour described in this thesis.

3.5 Summary

A brief introduction to the main class of feedforward ANNs that form the basis for these studies was presented. This was followed by a more detailed account of the subclass of partial recurrent neural networks that become the focus of interest later in the thesis. Against this background, the general argument against traditional connectionism that underpins the thesis was introduced. The next chapter describes work that uses and interprets ANNs in a different way, focussing on some antecedents of the work in this thesis.

CHAPTER 4

ARTIFICIAL NEURAL NETWORKS IN THE SIMULATION OF ADAPTIVE BEHAVIOUR

A REVIEW

4.1 Introduction

This chapter provides a critical overview of work in which ANNs have a principal role in the design of controllers for adaptive autonomous agents. A more detailed review of much of the work discussed can be found in (Rylatt, Czarnecki and Routen, 1998). Here only the most closely related research antecedents are discussed in any depth – these are examples of research that address *at various levels* problems of particular interest in these studies, relating to:

- basic performance issues such as cyclical behaviour;
- design questions such as the relative merits of monolithic and modular architectures;
- problems of a more philosophical nature, such as designer bias and the issue of grounding.

Other work using comparable means that addressed more distantly related problems is mentioned briefly.

The question of how learning should be approached is undoubtedly one that differentiates much of the research reviewed here (some researchers have quite polemical views on the matter). This fact, together with the, as yet, mostly non-

incremental nature of work in the field, means that organisation of the subject matter with respect to the conventionally recognised types of connectionist learning remains relevant. That the divisions are not merely convenient but underlie quite fundamental issues should become clear as the chapter and the thesis as a whole unfold. There is an additional area not so readily categorised by choice of conventional learning approach. This so-called neuroethological research tends to look at narrower aspects of closed-loop control, focussing on biologically plausible neural mechanisms that do no fit comfortably into the compartments of connectionist learning. This sub-field is consequently accorded a separate section for the sake of completeness, although no specific research antecedents are to be found here. Another important, indeed, fundamental learning issue is the question of where learning should start (ventilated in section 2.5). As an open issue that is not closed simply by the choice of conventional connectionist paradigm, this theme is developed across the boundaries of discussion and some tentative conclusions are drawn. Each subsection begins with a preliminary discussion drawing attention to some of the main issues raised. Individual examples of research representing the most specific research antecedents are subsequently discussed in more detail. This will show their relevance as practical precedents or as the source of concerns that are more foundational.

4.2 Supervised Learning Approaches

Supervised learning, using backpropagation , has proved the most fruitful area of connectionist research to date in terms of real-world applications. However, for adaptive behaviour research, it may be seen as intrinsically unpromising and indeed some researchers dismiss it out of hand, (for example, Gausier and Zrehen, 1994).

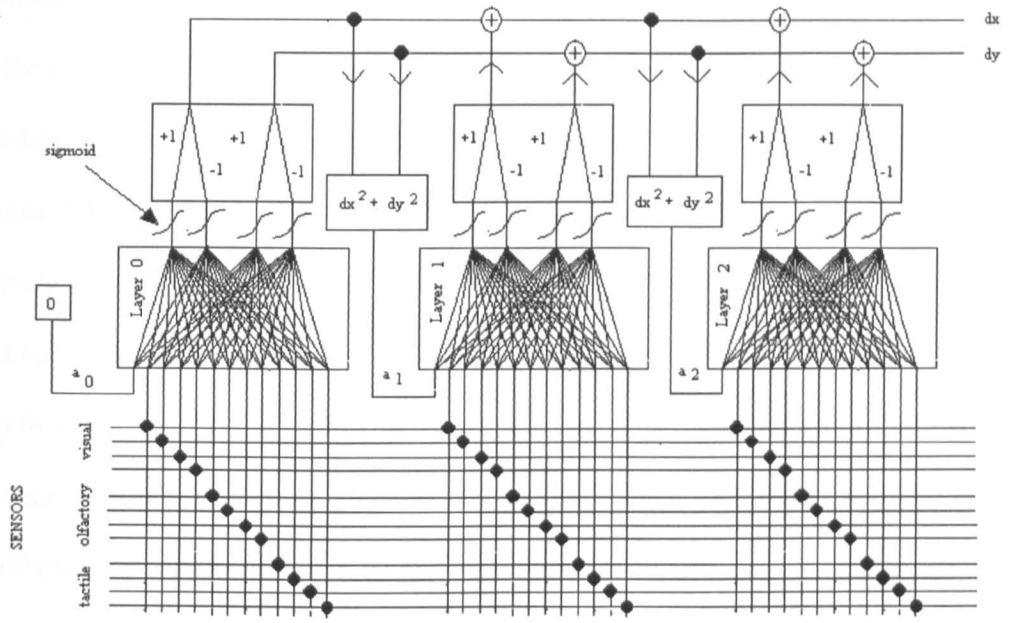

Figure 8: The Addam architecture (re-drawn from Saunders, et al. , 1994, and reprinted from Rylatt, et al., 1998).

This is because traditional supervised learning implies not only the existence of a "teacher" but also a set of correct examples (either collected and presented off-line, or generated by an existing plant being modelled by the network). These are needed to provide the precise error signal used in this form of gradient descent learning. It implies rather more than the human equivalent of learning from instruction, for, to provide correction, the neural network's teacher must have access to the internal representation of the agent's motor outputs. However, recall that behaviour-based and reactive approaches imply that deliberation is delegated to the designer (subsection 2.4.5). Therefore, teaching in some form may well be a useful adjunct as an efficient way of achieving basic competences, without hard-coding, and of gaining some benefits, such as generalisation.

Saunders, et al. (1994) proposed an alternative to the subsumption architecture known as the *pre-emption* architecture. It was instantiated as Addam (Additive Adaptive Modules), a simulated agent constructed from modular backpropagation networks (Figure 8). Each module "knows" when to exert or relinquish control so they co-operate with each other in a way that extends their limited individual abilities. For example, although it was only trained to avoid obstacles in the absence of the goal, and to move towards the goal in the absence of obstacles, Addam can navigate around obstacles in order to seek its goal. This shows that cyclical behaviour problems facing other reactive architectures can be resolved without arbitrary, priority-based schemes, time-outs or random movements. An example of such behaviour would be would be a sequence such as: moving towards the goal, encountering an obstacle, moving back from the obstacle, moving towards the goal, re-encountering the obstacle, *ad infinitum.*

The approach also combats a general difficulty with supervised learning approaches. It has turned out to be formidably difficult to train a multi-behaviour robot in a complex environment using conventional backpropagation. This is because it is hard for the teacher to put himself in the robot's situation in order to generate training sets for a representative set of situations. However, this work indicates that appropriate training pairs can be first be generated to support the learning of a set of individual behaviours. The huge state space problem can then be addressed by modularising behaviours in the subsumption style and by relying on emergent behaviour. This means that he teacher does not have to generate a complete trajectory to enable the agent to reach its goal.

The nature of the simulation represented by Addam is also worth noting. Addam exists in a world of ice and blocks, and searches for "food" using "olfactory" sensors. No attempt is made to simulate low-level motor control (the agent can simply move in eight different absolute directions). Implementation at an arbitrary level of abstraction is explicitly is justified by the authors as an enabling measure, so that the investigation could focus on the problem of interest. Overall, the work is particularly interesting in the way it relates to the subsumption architecture: retaining a behavior-specific modularity while addressing some of its practical and foundational difficulties.

The approach reported first by Tani and Fukumura (1994), and then by Tani (1996), is particularly interesting. The second paper reports the achievement of symbol-level behaviour in a neural network based agent. A behaviour-based robot, with a neural network "high-order" controller, is said to have constructed a "symbolic" process that accounts for its deliberative "thought" process. This is held to be evidence that symbolic processes have been grounded, clearly a very significant claim in relation to the symbol grounding problem (Harnad, 1990). The control architecture is a hybrid of a behaviour-based, obstacle-avoidance control layer, and a neural network based path-planning layer. As in the earlier work, using an artificial potential fields approach, the lower, non-neural layer guarantees the robot's safety. At the outset, therefore, it must be observed that the robot's detailed sensory percepts are not continuously available as input to the neural part of the control architecture. Precisely what is available to the neural network is in fact determined in the designer's domain ontology, as will now be made clear. The robot's task is to build a "forward model of the environment" and then, through an inverse dynamical process, to predict and execute an optimal path to

a goal. The model it builds, however, consists only of a directed graph of points at which the robot can choose to go in one of two directions. The cluttered environment in fact, is devised to ensure that these points are only two-valued. At each point, the high-order controller is faced with a choice between only two range profile maxima and has to decide whether to continue on its existing path or switch to the other one. Once the decision is made, the robot continues under the control of the conventional behaviour-based layer until the next decision point is identified, and control is switched back to the neural network layer. Effectively, this means that the neural part of the controller "sees" only these decision points. The designer determines a simple classification of desired features that will initiate the higher-order controller, so this is not an autonomous part of the robot's operation. The claims must be considered against this obvious lack of a true tractable medium at the level of "raw" sensory input (this argument is developed more fully in Chapter 8). On the other hand, the representations learned by the neural controller are clearly far removed from traditional connectionist representations that merely re-encode the designer's own symbol-level representations.

Pal and Kar (1996) describe a supervised learning approach to sonar-based mobile robot navigation using both a neural network with only feedforward connections and a recurrent neural network. The robot has to reach a goal point without colliding with obstacles on the way. The method depends heavily on on-board odometry (in other words, distance and direction to goal are "known" to the controller in a non-naturalistic manner). The sensory inputs to the neural network are heavily biased by the designer's domain ontology, consisting of range data measured from the predetermined goal direction. The work is therefore primarily of interest from the

project-level perspective. However, it seeks to address one of the deficiencies of conventional behaviour-based controllers identified in subsection 2.4.5, that of cyclical behaviour. It does so by showing the importance of context in situations where reactive behaviour is inadequate – a central theme in this thesis that certainly has implications far beyond the purely practical. The authors first describe an experiment devised to show that a neural network controller without context units gets trapped in the same kind of cyclical behaviour expected of a conventional behaviour-based robot when faced with certain kinds of obstacles. They go on to show that a similar controller with context units is able to avoid entrapment and reach its goal. The work is also interesting in that it appears to vindicate the use of simulations for training neural network controllers using quite simple representations of sensor data. Controllers trained in this manner evidently performed successfully when transferred to the real robot in its physical environment. This theme will be taken up in Chapter 5. Because of the goal-oriented approach and heavily biased sensory encoding, the controller has serious limitations. For example, it has no means of escaping (or even recognising) dead-ends, or of avoiding any kind of concave objects. Additionally it appears to generalise very poorly, being unable to cope with obstacle configurations very much different from those on which it was trained. The authors also acknowledge that the supervised training regime was laborious. They speculate that a reinforcement learning approach might be preferable (although, at the outset, they suggest it is only desirable when appropriate responses to the environment cannot be predicted with any certainty by the designer).

Sharkey, Heemskerk and Neary (1996) argue that subsumption-style behaviours can be demonstrated using supervised learning techniques. They demonstrate that

although a layered subsumption-style approach to co-ordination works satisfactorily, a single controller has superior performance. However, the method proposed entails some preliminary modularisation. This enables the robot to acquire the desired first-level behaviour (avoiding obstacles). Subsequently, a transfer technique is used to install the weight set from the first trained net as the initial weights in the composite net. This is then trained to acquire the second level behaviour ("find goal") while preserving the first-level behaviour. The benefit of this approach is claimed to be that there can be an "analogue" change of behaviour rather than a binary switching of behaviours according to a fixed priority scheme typical of the conventional behaviour-based approach.

In a piece of research with ramifications that became pivotal for these studies, Ulbricht (1996) describes how a recurrent neural network can handle so-called time-warped sequences. These are sequences in which significant elements are repeated an arbitrary number of times with arbitrary pauses between. For example, an agent might observe a particular environmental feature to its right on its way to a junction. It would continue to observe the feature in passing and then lose "sight" of it, perceiving only the blank wall until it detects the junction ahead. When it arrives at the junction, it must remember the significance of the previously observed feature so that it can decide which branch to take in order to reach a goal situated some further distance along one of the routes

Although it is couched in the terminology of SAB research, Ulbricht's approach is closer to traditional connectionist experiments. These have a quasi-symbolic character (see section 3.4) and it is hard to see how her encoding scheme could scale even to a

very simple simulated robot or animat with realistic sensors. Briefly, perceptual time slices are represented by symbols according to the current state, so that, for example, successive "pause" states would be represented as {P*}. Obviously, this is a very high-level abstraction of the real situation a situated agent would face. Unfortunately the novel recurrent network - the *input state network* - proposed by Ulbricht depends rather heavily on this initial symbolic representation and uses a method of transforming the input into a distributed representation. Another potential weakness of the approach is that the temporal decay employed at the input state layer appears to result in a rather inflexible scaled degradation across the input range. The random process of distributing the input representation might result in some representations that are disproportionately affected by this fixed scheme. Although there is an argument that, in this particular problem, inputs carry the information rather than outputs, the intermediate state representation of the hidden units must clearly carry some useful information. Additionally, the general task relatedness of the hidden layer has made the SRN one of the most useful connectionist models for temporally extended problems. Even so, her work is very suggestive of abilities that a real autonomous agent would need beyond the relatively simple behaviours such as phototaxis and obstacle avoidance. The implications of this will become clearer in Chapter 9.

4.3 Self-Organising Approaches

Systems that attune themselves to regularities perceived in the environment are not inherently promising as the sole basis for controlling mobile agents that must have a predisposition to act. Self-organising (it should be noted that the term is occasionally

used far more generally), or unsupervised learning systems, adaptively recode information from the environment. However, they typically do not have the closed-loop characteristics of the controllers reviewed here. In practice, self-organising aspects of ANN control architectures turn out to be dependent on reinforcement learning in some guise or other, or to function as pre-processors for robotic percepts. For example, Fagg, Lottspeich and Beckey (1994) report an approach that avoids the problem of interference between different regions of the state space noted in section 4.2, based on their earlier work modelling primate visual/motor conditional learning. It employs a self-organising, winner-take-all (WTA) mechanism primarily as a feature detector. At this level the WTA, operation is novel in that each unit has influence over only a small neighbourhood of the field, the overall effect being one of contrast enhancement between the input patterns, lessening interference. This work is also interesting with respect to the question of how reinforcement policies should be devised for different tasks.

However, the categorising power of these networks perhaps affords a glimpse of future systems that will exhibit symbol-level autonomy based on internal representations grounded in sensorimotor experience. The examples discussed in this section are representative of this - in the very long run - rather more ambitious, bottom-up approach. Verschure and Pfeifer (1993) show how Hebbian learning mechanisms can enable an agent to develop emergent anticipatory behaviour (collision avoidance) in an agent that has certain predetermined values. These include the ability to detect collisions and move in a certain direction. The claimed the emergence is heavily dependent on the predetermined value scheme – essentially reflexes wired in by the designer in mimicry of evolutionary pre-wiring. Gaussier and

Zrehen (1994) too exploit a form of Hebbian learning to achieve collision avoiding behaviour that was not pre-programmed. In both approaches, some form of reinforcement is used to guide the learning process. In the first example, it is implicit in the set of values bestowed on the agent, whereas. In the second example, it is more obvious in the form of a global pleasure/pain function. Moreover, in both some form of topology-preserving map is generated more or less in real-time by novel means. Gaudiano, Zalama, Chang and Coronado (1996) discuss obstacle avoidance using a neural model of operant conditioning originally proposed by Grossberg (1971). Operant conditioning is also referred to in the psychological literature as reinforcement learning, but although a negative reinforcement signal is used to encourage learning, it is the self-organising aspects of this model that are of most interest.

4.4 Reinforcement Learning Approaches

Reinforcement learning is currently a popular approach to learning in control applications. It can take a number of forms: Q-learning, the most popular, was briefly discussed in subsection 2.5.2 where it was suggested that it has certain foundational difficulties as a prospective paradigm for a programmatic AI. Generally, reinforcement learning is employed in situations where a representative training set is not available and the agent must itself acquire this knowledge through trial and error interaction with its environment. In contrast to supervised learning, the emphasis is on exploration rather than generalisation. It has been indicated that it often plays a role in approaches that are also described as self-organising (section 4.3). In these

manifestations, the term reinforcement is used broadly in a sense that derives from psychology or is only generally inspired by the more specialised interpretations that have currency in control applications. In particular, the global reinforcement signal is only a coarse measure of utility, for example: moving from a state of collision to a state of no collision. More typically, reinforcement learning research concerns itself with problems in which reward is delayed or sparse. There is typically some notion of optimising task performance over time rather than acquiring basic competences: for example, learning to navigate efficiently rather than merely avoiding obstacles reliably. Barto (1990) distinguishes three types of reinforcement learning task: non-associative reinforcement learning, associative reinforcement learning and adaptive sequential decision learning. In the first of these, a learning system receives only evaluative input. No examples are included here, but they have been studied under the umbrella of genetic algorithms, for example, Harvey, Husbands and Cliff (1994), Almassy and Verschure (1992). In the second type of task, a controller aims to maximise the immediate evaluation at each step, and receives information in addition to the evaluation of its control signals. Typically, in the class of problems discussed here, this is in the form of inputs from a robot's sensors. In the third type, the maximisation of long-term performance may entail foregoing immediate favourable evaluations.

In Lin (1991) and Lin (1993), the method used to perform the sequential decision tasks is Q-learning. The degree of autonomy achieved by the robot is limited because the domain knowledge necessary for task decomposition must be provided by the human designer. It is interesting to compare this work with the modular pre-emption architecture discussed in section 4.2. The lesson from both these examples appears to be that training multiple simple networks using some form of task decomposition is

easier than training complex monolithic nets. Of the two, Lin's approach leads to a more efficient state-space representation because the robot can ignore sensory inputs that are not required for a particular subtask. On the other hand, the approach requires explicit, external rules. These determine which skills should be used, and when skills should be switched. This appears to rule out the emergence of new behaviours observed in the pre-emption architecture and may perhaps lead to the inflexibility more characteristic of the subsumption architecture. Thrun (1994) too employs Q-Learning, combining it with a connectionist version of explanation-based reasoning. A rare example of structural learning is to be found in Millan's approach (Millan 1994), where nets are built dynamically based on the robot's experience in order to facilitate incremental learning[6]. Chester and Hayes (1994) also address the problem of incremental learning but their algorithmic approach seems to carry questionable computational overheads. A hybrid architecture using modular networks and so-called reinforcement learning is described by Franchi, Morasso and Vercelli (1994), consisting of sets of expert and heuristic modules together with a critic implemented as a gating network. The experts are simple perceptrons and the heuristics are fixed nets or rule-based algorithms that implement mutually exclusive behaviours. A human operator provides reward and punishment signals.

In some experiments with a relatively simple network, the *pattern associator network* (Nehmzow, Hallam and Smithers, 1989; Nehmzow, Smithers and McGonigle, 1993; Nehmzow and McGonigle, 1994; Nehmzow, 1995) only linearly separable problems are set. It is well known that SLPs can only learn functions of this kind. In this

[6] Standard backpropagation approaches are prone to the danger that learning can be undone by new training instances.

limited setting, the pattern associator's virtue of rapid learning could evidently be exploited. In the first two of these papers, the learning that occurs is described as "self-organising", though the architecture appears to include a "teacher" to provide some form of reinforcement signal. Subsequently, the pattern associator is trained in a manner described as "supervised", but in fact employs an external teacher so that the signal is more typical of reinforcement learning in that no explicit correct exemplar is given as a target. Instead, the teacher uses one of the robot's light sensors to signal when a motor action for a given sensory stimulus is correct or incorrect in the manner of reward and punishment signals. Although this scheme may appear unsophisticated, the ability of the robot to learn quickly - not only individual behaviours but also combinations of behaviours - seems quite impressive. However, a possible concern is that the approach generally seems to rely heavily on the designer's domain ontology even to learn quite basic behaviours. For example, to locate a wall the robot is programmed to move a fixed number of steps to one side, then a fixed but larger number of steps to the other side, and so on, until it makes contact. This approach is calls to mind the behavior-based engineered solutions (subsection 2.4.5).

Meeden, McGraw and Blank (1994) describe some interesting learning experiments involving Barto's second type of reinforcement learning. Complementary reinforcement backpropagation (CRBP, Ackley and Littman, 1990) was adapted for temporally extended problems of obstacle avoidance and light seeking using a simple model car-based robot, Carbot. In CRBP, a real-valued *search vector* on the output layer is interpreted as a set of probabilities from which a binary vector can be generated stochastically. The difference between this and the search vector provides the error measure for backpropagation: the CRPB algorithm then determines the

direction of backpropagation according to whether the actions produced by the output are rewarded or punished. Different learning rates are applied in each case, according to the information value of the reinforcement signals.

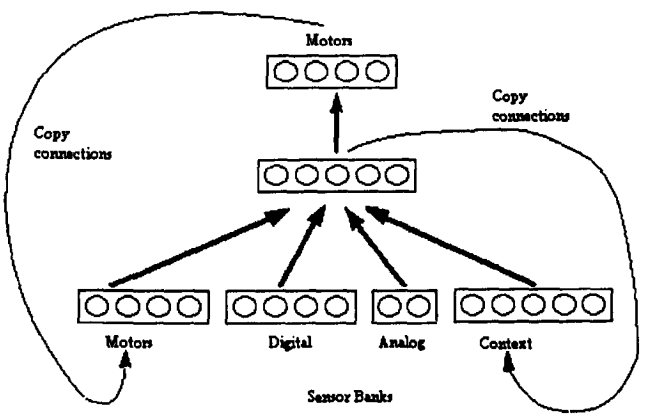

Figure 9: Meeden's control architecture for Carbot. Re-drawn from Meeden et al., 1994 and reprinted from Rylatt et al., (1998).

In this domain, because Carbot is rewarded just for moving in the environment without collision, a much lower ratio was found appropriate than in the original static application. This work is also interesting from a network topology perspective in that some of the experiments use SRNs to provide a short-term memory (STM), or context (Figure 9). Attempts were reported to assess the utility of contextual memory by varying its size, but the results do not appear very conclusive. Nonetheless, this work is notable for its attempts to investigate and analyse, using quite rigorous statistical

techniques, a number of learning subtasks singly and in combination, including prediction and autoassociation, using a variety of topological variations on the basic controller. Arguments that Carbot's behaviour can be interpreted as plan-like are less convincing. It seems more likely that any apparently systematic behaviour observed can be interpreted as some kind of limit cycle in the dynamical system made up of Carbot and its very small, movement-restricting environment. It is evident that human planning often, even typically, proceeds off-line, that is, not during interaction with the subject of its proposed actions. Therefore, the leap to planning in any sense comparable with human planning must surely be preceded by real progress on the issue of representation.

Ziemke (1996a) used a modification of the self-adapting recurrent network (SARN) as the control architecture and CRBP as the learning algorithm. SARNs were devised for formal language recognition (Pollack, 1995) but in Ziemke's application, they enable the agent to behave as a finite state automaton at the level of pre-defined control tasks. The robot performs a sequence of manoeuvres involving collision avoidance, switching between distinct states each time it detects a light source. A SARN incorporates two networks whose relationship is characterised as "master and slave". This is a functional (rather than task-based) modularity, in which the master net adjusts the input to hidden layer weights of the slave net, and has no independent interface with the environment. The approach can thus be effectively construed as monolithic.

4.5 Computational neuroethology

The structure of most ANNs only crudely approximates the imperfect knowledge we have of the actual structures and processes in the brain and some approaches such as backpropagation may indeed be biologically implausible. Although current technology cannot hope to emulate the immense complexity of these networks, a possible way of enabling robots to operate in environments that are more complex is to build controllers based on far more faithful and detailed neurophysiological models. However, when the aim is to produce a complete, working robot, a problem with such biologically inspired approaches is the state of neurophysiological knowledge itself. Currently it may provide data only for a small subset of the required functions, leaving the designer to fill in the missing parts with *ad hoc* solutions. Thus, Beer, Chiel and Sterling (1990) advocate striking the "right balance between biological reality and computational and conceptual tractability". That an approach based on computational neuroethology can nonetheless be motivated by practical engineering concerns is confirmed by Hallam, Halperin and Hallam (1994). An important goal of their work is to reduce training times below those typically achieved, so that a robot can be trained on the many different sensor states it will face in a real environment. The position of Scutt (1994), that the lack of a strong physiological basis leads to the design of systems from scratch (cf. Brooks' original engineering approach, 1986), is perhaps more speculative. It rests on the proposition that somehow the road to the achievement of truly intelligent agents may thereby be missed. It may prove that the only way to simulate and organise the complexity observable in natural cognitive systems is from the bottom up, in Scutt's sense (cf. Brooks' later eclectic approach, 1994). Even so, the question of what the correct level of abstraction should be is not fully answered.

In the work of Beer et al. (1990), a compromise is reached by modelling characteristics that appear to have fundamental significance for the control of behaviour by neural systems. This turns out to be at an intermediate level between biological nerve cells and the simple units typically found in connectionist models. The most noticeable difference between this and standard ANN approaches is that the model uses time and voltage dependent intrinsic currents. These capture the net effects of mechanisms that give individual real nerve cells the character of dynamic systems rather than simple functions.

No such compromise is to be found in the work of Edelman (1992). Edelman, as a neuroscientist, has actually studied biological brains. His view is that connectionist-style neural networks are too simple ever to lead to anything like cognitive behaviour. The simulations and simple robots such as DARWIN IV and NOMAD developed by his group should be seen mainly as proofs-of-concept for Edelman's biological theory of mind, the Theory of Neuronal Group Selection (Edelman, 1987). It is clear that Edelman expects artificial forms of cognition to be developed only in response to improved understanding of actual brain mechanisms based on theories closely tied to advances in biology research.

Aitken (1994) proposes an interesting higher-level architecture inspired by the gross structure and function of the cerebral neocortex. His suggestion is that a structure of this kind could sit on top of a typical connectionist, reactive architecture that has mastered fundamental motor abilities; the neocortex model would then enable the agent to perform complex behavioural sequences. Conceptually, the idea requires the

lower levels of a layered reactive architecture to be viewed as an extended environment with which the higher-level architecture can interact. Essentially, lateral and vertical recurrent connections enable representations of correlations between motor and sensory states to be formed. Higher-level associations of these correlations can then be processed to compose behavioural sequences responsive to changes of context rather than simply being read out of memory. These ideas are significant in view of Toate's (1994) suggestion that a useful definition of true cognitive behaviour would entail the ability to break out of fixed patterns of behaviour, whether hard-wired or over learned. Aitken's proposal is that the neo-cortex model can respond to contextual cues and trigger appropriate behavioural sequences of more primitive motor activities, dealing with such problems as overlapping sub-sequences by forming yet higher level associations of correlations between modules. Unfortunately, Aitken does not report any experimental results, so his architecture can only be taken as a proposal for higher-level control. As described, it is an example of task-independent modularity whereby modules have generic motor, sensory and associative functions.

4.6 Summary

The chapter described work on adaptive autonomous mobile agents using neural networks, considering these primarily from the perspective of their chosen or innovated approach to learning. Promising areas for development and some unifying themes emerging from this mostly non-incremental body of research were identified as of particular interest for investigation, viewed in the context of the more general discussion in Chapter 2:

- the appropriateness of supervised learning for autonomous agent research;
- the issue of modularity; and
- the role of temporal processing (recurrent neural networks).

The full significance of these issues emerges as the investigations unfold, as discussed in the following chapters.

CHAPTER 5

THE ROLE OF SIMULATION

5.1 Introduction

This short chapter introduces the simulator used as the basis for the experiments described in the following chapters. Its design and facilities are briefly described, and compared with those of simulations described in other published work. Coupled with this is a more general argument concerning the value of simulation in work of this kind, using examples and opposing views from various published sources. Evidence that this debate is still open can be found in the review of work in Chapter 4. The issues are worth exploring, not only to provide some justification for the approach used in these investigations, but also because they are intrinsically interesting in relation to broader implications for research directions in the field.

5.2 Arguments for and against simulation

In a sense, it may seem unnecessary to try to justify the use of simulation in a field that, after all has the very word in its name. However, the kind of simulation described in this thesis is of a kind that has caused much argument over the years. Nehmzow (1992), for example, devoted a whole chapter of his doctoral thesis to the issue, complaining that most work in the field up to that time had been simulated (though he was referring more specifically to mobile robotics research). He argued

against the practice except in certain tightly constrained circumstances where requirements of available theory, data and predictability could be satisfied. Nehmzow held that simulation is viable only in situations where existing physical examples (he gave the example of a turbine blade) provide sufficient knowledge to make the simulation realistic. In a new field, like mobile robotics (Nehmzow argued), adequate theory and data do not yet exist to guarantee simulations with the necessary fidelity.

Of course, if the research aim is highly specific to a particular mobile-robot platform, depending, for example, on its precise dynamical characteristics, then simulation may have no role or only a limited one, as Nehmzow suggested. This is by no means always the case in SAB research where the aim may be to simulate something less specific, such as "minimal cognitive behaviour" rather than an engineering artefact with a precise performance specification or mission-critical application. Nehmzow, however, took the argument much further, building his case on the idea that certain interesting behaviours can emerge from the lower level dynamics of a particular robot. He argued that, if these dynamics were not modelled in the simulation, the new behaviour would not be able to emerge. An example, given by Nehmzow, was of a wheeled robot that escaped from a dead end, although there was nothing in its control program that specifically encoded such behaviour. The robot was programmed to turn right when its left touch sensor came on, and left when its right touch sensor came on. Therefore, in a tightly enclosing dead end it would be expected to exhibit a cyclical behaviour pattern and never be able to escape – exactly the behaviour Nehmzow observed in a simulated robot with these characteristics. However, in the real robot the separately driven wheels happened to have differential torque. Consequently, after some time, the left touch sensor usually encountered the right hand wall, causing the

robot to turn right and escape. From this example, Nehmzow argued that experiments involving real robots are essential, as it is otherwise impossible to foresee how such apparently trivial details can so profoundly influence behaviour.

On the other hand, for *some* kinds of adaptive behaviour investigations it might be more interesting to produce some behaviour in simulation that could *not* depend on such happenstance characteristics. If this behaviour was repeatable to some extent (by some measure of confidence) on a real robot, then it could be argued that some more general principle had been demonstrated. Jakobi's proposal (Jakobi, 1998) for a theoretical and methodological framework for the construction of robot simulators is interesting in this light. Though it must be read in the context of *evolutionary* robotics (where the need for simulation is more compelling in view of huge computational requirements) the principle of what he calls *minimal simulation* seems generally valid. As Jakobi argues, to model details of dynamical systems peculiar to a specific system that have no effect, or insignificant effects, on ultimate performance is not only unnecessary, but sometimes undesirable. This is because controllers may evolve or learn to exploit those features rather than more general and less brittle ones. In this light, the fact that Nehmzow's simulated robot did not escape from a corner does not mean that it could not do so if it possessed better adaptive mechanisms at the control level. However, it must be admitted that these remarks are made post hoc and that the full implications of Jakobi's arguments cannot be claimed as justification for the rather more minimal simulations forming the basis for the present studies. For these, independent support for this can only be found in less worked-out sources. For example, experiments reported by Meeden et al. (1994) indicating that it is sufficient to take into account certain readily observable characteristics of the real robot, rather

than a complex dynamic model, and to model the effects of noise quite crudely. According to the authors, simulation results were replicated very satisfactorily by the real robot using the same "transplanted" controller developed in the simulator. Because the real robot in practice encountered less noise, it often performed tasks more effectively than the simulator. However, it must be conceded that Meeden's simulator was based on measurements taken from the real robot.

Brooks (1991b) also questions the use of simulation. Although the first subsumption experiments were simulated (Brooks, 1986), Brooks subsequently emphasised the importance of "realistic" environments (Brooks, 1991a). He argued that any simplifications could result in a subtle chain of dependencies between system modules, making the complete system effective only in environments with similar simplified properties. This argument may well be valid in relation to the methodology of task decomposition underlying the subsumption architecture. From an alternative perspective, it may be seen merely to reinforce the argument that the incremental testing and debugging, engineering approach only leads to "one-trick ponies" (see Subsection 2.4.5 of this thesis).

To some extent, the need for simulation - and indeed how the term is to be interpreted - can depend on the research perspective. The "simulation" in "SAB" is of course to be interpreted very broadly and on different levels. It includes, but is by no means restricted to, the use of physical robots that may be used in a sense to simulate the behaviour observed in biological creatures. More commonly, of course, the term means computer simulation. However, the inter-disciplinary nature of the field admits investigation of agent behaviour at various levels, and it is considered legitimate for

research at a given level to assume that possible lower level problems are solvable. This would be a justification for the high-level simulations much favoured by the Q-learning community, for example, the typical "gridworld" experiments described in Sutton and Barto (1998). In these, there is no attempt to simulate low level sensorimotor aspects. The simulations in the present studies however were much finer grained. The intention was to explore the possibility of integrating aspects of learning and control initially at the sensorimotor level so that mechanisms involved in the grounding of more complex behaviour could be studied. Examples of similar approaches can be found in the literature: for example, Pal and Kar (1996) simulated sonar in a comparable, somewhat idealised manner. They showed that an ANN controller, trained using a simulator (apparently with no simulation of noise), and transferred directly to a mobile robot equipped with real sonar, could navigate reliably in the real world and avoid obstacles. Stein (1992) described a simulation using sonar beams modelled by a line-scan algorithm apparently similar in effect to the one developed for the experiments described later in this thesis. MetaToto (described as an "imagination shell" for the real subsumption robot, Toto) learned to navigate through the simulated environment using these sensors. Once embodied in the real Toto, it was able to navigate through the corresponding real world environment using real sonar.

It has been argued that simulation can be an acceptable surrogate for real robots in the context of research aims that are not fixed on performance issues relating to particular robotic platforms. To clarify the position in these studies still further, it is worth noting that the opponents of simulation appear to divide broadly into two camps that may be labelled *metrical* (Nehmzow) and *situated / embodied* (Brooks). The metrical

position may be summed up as essentially the desire to establish a separate science of mobile robotics based on measurement and optimisation. For example, not only should a robot be demonstrated to exhibit wall-following behaviour, but the accuracy of the wall following should be measured. Clearly, this view also entails some move towards standardisation so that claims and results could be compared and assessed realistically (for example, through the use of a standard robotic platform for research). To those workers who do not view their research in these terms but rather within the much broader constraints of SAB, such arguments can seem unduly restrictive.

The situated / embodied position is more problematic. Certainly, the argument that embodiment is a complete panacea for the ills of traditional AI should not be accepted too devoutly. For example, Vera and Simon(1993) observe that the designers of the Navlab robot vehicle abandoned a subsumption approach in favour of a hybrid neural network / symbolic AI controller because they found that performance established in so-called real world environments (such as laboratory rooms and classrooms) was not repeatable in true natural environments. This is not of course an argument for computer simulation, but it suggests that situatedness alone is no guarantee of transferability. It may also confirm the suggestion expressed above that choice of methodology and paradigm may be significant factors in determining whether transfer from development environments, simulated or otherwise, to more realistic worlds, can be effective. On balance, the evidence considered above indicates that systems based on neural networks at least can successfully transfer from simulation to laboratory experiment and even to real world performance. Aside from this, there is also a question of emphasis, in short, whether the aim is primarily to establish a new science

specifically of mobile robotics, or to investigate general ideas that may yield insights for a new AI.

5.3 The integrated mobile robotic agents and neural network simulator.

The experiments in these studies used a simulator specially designed and implemented by the author. This integrated mobile robot and neural network simulator (IMRANNS) does not provide any features beyond those required for this specific research, having been developed piecemeal to meet research requirements as the work progressed. The acronym IMRANNS is used in this thesis for ease of reference; it does not imply any wider use of the system. It was coded in C++ but has a conventional modular, rather than object-oriented, design and runs only under MSDOS. Most of the experiments were performed on a PC with a Pentium processor running at 120 Megahertz. Some features of the simulator are described below.

5.3.1 Mobile robotic agent simulation

IMRANNS provides a medium resolution, graphical simulation of mobile robotic agents modelled at a level similar to the simulations described in the last section, and to that found in the Khepera simulator (Michel, 1996). Inspection of the actual source code for the latter confirmed that there is was no attempt to build a dynamic model of a particular robot or to model sensors with complete realism. Movement on screen and the behaviour of sensors are fairly realistic (taking a bird's eye view) and can reasonably be expected to correspond, within acceptable tolerances for this kind of experimentation, to those of a real robot with similar characteristics, as discussed in

the previous section and confirmed by various workers using Khepera , for example (Ziemke, 1996b), (Jakobi, 1998). A screen shot of the test environment and monitoring screen (as it had evolved by the time of the experiments described in Chapter 9) is shown in Figure 10.

For the purposes of the experiments described in this and subsequent chapters, the simulated robot was provided with a range of sensors and motor commands. The number of different sensor types is fixed as follows, but all sensors can be switched on or off both individually and in banks according to the needs of a particular experiment:

- tactile strip sensors (or "bump-sensors");
- range finders (approximately representing "laser range-finders" or "active sonar"); and
- light sensors (or "photoreceptors").

The tactile sensors are modelled simply as points (pixels) on the periphery of the robot. These have Boolean attributes associated with them. The value of a particular point is set to true or false according to whether it is in "contact" with a point or points representing, for example, part of a wall or an obstacle. These points are colour-coded so that a test for adjacency can be made on each time-step using built-in graphics commands. The range finders are modelled using a line scan algorithm. A ray of pixels is projected, from the centre of the robot, through the locus of each range finder on the robot's circumference. Each pixel is checked for contact in the same

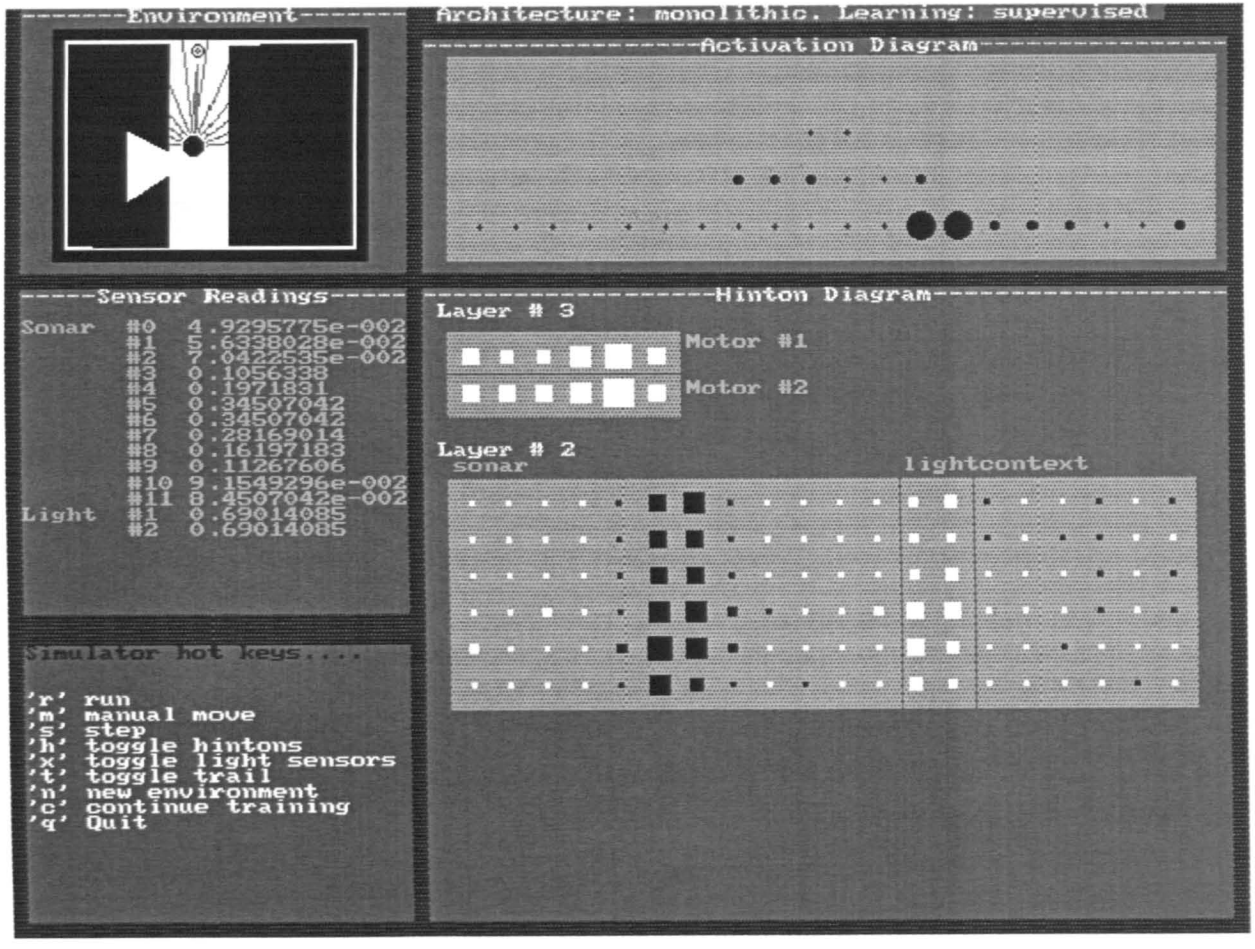

Figure 10: Screen shot of IMRANNS main user interface.

way as for the tactile sensors. This approach is more realistic (from the robot's point of view) than that described by Meeden (1993) or that discernible from inspection of the code for the Khepera simulator. Both of these require pre-knowledge of the location of obstacles so that the screen coordinates can be stored. However, the photoreceptors are not modelled with great realism owing to the computational demands this would entail. Instead, the absolute location of a light source is used simply to calculate the straight-line distance of a photoreceptor and the intensity is then modelled using an inverse-square law.

The motor commands available differ in the level of abstraction (amount of realism) and the kind of environment in which the robotic agent operates:

- two fixed directions (forward and backwards) for preliminary one-dimensional environments (not reported in detail here – they were carried out merely to confirm Meeden's findings (Meeden, 1993) and thus validate the simulator's performance);
- eight fixed directions (compass points) for the early two-dimensional environment studies described in Chapters 6 and 7;
- relative directions (steer right or left 45 degrees, straight ahead or reverse) for the later two dimensional studies experiments (Chapter 9).

The less realistic of these motor simulations were used because the early experiments were inspired by the work of Saunders et al. (1994) with the Addam architecture and Meeden's one-dimensional experiments (Meeden, 1993).

The simulation of the physical environment permits obstacles of arbitrary shape to be drawn within a room-like enclosure. These obstacles can be made opaque or transparent to the range finder sensors and the light-sensors. Light-sources can be installed at arbitrary locations and may be switched on and off by contact with the robot. The simulated robot can be moved around "by hand" (using the cursor keys) in its environment. This enables the collection of sensorimotor training data for supervised learning, the close inspection of localised behaviour during testing of controllers and placement of the robot at different start points for study variations.

The main robot-simulator viewport of IMRANNS permits sensors to be monitored both as numerical values and visually (as range finder "beams" and "light rays") while an experiment is running. It allows inspection of neural network parameters in the form of animated Hinton diagrams. Training progress can also be monitored by inspection of output error levels.

5.3.2 Neural network simulation

The source code of the simulator is not discussed in any detail here, as it is does not have any novel aspects. However, some general remarks on the approach used will be appropriate.

The neural network layers were modelled as dynamically allocated, multidimensional arrays encapsulated within structures (C-style "structs"), to give computational efficiency and flexibility. The interface to the neural network construction code is

somewhat crude. However, it enables neural networks such as multi-layer perceptrons, partial recurrent networks, and multiple network architectures to be specified and implemented rapidly via a simple "question and answer" style interface. This was chosen because a disproportionate amount of time would have been required to develop a sophisticated user interface, bearing in mind that only the originator of the system was foreseen as a user. Conventional backpropagation and CRBP algorithms can be selected. A control architecture, once specified, can be easily connected to a simulated robot with a specified sensorimotor configuration. Additional network input units representing abstract goals can be specified as well as extra output units for auto-association and prediction tasks, as described later.

5.3.3 Validation

The underlying neural network algorithms for backpropagation of the purely feedforward networks and the partially recurrent neural networks were tested and validated on neural network architectures built using the simulator. These corresponded to specifications provided by Freeman (1994) for the "T-C pattern recognition problem" and the "one-to-many time sequence problem". The simulator performed well in these tests, converging to solutions in cycle-counts equal to the best cases reported by Freeman. Additionally, simple one-dimensional experiments reported in detail by Meeden in her doctoral thesis (1993) were replicated. In these, the robot can only go forwards and backwards along a "track". This was done in order, primarily, to validate the CRBP algorithm (described later), but also to obtain some perspective on the way that the robot simulator performed. It was observed that

a simulation similar to the one described by Meeden in terms of architecture, sensors, motor commands and learning parameters, produced very similar behavior.

5.4 Summary

This chapter introduced the IMRANNS simulator used as the basis for all the experiments. It presented some in-principle arguments for using simulation and some in-practice evidence for its viability. Some advantages of IMRANNS over another much-used mobile robot simulator were described and details of its sensorimotor, environment and neural network modelling capabilities were given. Validation of the simulator was also described. Descriptions of the studies conducted with the aid of the simulator begin in the next chapter.

CHAPTER 6

ARCHITECTURES AND STUDIES (I)

LEARNING AT THE SENSORIMOTOR LEVEL

6.1 Introduction

This chapter is the first of the three main chapters concerning novel ANN-based architectures. They were designed and implemented as the focus of these investigations, and demonstrated in some simple studies of simulated adaptive behaviour. The chapter falls into three principal parts, describing respectively the first of these architectures, the studies undertaken with it, and some observations and conclusions. In the first part (6.2), an approach is described and justified that represents the first attempt at incorporating reinforcement learning in the form of CRBP (section 4.4) into a modular architecture. This can be viewed as a hybrid approach. Connectionist learning mechanisms were deployed at the sensorimotor level of its constituent modules, while at the overall control level there was an algorithmic solution to the problem of co-ordinating the lower level learning. In the second part (6.3), the general nature of the behaviours investigated is discussed. Some specific studies of behaviour based on the control architecture are presented, together with some observations in relation to certain of the antecedents discussed in Chapter 4. Finally (section 6.4), the main shortcoming of the architecture is examined – namely that learning is not integrated at the overall control level. Conclusions are

given which lead to the modification and development of the approach described in the following chapter.

6.2 The continuous reinforcement layered learning architecture

6.2.1 Background and justification of approach

The architecture and the studies described in this chapter represent a synthesis of some ideas and approaches discussed in section 4.2. The work of Saunders et al. (1994), based on the so-called pre-emption architecture instantiated as the simulated mobile agent Addam, suggested a possible approach to reactive navigation. It preserves some of the higher-level structural precepts of the subsumption architecture in a modified form, offering a way of overcoming the deficiencies of handcrafting (see subsection 2.4.5 of this thesis). The authors also claim increased the potential for emergent functionality (Steels, 1994) by exploiting the potential of connectionist adaptive learning. However, the methodology appeared to be weakened by a reliance on supervised learning that would be difficult to remove without significant architectural changes, as will shortly be made clear. Explicit teaching of robot behaviour may well have a significant role in the development of autonomous mobile robots. However, the position taken at the outset of these studies was that that the more interesting and challenging prospect is of robots that can learn from their own experiences (see, for example, Mahadevan and Connell, 1993). Ultimately, they should be able to operate successfully in circumstances that have not been foreseen in detail.

Meeden (1993) indicated that it was possible to adapt a form of reinforcement learning that does not carry the unrealistic assumptions typical of the mainstream of reinforcement learning approaches, such as Q-learning (see subsection 2.5.2 of this thesis). It offered, in CRBP, the additional advantage of a familiar, well-researched and successful learning algorithm used in supervised learning, backpropagation, in only slightly modified form. Meeden, however, investigated only a monolithic architecture and focused on single behaviours. Although she refers to "concurrent" training on two behaviours, Move-and-Avoid and the so-called Seek-Food (more properly light-seeking), concurrent performance of the two behaviours is not subsequently discussed or analysed, only the separately learned behaviours).

A question of considerable interest was whether a modular agent similar to Addam could learn similar behaviours by trial and error, rather than by explicit teaching. Supervised learning may be an improvement on handcrafting behaviour based layered architectures. However, as a design methodology for adaptive autonomous agents (rather than merely an ad hoc means of investigation), it does not do enough to lessen the influence of the designer's domain ontology.

Another challenge faced in these initial studies was the problem that when faced by discontinuities in the input space, a single network may experience learning difficulties. Such discontinuities are likely to be found between the different kinds of sensory input available to a SAB agent. The proposal was to combine trial-and-error learning, based on medium-fidelity local sensory data as in Meeden (1993), rather than global knowledge, with an architecture modularised with respect to different

sensory modalities[7] (as in Addam). This represented a synthesis of ideas with potential for solving the practical problems mentioned above. Additionally, although the use of reinforcement learning does not entirely allay foundational concerns, they are certainly lessened. This is particularly so in the context of the sensorimotor modelling approach common to both Addam and the architecture described in this chapter.

6.2.2 Design and implementation

The first new architecture to be discussed in this thesis is the so-called continuous reinforcement layered learning architecture (affectionately known as Crill), shown schematically in Figure 11, which was first described in Rylatt, Czarnecki and Routen (1996). This represented an early, partial attempt at introducing reinforcement learning principles into a modular architecture similar in outline to that of Addam. A modified CRBP algorithm was used only at the level of individual modules, that is, at the sensorimotor level. Even so, the implications of this for the overall control approach are quite significant.

Recall that although Addam's pre-emption architecture preserved the subsumption idea of concurrently active behavioural layers, it replaced the AFSMs of subsumption with ANN-based modules, trained using backpropagation. A module assumes control when it recognises a sensory input vector appropriate to its particular competence. The ability to do so depends crucially on the learning paradigm. Modules must converge to the required control outputs during training; their weights are then frozen

[7] While this idea clearly has some basis in neurophysiology (see, for example, Anderson, 1990), it should be noted that there is no intention to model brain structure in any detail – the analogy is only inspirational

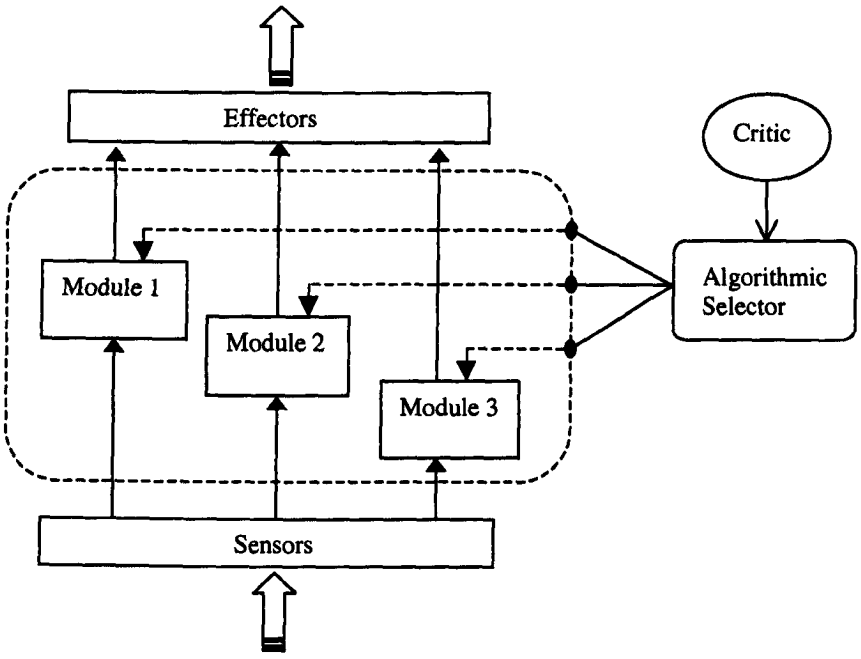

Figure 11: Modular reinforcement learning architecture with algorithmic selector.

to ensure that the same control outputs will be produced in the presence of corresponding input vectors.

Clearly, the same mechanism could not be relied on to achieve coherent behaviour between modules in the case where the agent learns continuously by trial and error in its natural environment. The last idea may be considered in relation to the observed behaviour of the young of many species. For example, a young animal may not have fully mastered the ability to walk, but will move towards food or flee from sudden danger, improving its walking behaviour in the process. It does not wait until it has perfectly mastered a skill or competence, but develops numerous competences

1. Construct backpropagation network with input dimensionality n and output dimensionality m.
2. Collect continuous sensory input vector i and forward propagate to produce search vector of the continuous values s_j.
3. Generate a binary output vector o as follows. Given uniform continuous random numbers ξ in the range (0.1,0.9),

$$o_j = \begin{cases} 0.9, & \text{if } \xi \leq s_j; \\ 0.1, & \text{otherwise.} \end{cases}$$

4. Compute the reinforcement signal $r = f(i,o)$.
5. Generate target output values as follows:

$$t_j = \begin{cases} o_j, & \text{if } r > 0; \\ 1 - o_j, & \text{otherwise.} \end{cases}$$

6. Generate output error values e_j as follows:

$$e_j = (t_j - s_j)s_j(1 - s_j)$$

7. Backpropagate errors.
8. Select learning rate as follows ($\mu_r > \mu_p$; $\mu_r, \mu_p > 0$):

$$\eta = \begin{cases} \eta_r \text{ if } r > 0; \\ \eta_p \text{ otherwise.} \end{cases}$$

9. Update output layer and hidden layer weights.
10. Go to #2.

Figure 12: : CRBP Algorithm (adapted from Ackley and Littman , 1990).

concurrently. In the meantime, some means of selecting the appropriate, if inchoate, competence must exist. The problem of selecting the right fully developed or implemented action, or competence, is known as *action selection* in ethology. In behaviour-based control, as noted in subsection 2.4.5 of this thesis, the usual choice of a fixed arbitration scheme leads to difficulties of cyclical behaviour and, ultimately, of coherence as complexity grows. In the context of modular reinforcement learning, it takes on a somewhat different aspect. A means of directing reinforcement to the appropriate module is required, because the actions of one module may result in a situation that indicates another module should receive reinforcement on the next cycle. In this initial architecture, the approach was hierarchical, as in subsumption (explicitly) and pre-emption (implicitly). The algorithm in Figure 12 was devised for this purpose. As in subsumption and preemption, the organisation is implicitly hierarchical, reflecting some top-down constraints. Similarly, layers are designed to be concurrently active (in the sense that they all sense the environment and have the capacity to act on the same time cycle). This is indicated in Figure 11 by the stepped arrangement of the schematic modules. Clearly, these constraints derive from the designer's domain ontology, but the introduction of bottom-up, trial and error learning at the module level at least reduces the designer's influence at that level.

Control algorithm

As in Addam, each module in the architecture was intended to receive input from all the simulated robot's sensors, but the control algorithm ("Algorithmic selector" in Figure 11) was designed to encourage each module to specialise in a particular competence. To achieve this, it must solve the problem of structural credit assignment at the modular level. It does so by ensuring that the module responsible for an action that gives rise to punishment will be rewarded if its next proposed action is beneficial

to the agent. Furthermore, it is designed to encourage progress towards an overall goal: if on the previous time step the reinforcement signal was a reward, the output from the default (top-level) module will be selected to drive the simulated robot. Accordingly, if the robot avoids punishment, it should tend to fulfil its overall goal (for example, light-seeking as in the studies described later). Reinforcement signals (*r* in Figure 12) are generated by the Critic (Figure 11) for each sensor-related, behaviour-defining situation that gives rise to the reward or punishment. For example, in the studies described later in this chapter these situations would be identified as follows:

- moving further away from (punish), or closer to (reward), the light source as indicated by the simulated light sensor readings on successive steps;

- moving closer to (punish), or further away from (reward) a wall, within a predetermined threshold, as measured by the simulated range-finder readings;

- colliding with an obstacle or a wall (punish), or moving away from a previous collision (reward) as indicated by the simulated touch-sensors;

The Critic monitors the state of the simulated robot's sensors on each control tick and has access to a one-step memory of the previous state. It was implemented as a function containing a case structure. Each case contains an implementation of a reinforcement policy specific to each sensor type and the function is called by the control algorithm for every type on each tick. The control algorithm is designed to first determine whether the agent was punished on the previous control tick and determine the case that gave rise to the signal. If so, it will select the associated

module's output to drive the robot and save that module's identifier. Otherwise, it selects the top-level module's output. If the reinforcement signal generated by the movement of the robot on the next tick is "punishment", only the module corresponding to the saved identifier will be backpropagated. Otherwise, all modules will be backpropagated.

CRBP learning algorithm

The CRBP learning algorithm used in these studies is summarised in Figure 12, to which the following remarks refer. The algorithm differs somewhat from Ackley and Littmans' (1990), having been modified for application in real-time, as in Meeden (1993) and Meeden et al (1994). Additionally, note that it is unrealistic to force convergence of output values on 1.0 or 0.0 (recall that the logistic activation function commonly used in backpropagation, as here, returns values that *asymptotically* approach 0 or 1). To avoid this, an approximation to a Bernoulli trial was devised (refer to step 2). This avoids "stretching" the probabilities, as in the original algorithm but the effect is the same. It aims to achieve some balance between exploration and exploitation of the input space by allowing some behavioural variety while encouraging the identification of actions that lead to the desired behaviour. The term at step 5 assumes that the logistic function (equation 6.1, omitting subscripts) is used as the network activation function, the term sj(1 - s_j) being its first differential:

$$f(s) = \frac{1}{1+e^{-s}} \qquad (6.1)$$

At step 7 the idea of differential learning rates for reward and punishment should be noted. Ackley and Littman found this to be an advantage and Meeden continued the

practice. Unfortunately, it increases the set of "magic numbers" that have to be set by hand. Attempts were made to set these learning rates dynamically (desirable from practical (project-level) and foundational (programmatic) perspectives) but sadly, this attempt to remove a source of designer-bias did not prove successful.

Modular neural nets

In addition to demonstrating the effectiveness of reinforcement learning in a modular approach, a subsidiary aim was to investigate different types of neural network as components of the modular architecture, to the extent that these would be compatible with CRBP. Precise details of topologies and parameters will be given under the individual studies. The general types investigated and reasons for doing so where as follows:

- Feedforward networks with hidden layer, also called multi-layer perceptrons (MLP), these are now the most common networks in connectionist research and, following the example of Addam, were expected to be effective as Crill modules.
- Feedforward network with no hidden layers (i.e. the simple or single-layer perceptron (SLP) as used by Nehmzow (1990).
- As above, but with one-to-one connections from the motor output units to corresponding extra units in the input layer (corresponding to the "motor" sensors described by Meeden et al., 1994).
- Partial recurrent neural network with one hidden layer (the SRN).

```
IF reinforcement_signal[1][time step t-1] =  PUNISH
    send output of module[1] to robot
    last_action = 1
ELSE IF reinforcement_signal[2] [time step t-1] = PUNISH
     send output of module[2] to robot
     last_action = 2
ELSE
      send output of module[3] to robot
      last_action = 3
 move robot
IF reinforcement_signal[last_action][time_step t] = PUNISH)
   FOR i = 1 TO NUMBER_OF_MODULES
        backpropagate(module[i])
ELSE
     backpropagate(module[last_action])
```

Figure 13: The Crill control algorithm.

6.3 Studies of simulated adaptive behaviour using the Crill architecture

Inspired principally and broadly by the Addam simulation, the aims of the studies in this chapter were:

- to determine whether the Crill architecture could accomplish a similar kind of overall goal by driving a similar simulated robot in a comparable environment;
- to work towards an alternative modular solution to the same action-selection problems that beset subsumption-style approaches by setting tasks that require coordination of multiple behaviours;
- and, on another level, to lever in the foundational advantages of an approach (in CRBP) less subject to designer bias than supervised learning.

6.3.1 Behaviours and related sensors

The behaviours investigated in these studies were similar to those described by other researchers whose work was included in the review in Chapter 4. Each of the behaviours is based on a single sensory modality, as described below.

Light-seeking

The behaviour of light-seeking, or light-following, has been investigated both in simulation and with a real robot by Meeden (1993, where it was described as "light-

as-food"); as a component of more complex overall behaviour, by Meeden et al. (1994); and in simulation by Ziemke (1996a). In these experiments it appears that the reinforcement function typically compares a real or simulated light sensor reading obtained at time step t with the reading obtained at time step t -1. The reinforcement-learning agent is rewarded if the reading shows an increase; otherwise, it is punished.

Contact-based obstacle avoidance

The use of tactile sensors, whiskers or bumpers is quite common in mobile robot research. They can be seen as a means of ensuring that the robot does not damage itself or its environment if other means of detecting walls and obstacles fail. However, they have also been used as the basis for navigation experiments by Nehmzow (1995, using real whiskers) and by Saunders, et al. (1994, using simulated bump sensors very similar to the ones implemented in IMRANNS for these studies).

Range-based obstacle avoidance

Another simple behaviour commonly the subject of adaptive autonomous control experiments is range-based obstacle avoidance using either infrared sensors or active sonar as range finders. The learning method is usually to set an arbitrary threshold in the detected range as the behavioural trigger. The agent is trained to perform some manoeuvre that will avert the collision that would have occurred if the previous behaviour (perhaps light-seeking or simply wandering) had continued.

6.3.2 Simulated mobile robot environment

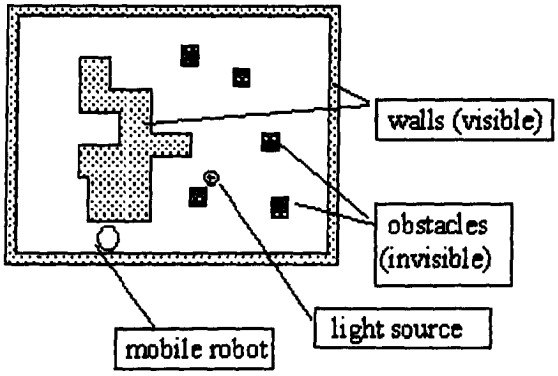

Figure 14: Screen shot of environment for Crill–based simulation of adaptive behaviour studies with call-outs added.

A simulated environment was designed for the studies to support the desired concurrent learning of the multiple behaviour described above (see Figure 14). There was no intention to model any particular real-world environment. Indeed, it was similar in outline to the imaginative environment used for Addam - but the simulation had features that would be more possible to replicated physically (obstacles rather than transparent "ice", walls rather than "blocks", light sources rather than "food").

The environment was two dimensional in appearance, intended to represent an enclosure surrounded by walls in plan view. The enclosure contained a large, irregular shape, intended to represent a low building or structure (clearly visible towards one side of the enclosure in Figure 14). This structure and the walls were implemented so that they could be detected by both the robot's simulated tactile and sonar sensors. A number of square obstacles were scattered about the open area and these were implemented so that only the bump sensors would be able to detect them.

They represent objects much lower in height than the walls – so low that the robot's srange finder beams would pass over them. They were intended to be equivalent to the Addam simulator's transparent objects through which Addam's so-called sonar could "see". Light-sources were implemented so that they could be placed anywhere in the open areas. They would be "visible" to the robot's light-sensors from all points in the enclosure, as if suspended above the height of both the structure and the obstacles. These were intended to parallel the odoriferous "food" sources detectable by Addam's so-called olfactory sensors. In a similar way, the Crill robot would be able to "sense" the lights sources, and move towards them (cf. Meeden, et al., 1994 who also describe a light-seeking task as "light as food"). These arrangements were made so that tasks similar to those set for Addam could be devised with similar goals and expectations.

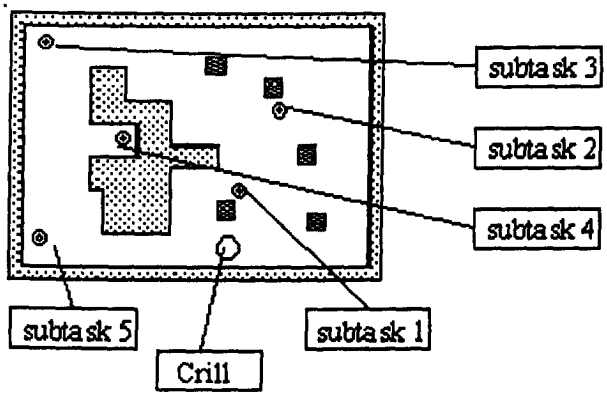

Figure 15: Arrangement of light sources for the Crill based studies of simulated adaptive behaviour. .

For the studies discussed in this chapter, the environment was set up as in Figure 15. Just as Addam consumed a series of food items, the idea was for the simulated robot to seek a series of light sources. Similarly, as the "odour" of a food source would

disappear once "consumed" a light source would be switched off when the robot made contact with it and the next in the series would be switched on. The extended behaviour, required to achieve the overall goal, would appear to be sequential to the observer. Hoever, according to to Colombetti and Dorigo (1994), as this would be determined by the dynamics of the agent's interaction with the environment rather than the internal dynamics of the robot's controller, it should be classified as reactive navigation. The light sources were so positioned, in relation to the configuration of obstacles, building and walls, as to require the flexible co-ordination of behaviours in order to achieve the overall goal. For example (Figure 15), the first light (subtask 1) was positioned so that the robot would have to navigate around an obstacle to reach the light source. The fourth light-source (subtask 4) was placed at the far end of a narrow blind alley.

6.3.3 Simulated robot details

The simulated robot for these studies was equipped as:

- 12 sensors, consisting of four of each of the three types described in section 5.3.1, light-sensors, range finders and bump-sensors.
- 3 binary motor inputs encoding the 8 (2^3) fixed directions of movement possible under the first of the schemes described in section 5.3.1.

Recall that the chosen motor scheme ensures that the binary complement of a given motor control vector will head the robot in the opposite direction thus giving the CRBP approach a meaningful interpretation at the motor level. For example, if on the

previous step the robot was heading north (motor-control binary code 101), and received punishment, the algorithm should tend to push the network responsible in the direction of binary output 010 (south).

6.3.4 Module details

Four distinct types of feedforward or partial recurrent ANN were discussed in section 6.2.2 as possible ways of implementing the sensorimotor modules of the Crill architecture. It was therefore decided to conduct studies with different instantiations of the architecture using each type. The same environment was used for each study, as described above.

For each instantiation of the architecture, the sensorimotor modules were uniformly implemented as shown in Table 1.

Table 1:Implementation details of Crill architecture.

Architecture	Crill					
Version	1			2		
Number of modules	3 (all uniform)			3 (all uniform)		
Net type	MLP			SRN		
Layer	Input	Hidden	Output	Input	Hidden	Output
Units	12	4	3	12 + 4 context	4	3
Connections (internal)		Fully with input units	Fully with hidden units	Context one-to-one with hidden units (feedback)	Fully with input units (including context)	Fully with hidden units
Connections (external)	One-to-one with sensors		One-to-one with motor units	One-to-one with sensors		One-to-one with motor units
Activation Function		Logistic	Logistic		Logistic	Logistic

Table 1(cont.)

Architecture	Crill (cont.)			
Version	3		4	
Number of modules	3 (all uniform)		3 (all uniform)	
Net type	SLP		SLP	
Layer	Input	Output	Input	Output
Units	12	3	12 + 3 output feedback units	3
Connections (internal)		Fully with input units	Output feedback one-to-one with output units (feedback)	Fully with input and output feedback units units (feedforward)
Connections (external)	One-to-one with sensors	One-to-one with motor units	One-to-one with sensors	One-to-one with motor units
Activation Function		Logistic		Logistic

6.3.5 Training details

For each of the four instantiations of the architecture the simulation was run 12 times.

To enable meaningful comparisons to be made across the studies, the same training procedure was used for every run:

- all weights were randomly initialised in the range [-0.1, 0.1] ;
- for rewarded actions a learning rate of 0.3 was used;
- for punished actions a learning rate of 0.1 was used;
- no momentum was used;
- no bias units were employed.

6.3.6 Results

The metric used to compare the performance of the different instantiations of the Crill architecture was the number of time steps taken by the simulated robot to achieve the overall goal. This was defined as extinguishing the final light source after successfully extinguishing all the preceding ones in the sequence. For this comparison, the arithmetic mean of the successful runs for each type was chosen, any unsuccessful runs being discarded. An unsuccessful run was defined as one in which the run was terminated because the simulated robot did not appear to be making any progress after a long period of observation. Standard deviations from the mean were also calculated to indicate the reliability of the metric for each set of runs. These results are summarised in Table 2. The table indicates that the instantiation of the Crill architecture based on feedforward networks with a single hidden layer (MLP) performed the overall task in significantly fewer steps than any of the other versions. The instantiation based on SRNs was in turn significantly better on this measure than either of the remaining two versions.

Table 2: Comparative performance of different instantiations of the Crill architecture.

Neural net type	Percentage runs completed	Arithmetic mean of steps taken in successful runs	Standard deviation from the sample mean
MLP	100%	1745.33	75.07
SRN	100%	3378.83	108.82
SLP with output layer feedback	100%	5800.16	265.86
SLP	75%	13042.44	3773.70

Of these, the instantiation based on networks with no hidden layer (SLP with feedback from the motor outputs) was very significantly better on the same measure than the version with no hidden layer and no recurrent connections (SLP). The inconsistent and generally inferior performance of the last version perhaps conforms most clearly to expectations. The deficiencies of this kind of network in static problem domains are well known, while Nehmzow's success with them (see section 4.4) clearly depended on a formulation of the problem space that explicitly avoided any linear separability issues (Nehmzow, 1992). In relation to this, the performance of the version with feedback from the motor units (output layer) requires explanation. Recall that Meeden et al. (1994) found that feedforward networks having this modification (which they characterised as a surrogate "motor sensor") generally performed better than ones not having it did. To some extent, then, it appears that this kind of motor information if available to the controller can compensate for the absence of a hidden layer. The addition of a hidden layer with recurrent connections to the input layer (an SRN) however appears to provide a significantly superior configuration. The explanation for this may relate to the observation of Cottrell and Sung (1991) that networks relying on feedback from just the output layer cannot remember information not directly exploited in their output. However, even though the addition of feedback in some form at the sensorimotor level appears to confer relative advantages, the significantly superior performance of the MLP instantiation to even the best of these suggests that – at this level and in the context of a higher level action selection scheme – hidden layer information is sufficient. These findings tend to support the evidence of Saunders et al. (1994), though they did not investigate feedback connections at all.

6.4 Concluding observations

The studies in the simulation of adaptive behaviour presented above indicated that a reinforcement learning approach could be applied to learning reactive navigation control tasks in a modular architecture. According to Maes (1995) and Saunders et al. (1994) subsumption agent can succumb to some kinds of cyclical behaviour. Like Addam, based on supervised learning, the simulated robot, controlled by different instantiations of the Crill architecture, based on reinforcement learning, were usually able to avoid this problem. This suggests that the well-known advantages of problem decomposition afforded by modular architectures can be extended to the problem of learning relatively complex, multi-behavioural goals through trial and error, though admittedly in a somewhat restricted sense. In this context of reinforcement learning, the Crill architecture addressed the particular,structural credit-assignment problem that results from modularisation whereby the high-level scheme of reinforcement and action selection encourages specialised competences to develop in each neural net module. However, this top-level arbitration scheme is algorithmic and hence inflexible. Foundationally, it represents another case of designer's domain ontology intrusion. Moreover, it is less satisfactory, at this level of description, than the preemption mechanism whereby modules are able to assume control when necessary (whether or not the philosophical notion of leaky levels is accepted). This is not to commit, however, to the notion (almost a doctrine!) of completely de-centralised control underlying the subsumption architecture. At the level of cognitive modelling, it is hard to avoid the idea that modularity must be subject to some overall control. Although the aim in the present studies is to build a relatively simple agent from the

bottom up, the very long-term aim would be that such agents could be developed to exhibit some cognitive behaviour.

These reflections led to the idea that the high-level arbitration algorithm should be replaced with a module, also based on ANNs, that would be able to co-ordinate the actions of the lower level modules. That this is a temporally extended problem was apparent from the nature of the existing Crill algorithm. The conventional panel of experts approach (e.g. Jacobs, Jordan, Hinton and Nowlan, 1991) at first sight seemed unsuitable because it is based on supervised learning and addresses only non-temporal problems. The next chapter describes the adaptation and development of this approach to the modular reinforcement-learning problem just described.

6.5 Summary

In this chapter, the nature of the behaviours under investigation was analysed with reference to other work in the research community, and the initial experiments were described. Experiment results were presented and some comparisons were drawn with research antecedents to demonstrate some advantages of the novel approach. Weaknesses of approach, too, were recognised and conclusions were given, indicating the need for the new approach described in the next chapter.

CHAPTER 7

ARCHITECTURES AND STUDIES (II)

UNIFYING COMPETENCE AND CONTROL LEARNING

7.1 Introduction

The main part of this chapter describes how the architecture described in the previous chapter was incrementally developed to achieve the primary objective of the first phase of these enquiries: a unified approach that integrates learning of competences and control. It begins with a broad presentation of rationale, analysing the nature of the control problem in greater depth. In the second part, further studies of simulated adaptive behaviour are described in which different instantiations of the new architecture are used to control a simulated mobile robot to perform the same set of tasks described in Chapter 6. Some comparative results and observations are then presented.

7.2 The recurrent mixture of experts control architecture

The previous chapter introduced an approach to reinforcement learning with immediate or local reward, in a modular control architecture. It was shown that such an architecture could enable a simulated mobile robot to perform adaptive control tasks (essentially reactive navigation) comparable to a similar agent trained using supervised learning techniques. Although this ameliorated *some* foundational

concerns, a serious weakness in the approach remained. This was identified as the top-level control algorithm responsible for co-ordinating the competence modules. The algorithm was another manifestation of the top-down constraints that determine the function of each layer in the behaviour-based approach to control. That these constraints originate in observer space and are therefore a consequence of the designer's domain ontology, rather than the agent's was discussed in subsection 2.4.5 and section 3.4. The full foundational implications of this problem were not realised until much later and they are not discussed until Chapter 8.

At the stage of the investigations under discussion, it was recognised that within the reinforcement learning paradigm the presence of some top down constraints was inevitable. At times, these may assume other guises, for example, as internal drives or reflexes (Gaussier and Zrehen, 1994). The fact that quite complex constraints, goals, drives, and so on, could be compiled down to simple, scalar signals lent hope to the long term program of constructing an artificial mentality from numerous interacting subsystems. However, in the present studies, the aim was to advance cautiously, building on known mechanisms and familiar resources wherever possible. A sketch of how the modular neural network approach to problem solving in static domains, known as the *mixture of experts* (ME), might be adapted for the control of autonomous robots was noted in subsection 4.4 of this thesis (Franchi, Morasso and Vercelli, 1994). The nature of the action selection control algorithm described in Chapter 6 suggested however that a gating network of this kind (described originally by Jacobs et al, 1991) would not be adequate. The nature of the control problem in terms of its multi-modal sensory input space suggested the overall appropriateness of a mixture of experts approach. However, the temporally extended nature of the credit

and blame assignment problem at the level of action selection seemed to require a temporal processing approach. The broad idea of using some variety of partial recurrent networks as the gating network was therefore at least superficially attractive but needed further working out.

7.2.1 The mixture of experts approach in static problem domains

The rationale for the proposed new architecture now turns for the moment from a consideration of the temporal aspects to describe the ME approach to problem solving in static domains (that is, those typical of traditional connectionism). It was devised originally by Jacobs et al. (1991) to address the problem of local discontinuities in the input space of a particular global problem. It has also been presented as a general approach that confers the well-known advantages of problem decomposition. Although monolithic networks can generally solve such problems, interference effects reduce their efficiency. These arise because the error attributable to a particular training example is backpropagated globally, thus tending to undermine localised configurations forming in the solution space. In the ME architecture, error is assigned and backpropagated locally according to an individual network's contribution to the solution on a given training step. The approach assumes that the data available to the system can be represented as a collection of different functions, or, more properly, probabilisitc relations, each defined over a relatively local partition of the input space. The general idea is indicated in Figure 16. *N* feedforward networks, perhaps having different internal characteristics but with a uniform number of inputs and outputs, see

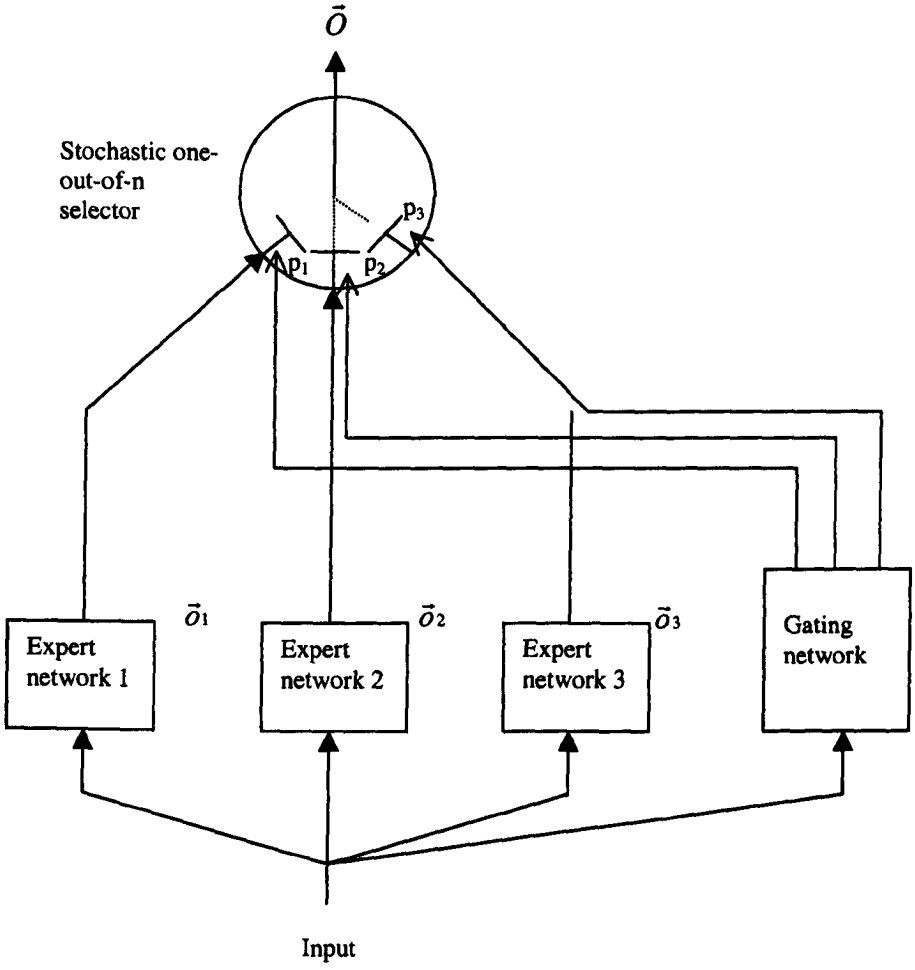

Figure 16: Mixture of Experts architecture re-drawn from Jacobs, et al. (1991).

the whole input space. All the networks produce outputs *in parallel* (that is they are all exercised on a single training / test cycle) under a supervised learning regime; otherwise in the usual way.

An auxiliary feedforward network with N output units, each paired with an individual network, also sees the whole input space and produces output on the same cycle. The output of this network in effect mixes the output of each of the other networks to produce the overall output of the system. It can be considered to do this probabilistically: if each of its outputs is somehow constrained to be in the range [0,1]

and all of them sum to 1, then the output functions as a probabilistic predictor or selector. Further, if each output vector from each of the N nets is compared with the target vector, and the error to be propagated is multiplied by the output of the corresponding auxiliary network output line, the amount of error backpropagated should be localised. The interference effects noted above can therefore be reduced (cf. Sharkey et al. 1996). As all the networks are taught in relation to the same training example on each cycle, over time the auxiliary network should choose the most appropriate mixture of networks to solve the overall problem. The error backpropagated through the auxiliary network of course relates to the correctness of its probabilistic suggestions. As there is no *a priori* training set for these outputs, the learning at this level is, in a sense, trial and error, though the error signal does derive from explicit training examples for the overall output of the system. The probabilistic network is usually called a *gating network* and the networks it controls are called *experts.* The latter term is, of course, strictly only appropriate once they have been successfully trained hence the term sometimes used in these studies, *inchoate experts.* The idea is that they will become experts or specialists on a particular part of the problem space or a specific task.

7.2.2 Giving the mixture of experts architecture a short-term memory

The ME model discussed in the previous section possessed features that recommended it as a possible alternative to the fixed modular scheme (so-called Crill) described in Chapter 6. The idea of networks specialising in more-or-less local areas of a flexibly partitioned input space was one that appeared to map satisfactorily onto

the adaptive autonomous agent problem. Another attractive idea was that an auxiliary network could learn how to assign ANN modules to different subtasks. This would support with the move away (in these studies at least) from the behaviour-based doctrine of totally decentralised control towards a model of more cognitive behaviour that would ultimately supply its own top-down constraints. It scarcely needs repeating that no detailed biological model is implied here. Clearly, the gating network is no cerebral neocortex (cf. Aitken, 1994, discussed in subsection 4.5 of this thesis), but at a much higher level of description, the parallel can be drawn. However, there were two obvious immediate problems:

- the ME architecture was closely associated with supervised learning schemes.
- the new class of problems had temporally extended characteristics that would pose an unfamiliar *control* problem for the gating network, quite different from its role in static domains.

The first of these problems could be answered in principle as follows. The mixture of experts architecture could be trained by gradient descent methods, even if this was not the most efficient approach. Thus it was possible to consider using CRPB in place of conventional backpropagation, at least for some initial experiments to establish the broad feasibility of the approach. The second issue to be considered was more fundamental, involving some reflections and considerations that would lead to the preoccupation with matters temporal that dominate the last chapter of this thesis. At this stage, the considerations were mainly project-level, but their potential foundational significance needs to be borne in mind. To encourage this, it is sufficient for now to quote the assertion, "*The overriding task of Mind is to produce the next action*" (Franklin, 1995 – his italics). The resonance of this phrase suggests the oneness of some essential project-level and programmatic AI concerns. Certainly, the

problem being faced in these studies was how a gating network could be modified to produce the next system action through adaptive learning. Recurrent networks were discussed in section 3.3. Here, it is worth briefly recalling some examples of SAB research reviewed in Chapter 4. This will highlight their findings concerning this class of neural network, before moving on to an explanation of why and how they were incorporated into an ME style architecture in the furtherance of these studies.

Recall that Meeden et al. (1994) used the SRN network quite extensively in their experiments. However, the statistical results were not very conclusive. Pal and Kar (1996) showed that some mobile robot navigation problem not solvable using a reactive approach, could be solved using a short-term memory. This was provided by recurrent connections in an ANN-based controller, along the lines of the Jordan network. Tani (1996) used another Jordan-like network to train a mobile robot off-line to achieve a form of model-based learning of a navigation task, and offered an analysis of its behaviour in dynamical systems terms. Ziemke (1996a) investigated Jordan and Elman networks for the control of a mobile robot in sequential decision tasks. He found that their performance was inferior to another partial recurrent network, a second-order network, but the basis of comparison may well have been skewed in favour of the latter. Some of the implications of the later work will be examined and investigated in later chapters of this thesis. Only the first of this series of results was available in the formative stages of these studies. However, clearly there is a growing interest in the potential of this kind of network as a means of providing an STM to enable adaptive autonomous agents to go beyond merely reactive behaviour. This is true both at the project-level (Pal and Kar, Tani) and at the programmatic level (Meeden, Ziemke, Tani). All the examples cited employed

monolithic architectures, though only Ziemke has argued overtly against the principle of modularity. To cut from the chase briefly, it will be worth looking at this argument and putting the opposing view at this point.

Ziemke's (1996a) concern was that to predetermine the gross structure of autonomous agent controllers, on the basis of the designer's decomposition of a particular problem, does not help robotic agents to formulate their own principles of behaviour. Even so, to insist on "generic" structuring of a monolithic neural network during task performance may be an extreme position. If such structural learning is indeed generic and not ad hoc then abstraction and re-use of its defining features ought to be possible, but there is no indication that this is so. Moreover, although task-based modularisation in the style of software engineering presumably has no biological parallel, neurophysiological evidence supports some kind of modularity, conforming broadly to boundaries between sensory modalities, specialised motor functions and separable cognitive functions. Such areas of "expertise" in the human brain have of course arisen phylogenetically. It therefore seems more "natural" to define similar areas in our agents than to expect them to emerge ontogentically (typically, during the performance of a particular task). In the context of an agenda for a new AI based on connectionism, Dorffner (1997) too expressed the view that modularity in some form will be necessary. This was for the more pragmatic reason of confronting the problem of scaling SAB-style agents to more realistic and useful task performance.

To conclude, the SRN network was chosen as the required short-term memory, based on its in-principle power and evidence from the comparative studies showing its in-practice advantages, particularly over Jordan-style networks. It was intended that this

would enable the ME model to be adapted for the control of temporally extended behaviour. The generic, so-called recurrent mixture of experts control model is shown in Figure 17. To avoid confusion with the later hierarchical RME of Tani and Nolfi (1998) it will subsequently be referred to as MERGe (**M**ixture of **E**xperts with **R**ecurrent **G**ating **n**etwork).

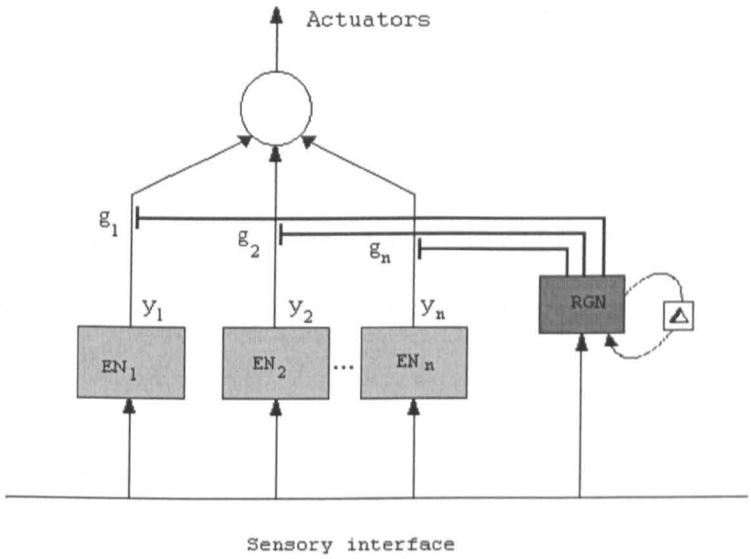

Figure 17: Schematic of a recurrent mixture of experts architecture first described in Rylatt, Czarnecki and Routen, (1996). (EN = expert network, RGN = recurrent neural network, the ρ symbol represents a time delay of 1 step, following the convention in Lin al. (1996) - in this case the hidden layer in the RGN is copied to the input layer.

7.2.3 The adaptive autonomous agent problem revisited

In the previous chapter, the nature of the problem was presented at an intuitive level typical of much similar work reported in the literature. In order to show how the two broad groups of ideas discussed above (recurrent neural networks and the ME architecture) bear on this problem, deeper analysis is required. A good starting point is Kaelbling's (1993) model of what she calls an *embedded system,* typically a mobile

robotic agent. In this model, the agent moves around and so affects the world in which it is situated. Even if nothing else moves or changes in this world, the agent's own movements will affect its input stream. It is therefore always appropriate to regard the agent / world system as a dynamical system. Beer (1995) provides a sketch of dynamical systems theory to support this view. Recall too, that one of the foundational aims to emerge in the investigations was to minimise the influence of the designer's domain ontology on what the agent perceives (given the inherent limitations of its sensors), and what the agent decides to do (similarly, given the limitations of its effectors). The agent's knowledge of the world must therefore be derived mainly from this raw input stream, together with some compiled designer's domain knowledge in the form of scalar reinforcement signals.

From the agent's perspective, the combination of input stream and reinforcement signal represents the state of the world. In this conceptualisation, the world undergoes a series of state transitions caused by the actions of the agent, and can be modelled as some kind of finite-state automaton. However, this commits to an essentially computational account of the agent's behaviour. A typical reinforcement learning approach would interpret the actions of the agent as the cause of state transitions in the world. Commonly, the agent itself is regarded as having no internal state. In this case, the agent's next action is causally dependent only upon its current input. This view corresponds to a purely feedforward neural network-based agent like Addam (section 4.2 of this thesis). Once trained, such networks have nothing that can be regarded as internal state: the weights are fixed and the activations are lost between processing cycles. They perform a static mapping from inputs to outputs that implement a reactive, or stimulus-response behaviour pattern. Alternatively, the agent

can be considered to have internal state. In this case, both the agent's current inputs and its last state may causally determine its next action. The simplest example of an agent with internal state would be a binary switch between input and output. The agent's actions would differ according to the last setting of the switch and its current inputs on each cycle (Colombetti and Dorigo, 1994). Internal state of a more complex kind is represented by the feedback layer activations of a recurrent neural network.

The question of exactly what constitutes state in such networks is not always obvious. In the SRN network, clearly the hidden layer activations represent state. In the Jordan network, the state is represented by the output layer activations. The inappropriateness of Jordan-style representation in some situations can be readily appreciated by considering the case of a network with multiple hidden units and a single outpu. Here, the output unit activation is an inadequate representation of the network's state. The SRN network on the other hand, provides a potentially richer representation of state by storing the hidden layer activations, but the state is *hidden.* This network must simultaneously form internal task-relevant representations of the current raw input pattern and of the temporal history. Elman (1995) makes the point that these encodings, because they are produced by the underlying feedforward network architecture, will tend to be highly abstract in nature. Accordingly, like purely feedforward network hidden layer representations, they will have a functional, rather than pattern-based, similarity structure. Thus, instead of being like a tape recording, the history representation should enable such a network to make relatively flexible and sophisticated use of the temporal information.

Now recall that the ME architecture was intended for static domains. It tries to allocate a specific expert network to a subset of the inputs, or to a subspace of the input space. In this sense, it maps quite well onto the static aspects of the adaptive autonomous agent problem of how to learn appropriate responses to multi-sensory stimuli. Clearly, at this level, outputs are a function of inputs, but the nature of the sensory data means that there will be discontinuities in the global function being learned. It therefore makes sense to fit a number of models on each side of the discontinuities rather than try to fit a single model across them. At this level, a computational interpretation of the processes involved may be the right one (see for example Beer, 1995). This may justify using purely feedforward networks as the "sensory experts".

It may be that feedforward connections are typical at the immediate sensory levels in biological neural networks , while recurrent connections are common at cognitive levels. If so, a dynamical systems account of the latter may be appropriate. Of course, this somewhat speculative justification for considering the use of recurrent connections at some higher level of control (in this case the gating network) needed more working out. Consideration of the aforementioned finite state model of the agent and its environment helped to reveal the relevant aspects of the problem more clearly. On each cycle, the agent perceives the state of the world through the vector of sensory inputs and the reinforcement signals. If the correctness of the outputs depends on appropriate structural credit assignment at the modular level, the problem for the gating network is to assign the appropriate sensory expert to a presented input vector. The nature of the cluttered world faced by the agent and the differing sensory manifold it presents, together with the task-related reinforcement signals, suggested a

dynamic soft switching process hidden in the world. At this level of description, it was possible to think of a hidden Markov model in which the next state is determined by the last state and the last inputs and outputs. The task of the gating network could therefore be recast as the modelling of this process; consequently, it was reasonable to infer that internal state is required in the model. The subtleties of temporal representation discussed above in the Elman network suggested it could be a suitable model both for this and for tasks that are more complex. As the hierarchical control task only appears to require a memory of the immediately preceding state, it could be argued that a simpler approach might have been adequate. However, the prospect of an architecture that would perhaps be capable of complex tasks involving more subtle temporal relationships was regarded as sufficient justification for taking a longer view. For example, it is hard to see how the non-connectionist example of an agent with internal state provided by Colombetti and Dorigo (1994) could be developed significantly in this way.

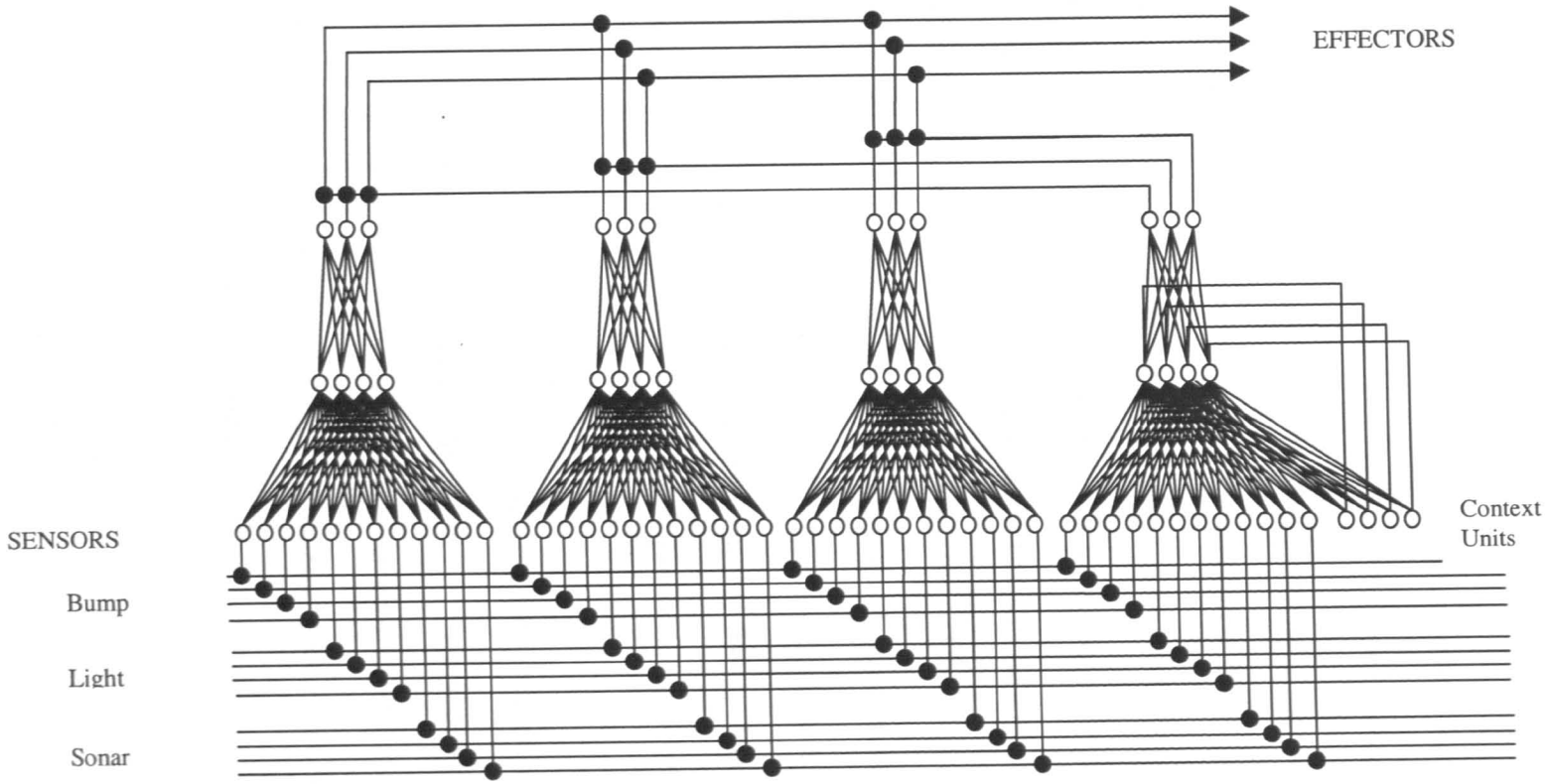

Figure 18: Detailed diagram of MERGe architecture (Version 1).

7.2.4 The new architecture in detail

The broad rationale for the new architecture and its general outline should now be clear. Before the control experiments performed with it are described, some details of topology, connectivity and training method are set out in this section. The first version of the architecture is shown in the detailed diagram in Figure 18. This shows how the sensors are connected at the input layer and gives a clear illustration of the gating network output connections to the expert network outputs. In conjunction with the block diagram of the conventional mixture of experts network in Figure 17, this should help to clarify the nature of the processes by which the overall output of the control network is reached. However, the nature of the control algorithm still needs some explanation. Recall that in section 7.2 the potential for retaining CRBP was mentioned. Clearly, however, some amendment to the algorithm was necessary so that it would conform to the somewhat different problem posed by the new control layer.

In its simplest form (for example, Jacobs, et al., 1991) the gating network uses a normalising activation function at the output layer to make its choice of the most suitable expert on a particular training case (in supervised learning). This function has the form shown in equation 7.1, where g_k is the k^{th} gating network output and net_k is the k^{th} weighted sum of all the connected units on the gated network layer beneath:

$$g_k = \frac{\exp(net_k)}{\sum_R \exp(net_k)} \qquad (7.1)$$

1. Construct R feedforward backpropagation networks E_q each with an input layer x with ($1..i..N$) sensor units, hidden layer h and output layer s with ($1..k..M$) units (the expert nets).
2. Construct recurrent backpropagation network G with input layer with ($1..k..M$) sensor units x, L context units c, hidden layer h with ($1..j..L$) units and output layer g with ($1..k..R$) units (the gating net)
3. Collect continuous sensory input vector $\mathbf{X}$ of dimensionality N.
4. For each E_q and for G, copy $\mathbf{X}$ to x
5. Forward propagate through each E_q to produce search vector of the continuous values $\mathbf{s}$.
6. Forward propagate through G to produce the softmax output vector $\mathbf{g}$ (equation 4.2)
7. For each E_q multiply outputs s_1 to s_m by g_q
8. For $k = 1$ to M sum s_k in E_1 to E_r to give system search vector $\mathbf{S}$ of dimensionality M.
9. Generate a binary output vector $\mathbf{O}$ of dimensionality M as follows. Given uniform continuous random numbers ξ in the range (0.1,0.9),

$$\mathbf{O}_k = \begin{cases} 0.9, & \text{if } \xi \le \mathbf{S}_k; \\ 0.1, & \text{otherwise.} \xi \end{cases}$$

10. Compute the reinforcement signal $r = f(\mathbf{X},\mathbf{O})$.
11. Generate target output values as follows:

$$t_k = \begin{cases} \mathbf{O}_k, & \text{if } r > 0; \\ 1 - \mathbf{O}_k, & \text{otherwise.} \end{cases}$$

12. For each E_q generate output error values e_k as follows:

$$e_k = (t_k - s_k)s_k(1 - s_k)$$

13. Generate output error values e_k for G as in equations 5.2 and 5.3 (see text).
14. Backpropagate errors through each E_q.
15. Backpropagate errors through G.
16. Select learning rate as follows:

$$\eta = \begin{cases} \eta_+ & \text{if } r > 0; \\ \eta_- & \text{otherwise.} \end{cases}$$

17. Update weights in each E_q and in G.
18. Copy h^G to c^G
19. Go to #3.

Figure 19: CRBP Algorithm for MERGe architecture.

This activation function encourages convergence of the gating network to an output sequence in which each output vector has a single output with value one and the rest are zero. In other words it competitively selects an expert to specialise in each training case. The reason for this is revealed by the first differential of this function with respect to the error (omitting inessential subscripts):

$$f'(net_k) = \frac{\exp(g_k)[\sum_R \exp(g_k)] - [\exp(g_k)]^2}{[\sum_R \exp(g_k)]^2} \tag{7.2}$$

Because the (squared) absolute sum of the error from each expert network is backpropagated through the gating network, equation 7.2 has the effect of selecting a single expert at the expense of the others. It does so by encouraging error reduction in that expert and not in its competitors, for example:

$$\frac{\partial e_q}{\partial o_q} = g_q \delta^2 \tag{7.3}$$

In equation 7.3, δ is the error vector and e_q is the error at the output layer of expert q. It remains to be explained how this idea was adapted to serve under a CRBP learning regime.

Recall the original CRBP algorithm shown in Figure 11, Chapter 6. The algorithm for the novel MERGe architecture is shown in Figure 19. A number of questions had to be addressed in adapting the basic ME idea for use as a continuous feedback, closed loop controller (the points are numbered below for subsequent reference):

1) How should the output of the system be generated?

2) How should the error of the individual experts be calculated for backpropagation?

3) What form should the short-term memory take?

4) What algorithm should be used to determine system state?

These questions fall into two groups: the first group of two questions relates to the general problem of fitting CRBP into the ME scheme; the second group concerns the issue of adapting the ME approach to temporally extended problems. These will be discussed in turn.

With respect to question 1 above, consider that the ME architecture for supervised learning can be imagined to produce its output in one of two ways. The output unit value for each expert can be multiplied by the related output from the gating network and then linearly combined with each corresponding expert output to form the overall system output vector. Alternatively, all the expert network output vectors can be treated as separate system output vectors on each cycle (that is, each one is compared with each and every target vector). In the CRBP scheme, only the first of these approaches is sensible, as the second scheme would require that several separate motor commands be issued for each input vector.

However, this choice has consequences that become clear in considering question 2 above. Referring again to the original ME architecture based on supervised learning, the result of choosing a linear combination of outputs was a mixture of co-operating experts that made a proportional contribution on each training case. In the CRBP context each input vector corresponds to a training case, so the expected consequence of adopting this scheme would be that each sensory modality might contribute to the

motor output vector on each step. Although this idea is quite attractive, there are implications for the learning process that are less desirable. As Jacobs et al. (1991) point out, this strong coupling between the experts makes it hard for the gating network to learn the task-decomposition. This is because adjustments to weight vectors in one expert effectively influence changes in the weight vectors belonging to other experts. The authors indicate that this is not problematic if the experts can be trained separately on each set of distinct, subtask-related cases. Indeed this was the broad approach used in training the Addam architecture to perform mobile agent tasks (section 4.2 of this thesis). However, the intention here was to devise an approach to continuous learning that did not rely on the designer to separate out the situational categories before the agent actually encountered them in their environmental context. The solution to questions 1 and 2 above may appear makeshift, but it to be proved effective. It was to assume that each expert contributes proportionally to the output vector, but that the error to be backpropagated should be calculated according to equation 7.3. This assumes that each expert makes its own complete proposal on each step. There is of course a random aspect to the CRBP target vector so there is no inherent contradiction in this strategy. For example, where an expert contributes the largest component to a particular motor decision that proves good (that is, receives reward), the error backpropagated is proportional to its contribution. In this way, the inchoate experts should co-operate to explore the problem space but should compete to exploit the emerging domain knowledge, that is, to become experts on a particular subtask (in this case the sensory subdivisions previously identified).

Answers to the second group of questions have been prefigured in section 3.3. For the reasons discussed there in general terms, the answer to question three above was to

use an SRN. In relation to this specific problem, its recognised ability to predict sequences (though of course only symbolically-encoded ones) was interesting. It seemed possible that it might be able to learn to solve the high-level control problem (the generalised sequence of action selections necessary to navigate through the cluttered domain). The algorithm in Figure 12 (Chapter 6) represents a handcrafted solution to be replaced by this new approach. To use Mozer's terminology (subsection 3.3.2 of this thesis), a TIS memory was chosen to be the gating network for the adapted ME architecture.

The last question (question 4 above) to be answered concerned the choice of learning algorithm for the recurrent network. Some general possibilities were discussed in subsection 3.3.2. The solution to problems, such as instability and exponential computational demands posed by more specialised training algorithms, offered by the SRNs only slightly amended version of the standard backpropagation algorithm appeared even more attractive in the context of CRBP. It was accordingly adopted. Figure 19 shows how it fits in to the CRBP scheme.

7.3 Studies of simulated adaptive behaviour using the RME control architecture

The simulations in the studies described in section 6.4 of this thesis were replicated to compare the performance of the new integrated architecture with the observed performance of the original so-called Crill architecture. In addition, studies were undertaken to investigate the role of recurrent connections and the implications of the new architecture in terms of the deployment of sensory modalities.

7.3.1 Behaviours and related sensors

Recall again that the simulated robot's goal was to seek out a series of light sources. These were arranged to test the agent's ability to select conflicting actions and avoid getting stuck in the cyclical behaviour patterns typical of simple behaviour-based mobile robot controllers. Inputs to its neural network modules came from three distinct banks of sensors. These consisted of bump sensors, range finders, and light-intensity sensors and the behaviours related to these sensory capacities were as fully described in Chapter 6:

- light-seeking;
- contact-based obstacle avoidance;
- range-based obstacle avoidance.

7.3.2 Simulated mobile robot environment

The same environment was used for each study, as fully described in Chapter 6. For the studies discussed in this chapter the environment was set up as in Figure 14 with the intention that the simulated robot should seek a series of light sources. As before, a light source would be switched off when the robot made contact with it and the next in the series would be switched on. The light sources so were positioned in relation to the configuration of obstacles, building and walls as to require the flexible co-ordination of the behaviours in order to achieve the overall goal of reaching every light source in the sequence.

7.3.3 Simulated robot details

The simulated robot for these studies was equipped as follows:

- 12 sensors, consisting of four of each of the three types described in subsection 5.3.2, "light sensors", "range-finders" and "bump sensors".
- 3 binary motor inputs encoding the 8 (2^3) fixed directions of movement possible under the first of the schemes described in subsection 5.3.2.

Recall again that the chosen motor scheme ensures that the binary complement of a given motor control vector will head the robot in the opposite direction thus giving the CRBP approach a meaningful interpretation at the motor level (section 6.3.3).

7.3.4 Module details

Four distinct types of feedforward or partial recurrent ANN were discussed in section 6.2.2 as possible ways of implementing the sensorimotor modules of the Crill architecture. It was therefore decided to conduct studies with different instantiations of the architecture using each type.

Recall also that, over the series of trials, the algorithmically controlled Crill, with modules having one hidden layer and no recurrent connections, was found to be the most effective in terms of the number of cycles required to complete all the tasks. Consequently, modules conforming to this pattern were chosen for the architecture with TIS gating network control.

As in the Crill studies, for each instantiation of the architecture, the sensorimotor modules were uniformly implemented as in Tables 3, 4 and 5.

Table 3: Details of MERGe architecture (v.1)

Architecture	MERGe v. 1					
Module	Expert 1			Expert 2		
Net type	MLP			MLP		
Layer	Input	Hidden	Output	Input	Hidden	Output
Units	12	4	3	12	4	3
Connections (internal)		Fully with input units	Fully with hidden units		Fully with input units	Fully with hidden units
Connections (external)	One-to-one with sensors		One-to-one with motor units	One-to-one with sensors		One-to-one with motor units
Activation Function		Logistic	Logistic		Logistic	Logistic

Table 3 (cont.)

Architecture	MERGe v. 1(cont.)					
Module	Expert 3			Gating		
Net type	MLP			SRN		
Layer	Input	Hidden	Output	Input	Hidden	Output
Units	12	4	3	12 + 4 context units	4	3
Connections (internal)		Fully with input units	Fully with hidden units	Context one-to-one with hidden units (feedback)	Fully with input units (including context)	Fully with hidden units
Connections (external)	One-to-one with sensors		One-to-one with motor units	One-to-one with sensors		One-to-many with expert net outputs
Activation Function		Logistic	Logistic		Logistic	Softmax

Table 4: Details of the MERGe Architecure (v.2)

Architecture	MERGe v. 2					
Module	Expert 1			Expert 2		
Net type	MLP			MLP		
Layer	Input	Hidden	Output	Input	Hidden	Output
Units	4	4	3	4	4	3
Connections (internal)		Fully with input units	Fully with hidden units		Fully with input units	Fully with hidden units
Connections (external)	One-to-one with sensors		One-to-one with motor units	One-to-one with sensors		One-to-one with motor units
Activation Function		Logistic	Logistic		Logistic	Logistic

Table 4 (cont.)

Architecture	MERGe v. 2(cont.)					
Module	Expert 3			Gating		
Net type	MLP			SRN		
Layer	Input	Hidden	Output	Input	Hidden	Output
Units	4	4	3	12 + 4 context units	4	3
Connections (internal)		Fully with input units	Fully with hidden units	Context one-to-one with hidden units (feedback)	Fully with input units (including context)	Fully with hidden units
Connections (external)	One-to-one with sensors		One-to-one with motor units	One-to-one with sensors		One-to-many with expert net outputs
Activation Function		Logistic	Logistic		Logistic	Softmax

Table 5: Details of the Mixture of Experts architecture (with CRBP learning algorithm).

Architecture	ME					
Module	Expert 1			Expert 2		
Net type	MLP			MLP		
Layer	Input	Hidden	Output	Input	Hidden	Output
Units	12	4	3	12	4	3
Connections (internal)		Fully with input units	Fully with hidden units		Fully with input units	Fully with hidden units
Connections (external)	One-to-one with sensors		One-to-one with motor units	One-to-one with sensors		One-to-one with motor units
Activation Function		Logistic	Logistic		Logistic	Logistic

Table 5 (cont.)

Architecture	ME (cont.)					
Module	Expert 3			Gating		
Net type	MLP			MLP		
Layer	Input	Hidden	Output	Input	Hidden	Output
Units	12	4	3	12	4	3
Connections (internal)		Fully with input units	Fully with hidden units		Fully with input units	Fully with hidden units
Connections (external)	One-to-one with sensors		One-to-one with motor units	One-to-one with sensors		One-to-many with expert net outputs
Activation Function		Logistic	Logistic		Logistic	Softmax

7.3.5 Training details

For each of the instantiations of the architecture, the simulation was run 12 times.

To enable meaningful comparisons to be made across the studies the same training procedure was used for every run:

- all weights were randomly initialised in the range [-0.1, 0.1] ;

- for rewarded actions a learning rate of 0.3 was used;
- for punished actions a learning rate of 0.1 was used;
- no momentum was used;
- no bias units were employed.

7.3.6 Results

The metric used to compare the performance of the different instantiations of the Crill architecture was the number of time steps taken by the simulated robot to achieve the overall goal(extinguishing the final light source after successfully extinguishing all the preceding ones in the sequence). For this comparison, the arithmetic mean of the successful runs for each type was chosen, any unsuccessful runs being discarded. An unsuccessful run was defined as one in which the run was terminated because the simulated robot did not appear to be making any progress after a long period of observation (10,000 control ticks). Standard deviations from the mean were also calculated to indicate the reliability of the metric for each set of runs. These results are summarised in Table 6.

Table 6: Comparison of MERGe and ME controller instantiations with best Crill instantiation

Architecture	Number of runs	Percentage successful runs	Arithmetic mean of successful runs (in control ticks)	Standard deviation of successful runs
Crill (MLP version)	12	100	1745	
MERGe v. 1 (3 X 12 inputs)	12	100	1835.41	108.10
MERGe v. 2 (3 X 4 inputs)	12	50	1267.16	125.69
ME (3 X 12)	12	0	-------	--------

To test the validity of the use of recurrent connections in the gating network, a study was undertaken as a control in which no such connections were employed. Table 2 shows the comparative performances over a series of trials of all three architectures. It can be seen that there was little to choose in terms of performance between the algorithmically controlled architecture and version 1 of the MERGe architecture. The similarity of performance (that is, no gain) reported was regarded as acceptable in the context of the adequate, rather than optimal, control culture serving the (putative) emerging new paradigm of mind (Franklin, 1995).

Without recurrent connections however the ME's performance is shown to be ineffective. It was unable to complete all the tasks and usually failed at subtask 3 (Fig.15, Chapter 6) which provides the first of two difficult navigational situations.

At this point in the research, the hitherto unchallenged (but somewhat counter-intuitive) assumption that each net should receive the full set of sensory data was re-examined, although experiments with Crills had seemed to confirm this overview of the input space was necessary for the agent to complete the whole series of tasks. Indeed, the device of limiting each module to the inputs from one specific sensor type proved disastrous, the agent usually failing to progress beyond subtask 2. However, the introduction of a gating network with an "overview" suggested that copying every module input layer might not be the most efficient approach in view of the *a priori*

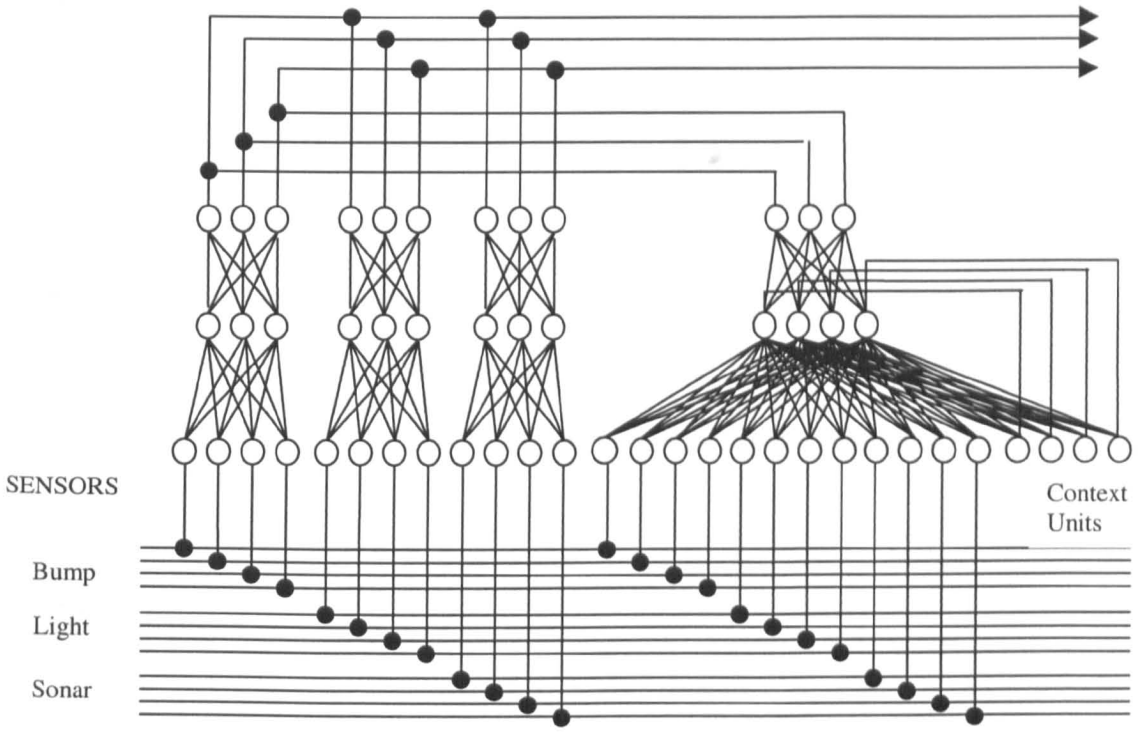

Figure 20: Detailed diagram of MERGE architecture, version 2.

articulation of the problem space along sense-modal lines. Further experiments were conducted in which each of the inchoate expert nets had access only to its related sense-modal inputs, the gating network, of course, still having access to the whole input space. This architecture is shown in detail in Figure 20. It showed ability to complete the overall task in under 1200 cycles, a figure not approached by the other configurations tested, but its performance was not so robust as it was sometimes unable to complete the overall task.

7.4 Concluding observations

The architecture designed for the studies in this chapter represents the culmination of what can be regarded as the first phase of the enquiries that make up the thesis. This

work successfully built on the achievements of its closest antecedents and it is worth recounting the main points here in view of the material that is to follow:

- the advantages of a modular approach under CRBP for the investigation of multiple behaviours (cf. Meeden et al., 1994) were demonstrated;
- it was shown how to use reinforcement learning to address specific problems of the behaviour-based approach, thus lessening still further the designer bias only partially relaxed by a supervised learning scheme (cf. Saunders et al., 1994);

7.5 Summary

The chapter presented a more integrated approach to the learning of competences and action selection. Additional background material was introduced firstly on recurrent neural networks and, secondly concerning a representative modular architecture for static domains. Following a more exhaustive analysis of the problem domain, it was explained how these two ideas could be integrated to form the basis for the new control architecture. It was shown that the new architecture could achieve a level of performance similar to the original architecture when tested in replica experiments with the advantage of a unified approach to control and sensorimotor learning. That this is still essentially a reactive control approach is discussed in the next chapter, where ways of stepping beyond this limitation are also proposed.

CHAPTER 8

LOOKING BEYOND THE INSTANT

SUBSTRATES FOR TEMPORAL EMBEDDING

8.1 Introduction

This chapter focuses on the intimately related foundational problems of representation and grounding, building on insights gained in the preceding investigations to delineate more clearly than hitherto the role of temporal processing in these most fundamental issues. First, these insights are marshalled to construct a more comprehensive critique of the major behaviour-based alternative paradigm. The work in this chapter is offered as a small contribution to the increasing debate on fundamental psychological and philosophical issues apparent within the SAB research community. As such, it can stand alone, but it also serves as a bridge between the studies in the last two chapters and those in the next. Much of the material was presented in Rylatt and Czarnecki (1998) but here there is a more detailed critique of the subsumption architecture and examination of the concept of *situated action,* intended to clarify the argument still further.

8.2 Representation and the subsumption architecture

In this section the critique of the subsumption architecture (section 2.5) is extended and deepened. Based on an examination of its stance on the issues of representation

and grounding it is argued that this and related approaches are unsatisfactory as the basis for a new AI programme, despite impressive project-level achievements. During the development of the argument, some further key ideas are introduced that are essential to an understanding of the proposals that follow in section 8.3. For discussions of the underlying issues, see Harnad (1990), Steels (1995).

8.2.1 Physical grounding

The ideas in this chapter stem partly from some deeper reflections on the well-known aphorism "the world is its own best model" (Brooks, 1991a). To trace the idea back to the seminal work (Brooks, 1986), it can be read simply as an alternative approach intended to circumvent the problems faced by conventional planner-based architectures. These generally failed to scale up to real-world performance requirements because they relied on a central world model that needed continual updates to maintain consistency with the changing environment (section 2.4.4). Instead, Brooks' early creatures worked on the principle of a tight control loop with "little slack" between input and output. Brooks went on to claim much wider significance for the new approach, and to set out further principles establishing a position essentially in opposition to the prevailing view of cognition as symbol manipulation. As these ideas were not mentioned in the more pragmatic earlier papers (for example, Brooks, 1986), it may be that this position resulted from reflection on the practical issues, and from the idea of *situated action* (SA). The latter appears to have been retrofitted and developed into a "stronger" form known as *nouvelle AI*. According to Brooks, this stronger version follows from what he called the Physical Grounding Hypothesis (PGH, Brooks, 1990). The PGH required that "intelligent systems must have their representations grounded in the physical world2. It will be

argued that, in the case of the subsumption architecture at least, this was merely a prescription for building intelligent systems connected to the world via sensors and effectors rather than keyboards and cathode ray tubes.

As a "hypothesis", the PGH really lacks sufficient substance and structure to be testable. It may be that the ideas it encapsulates simply needed this kind of rubric to highlight their opposition to symbolic AI's own conceptual foundation, the Physical Symbol System Hypothesis (PSSH, Newell and Simon, 1976). Its acceptance requires something like a leap of faith, for Brooks (1990) segued to a position holding that once a commitment to the PGH has been made, "the need for symbolic representations fades away entirely". It is this dictum that underlies the claim for the PGH as a stronger version of SA and it will be instructive to examine this claim here as it seems to have gone more or less unchallenged in the literature.

Firstly, it should be noted that SA is rooted in ethnomethodology (and hence sociology). It is essentially a critique of AI planning, holding that planning (that is the manipulation of background knowledge in the form of symbols) has a subsidiary rather than a primary role in human activity. It has nothing to say about physical grounding or about the mechanisms that give rise to representational content for an individual intelligent system. Rather, the "activity" it promotes over planning to the leading role in human behaviour is "free-floating" in a similar sense to the symbols of the PSSH. Indeed, as the account is wholly in social terms it is difficult to see why the PGH is claimed to be a version of it in any sense, whether stronger or otherwise. Superficially, however it is possible to concede that the PGH at least appears to put

something tangible in place of ill-defined "activity".

However this may be, there seems to be a paradox in the PGH. Firstly, note that although traditional symbol systems were to be swept away, the kind of representations that could replace them was not discussed. For PGH agents, input/output representations were to be virtually semantics-free; everything had to be explicit, expressed by the fixed topology of AFSMs (see subsection 2.4.4.) and all knowledge had to be extracted from physical sensors. Thus, paradoxically, the PGH appeared to rule out rather than support the ultimate appearance of higher-level abstractions grounded in sensorimotor representations. Indeed, such abstractions, according to Brooks (1990), "have to be made concrete" (i.e. supplied by the designer and implemented as top-down constraints on bottom-up development).

Although Brooks seemed to acknowledge that representations might play a part in higher-level intelligence, their role would not be developmental. It seems that somehow they could be introduced when the time was right. In retrospect, this seems a curious view, for by this light, "grounding" as an initial requirement for the design of autonomous agents, would happen *in the absence* of representation. Consequently, it is not merely mischievous to ask just what it was that the PGH required to be grounded. The not very satisfactory answer is to be found, if anywhere, in Brooks' defence against charges of denying that intelligence uses any form of representation. On this point, at least, the position in Brooks (1991a) seems superficially more reasonable: he claimed no more than that representation is the "wrong unit of abstraction" for the "bulkier" parts of intelligence (that is, sensorimotor abilities). Therefore, by his lights there is no contradiction in holding that this part of

intelligence (as suggested earlier, also the "hardest" part) can indeed be achieved without representations. However, it remains to be determined what kind of representation ultimately will be required and, most significantly, how it will be supported. The kind of representations that are not going to be required are identified in (Brooks, 1991a) as "explicit manipulable (sic) internal representations". Although these are of course just the kind of representations those levelling the aforementioned charges would be concerned with, Brooks claims here that even human-level intelligence is achievable without them. By way of explanation, Brooks falls back on the following argument. We only recognise the power of thought in others (other intelligent systems) by introspecting our own intelligence and so, by the same yardstick, eventually we shall witness the *emergence* of intelligence in our own situated, artificial agents. Emergence is another key idea in behaviour-based control theory (Steels, 1994). In this instance, it seems to imply simply that intelligence will appear, in artificial agents, as the epiphenomenon of interactions with the same complex world that gave us our existence proof of intelligence.

8.2.2 The fallacy of observer idealism

The position implied by Brooks' argument for *nouvelle AI* was essentially a radical one. It was tantamount to holding that representational content and relationships are observer dependent. Interestingly, observer-dependence is the basis of another, much more philosophically sophisticated variant of SA called *enactionism* (Maturana and Varella, 1988). Therefore, it is significant that its proponents should have pinned their hope for a practical demonstration of their theories on Brooks' creatures. In a sense, Brooks' work can be viewed as a project within this emerging programme or intended

paradigm. Remarks such as the following are typical and have a similar optimistic ring:

> We are willing to bet with Brooks that in a relatively short term such artifacts will have evolved into generations of sufficiently intelligent Creatures whose efficacy can begin to be exploited. (Varela, Thompson and Rosch, 1993).

Bickhard and Terveen (1995) have criticised the enactionist view, because it commits the foundational error of *observer idealism*, a philosophical position considered dubious in that it shifts all accounts of representational content to mysterious and unknowable observers. Such observers are akin to the homunculi that haunt traditional accounts of representation, and hence are open to the same objections of infinite regress. Curiously, the behaviour-based movement, primarily represented by Brooks, seems to have been exempted from this judgement. For example, Bickhard and Terveen proposed a radical alternative account of representation called *interactivism*, emphasising the critical importance of intrinsic timing and dynamic topologies. Their alternative conception seeks to avoid the untenable philosophical positions to which symbolic AI and other rival accounts fall back. Yet, their account incorporates a view of the subsumption architecture that is critical only to the point of admonishing its creator for overlooking its representational potential.

It will become clear that this thesis is moving towards a position that has much in common with interactivism. However, part of its contribution will be to show that, *per contra*, despite the many useful insights offered by Brooksian *nouvelle AI* and its undoubted practical successes, it too is foundationally flawed. That *nouvelle AI* falls back to an observer-idealism has already been indicated in the above discussion of

Brooks' remarks. In order to show, more than anecdotally, why this is inevitable, and subsequently, why an alternative substrate based on ANNs of a certain kind is not so fated, it will be necessary to look in more depth at the substrate on which the subsumption architecture rests. This will be done in the section 8.2.4. Beforehand, some further discussion of a key idea in enactionism, and its ramifications, will help to focus this issue of substrate viability. It will also cement the idea that the fallacy of observer idealism leads to an uncritical acceptance of substrates for an artificial mentality, mainly on the grounds of their interactive credentials.

8.2.3 Structural coupling

The notion of *structural coupling* that underlies theories based on SA, such as enactionism, differentiates them most clearly from cognitivist ideas based on information processing. According to this notion, cognition is a structure that evolves from the interaction of the individual agent with the unstructured world it encounters. Accordingly, autonomous agents cannot be information processors, because there is no information "out there" to be processed and no representational atoms to encode. Structural coupling should not be confused with idealism; it does not posit a detailed, pre-determined mental structure to be projected as the outer reality. In some sense, at the philosophical level, it represents a middle way between atomistic materialism and idealism. From the SA perspective, because cognition arises from *closely coupled* sensory and motor activity, disembodied minds, even those with some sensory apparatus, are ruled out. On the question of precisely what kind of structures undergo this coupling the SA movement, generally, is not definitive. However, it seems clear that structural coupling implies at least that cognition must be distributed across

collections of sensorimotor modules, or networks, acting in parallel. At this level of description, the notion is relatively neutral concerning the nature of the substrate. This lack of definition, together with the acceptance of observer idealism noted above, admits many possible candidates for the role of cognitive substrate, including of course the AFSM. That this relative neutrality is not a tenable position if the attractive notion of structural coupling is to be accepted will now be argued by way of example.

8.2.4 Augmented Finite State Machines

Firstly, recall (and bear in mind) that the subsumption architecture was conceptualised as a layered architecture permitting only minimal information processing between input and output within the modules composing each layer, or between layers. If it is accepted that no representational significance can be ascribed to the simple numerical values passing along the "wires" of the architecture[8], the atomicity [9] of these systems must be defined at the level of the AFSM (see section 2.4.4). Consider also that the atomicity of symbolic AI is defined at the level of its constituent symbols. These are, in computational terms, its instantiable variables. Its representational power, as well as the probably fatal problems it faces, derives from the combinatorial possibilities inherent in its atomicity. It will now be argued that the atomicity of *nouvelle AI* denies it both the (at least interesting) modelling capacity of traditional AI and the possibility of ever achieving a true representational content free from user semantics. An example of a (software) AFSM is illustrated in Figure 21.

[8] Brooks (1991a) maintained that low-bandwidth communications and simple numerical instantiations of variables do not amount to any explicit, internal symbolic manipulation.

```
(defmodule avoid 1
        :inputs (force heading)
        :outputs (command)
        :instance-vars (resultforce)
        :states
                ((nil (event-dispatch (and force heading) plan))
                (plan (setf resultforce (select_direction force heading))
                        go)
                (go (conditional-dispatch (significant-force-p resultforce) 1.0)
                                start
                                nil))
                (start (output command (follow-force resultforce))
                nil)))
```

Figure 21: Example of an subsumption AFSM module

The AFSM has registers, instance variables and clocks in addition to internal states, inputs and outputs. Of course, finite state machines are not inherently mysterious. Also referred to as *state machines*, or *sequential machines*, they are the foundation on

[9] This neologism is intended to expresses the fundamental unit at the level of description appropriate for understanding intelligent behaviour.

which the components of digital electronic computers are based, and the conceptual tools for the theories of computability underlying computer science. In addition, as outputless *finite state automata* they are the basis for modern programming languages. At this level of description, indeed, neural networks too can be encompassed. Therefore, it might seem, at first sight, that this choice of building blocks imposes no more constraints on function than are suffered by any other conceptualisation that can be realised, or simulated, using existing computing devices. However, it is significant that Brooks chose to define the basic modules at this level; in a sense, it represents a statement of intent.

To appreciate the last point, consider that finite state machines (FSM) are not the building blocks of symbolic AI, where atomicity is defined rather at the level of combinatorial symbols. FSMs exist at some level that stands in the same relationship to symbolic AI as do electrochemical descriptions of neural processes to cognitive psychology / neuroscience. Symbolic AI exists in a conceptual framework that is independent of physics and chemistry, starting from free-floating symbols that can only be defined by circular arguments. Similarly, the subsumption architecture, arguably the representative architecture of *nouvelle AI* and the white hope of enactionist theories, rests on an arbitrary mechanism chosen so as to conform to a user semantics. This kind of FSM is not a generator or recogniser of symbol strings as in symbolic AI because its inputs and outputs are simple numeric values or bandwidth-limited wires. Its atomic nature is apparent at the behavioural level as a particle that has its functional boundaries determined in an observer space. This can be clearly seen, though the interpretation is novel, by inspecting the AFSM in Figure 21. The Avoid module relies on a simple artificial potential fields concept (Khateb, 1986) that

converts sensor readings directly into motive force. Recall that, at this level of description, undesirable project-level consequences for *nouvelle AI* - stemming from this attempt to build an artificial mentality with engineered components - were identified in section 2.4.5 of this thesis. Here, however, the argument shifts to the programmatic perspective. Essentially, although the subsumption approach engineers *out* the problems associated with encoded symbols, it does not (and indeed cannot) engineer *in* any possibility of *for-the-machine semantics* (that is, semantics that are not imposed by the user). It should now be clear that its atoms, the putative building blocks of a machine mentality, are, essentially, particles of from an engineering domain-ontology, hard-wired together in heavily task-oriented configurations. Such a massively predetermined, minimally interconnected system, with hard-bounded subsystems is most unlikely substrate for the formation of concepts necessary for a reflective intelligence.

Although these systems have a functional organisation that can support some kind of non-symbolic "representation", it is to be expected that this will always be of an ad hoc nature. Mataric (1992) provides an example of this, describing the robot Toto's "memory" as a dynamic graph of "landmark recogniser modules" corresponding to gross sonar configurations (with user-semantics such as "wall right") and compass readings. This characteristically ad hoc, engineering approach, originally criticised in this thesis at the project-level (section 2.4.5) can now be attributed more fundamentally to an atomicity that is not amenable to the integration of truly interactive subsystems between input and output. However there is another, perhaps even more fundamental problem with the FSM substrate that also undermines symbolic AI and some forms of connectionism, as discussed in the next section.

8.3 Embedding autonomous agents in time

A necessary preamble to the main argument in this section introduces some key terms and continues the discussion (begun in section 8.2) as a means of elucidating them. The specific idea of *temporal embedding* was first aired (Rylatt and Czarnecki, 1998) during a SAB conference session on philosophical issues. These indicated the increasing awareness of the need for foundational debate in the field (see also, for example, Clark and Wheeler, 1998; Spier and MacFarland, 1998; Scheier and Pfeifer, 1998). The relevance of the notion has subsequently been recognised in, for example, (Ziemke, 1999). The related term *temporal grounding* (Nehaniv, Dautenhahn and Loomes, 1999) also represents – *at some level* - a concern with building systems that transcend simple reactive behaviour. However, as will become clear, the argument in this chapter is differentiated as it proceeds essentially from substrates (i.e. that temporal processing mechanisms at an appropriate level of granularity need to be in place as a prerequisite for grounding).

8.3.1 Non-conceptual contents

Clark (1993) regarded *non-conceptual contents* as the prerequisite for systems that ultimately aspire to *conceptual contents* (these can be thought of as broadly equivalent to the manipulable representations discussed in the previous section) and hence for the ultimate programmatic aims of AI. Unlike Verschure (section 3.4 of this thesis), Clark accepted that NETtalk did indeed learn to recognise similarity between vowels. Hoever, heargued that the knowledge NETtalk gained was too task-specific: "highly

intertwined with its (NETtalk's) ability to use the knowledge to perform text to speech transformation - it lacks the general idea of a vowel". From this, he proceeded to the position that networks with the ability to negotiate a certain domain are capable of supporting contents that properly consist just in that ability. This a position (not stated by Clark) appears close to the Heideggarian analysis of a world fundamentally experienced non-reflectively as *ready-to-hand.* He did not claim that NETtalk supported this kind of content, only that it might be capable of doing so if some unspecified conditions could be satisfied. However, he did suggest that networks of NETtalk's type which he called "first-order" [10] are excellent candidates to become systems that will form such contents. This is simply because they "know their way around" a domain without conceptualising it (that is non-reflectively or without forming manipulable representations). The significant question for Clark was how conceptual contents can arise from these non-conceptual ones.

8.3.2 A starting point for a "developmental" approach

The explanatory lacuna noted at the end of the previous section cannot be bridged by conventional connectionism. Verschure's convincing argument (discussed in section 3.4) showed that even at the level of inputs this approach was subject to a symbolic interpretive bias. Additionally, Bickhard and Terveen (1995) concluded that connectionist representations always have an underlying symbolic interpretation, observing that this resulted from a compulsion to regard as encodings the activation patterns that differentiate input patterns in typical connectionist architectures. Non-

[10] Feedforward networks relying on backpropagation for gradient descent error minimisation – a usage that is unfortunately not consistent with the usage of the term 'second-order' later in this thesis.

conceptual contents, however they may be construed, cannot be encodings. It therefore seems futile to look for evidence of them in networks of this kind (pace Clark above). It is obvious that the newer situated form of connectionism has moved away from an insistence on regarding representations as emergent atoms. However, some versions, to some extent and at least implicitly, seem to share the anti-representational stance of *nouvelle AI*. They perhaps seek to replace its reactive components with neural networks construed as stateless input/output devices, for example, and, most pertinently, the work of Saunders et al. (1994, section 4.2 of this thesis). Indeed, much of this earlier work, while beginning to recognise the importance of internal state, did so more from a typical behaviour-based control-oriented perspective, although it was not predicated on any strong rejection of the role of representation. It seems clear that if internal state is eschewed in situated connectionist approaches, it may lead to foundational errors similar to those noted in section 8.2. The need for a developmental approach was a key insight provided by Clark (above) but the evident need is for a starting point that is neither subject to interpretative bias nor oblivious to representational issues. In relation to autonomous agents, the explanatory lacuna is how even non-conceptual contents *for-the-machine* arise in the first place. The notion of embedding autonomous agents in time is intended to encapsulate the view that a temporal processing economy integral to the substrate is a prerequisite for a developmental approach. The notion assumes that such an agent will also be embedded in space either literally, or, as here, in simulation.

8.3.3 Naive time

The issue of time was omitted from the discussion of representation in section 8.2. Against that background, the behaviour-based approach can be seen as an attempt to embed agents in space so that they would not suffer from the closed system brittleness of traditional, disembodied AI systems. However, the problem of brittleness is reducible to the problem of representational contents and symbolic AI's difficulties in this respect are partly attributable to the lack of a temporal essentiality. This arises from its theoretical foundation. The absence of any conception of time (beyond the idea of *sequence*) is a feature of Turing machine theory, as noted by Bickhard and Terveen (1995), in their critique of the symbolic AI view of representation (referred to as *encodingism*). The argument is that physical Turing machines, such as computers, are provided with clocks to drive their sequential processing, but these are *engineering* accessories extraneous to the model and can provide only a user-semantics. It might be argued that subsumption robots of *nouvelle AI* are more intimately involved with time, and they indeed appear to have been exempted from the above critique. However, this would be to confuse real-time behaviour (obviously something these creatures are good at) with *real* time (in the sense defined below); and of this, they too have no conception. The clocks that augment the finite state machines that form their substrate are also examples of engineered time, so the management of temporal effects and dependencies is extraneous to the model. It follows that, as with symbolic AI systems, their temporal imperviousness denies them for-the-machine representational content. It must be emphasised that it is not the intention in this thesis to attempt to argue conclusively that any kind of machine can achieve this, only that some substrates are intrinsically more promising than others. True interaction with the environment must intimately involve basic mechanisms, a

flow of internal system processes unimpeded by observer-determined abstraction barriers[11]. The loop through the world is not enough - there must also be many recursive loops through a tractable medium internal to the agent that effectively embed the agent in time as well as in space.

Clearly, conventional connectionist research, too, has been concerned with spatial patterns or with giving a spatial representation to essentially temporal structures such as natural language or speech. Adducing the example of scientific research into human audition, Port et al. (1995) argued that the standard models suffer from predication on what they call *naive time.* This is the notion that what we call real or biological time, consisting in the information available to an organism intrinsically bound with timing, uses a representation of absolute time, that is time measured by reference to some external clock. One consequence of this is that such models usually incorporate a temporal buffer (as in NETtalk). Input sequences of arbitrary length are collected and presented to the model contemporaneously. There appears to be no direct biological evidence for this approach but buffered models are common in connectionist research into processes that unfold over time. The shortcomings of this approach and of explicit time models generally, were described by (Elman, 1990), in the context of human language processing experiments. Although, in his experiments the system was not situated (in the sense of having real or realistically simulated sensors or motor outputs), Elman's alternative *implicit* model of time (the SRN) contributes strongly to the notion of embedding in time. From this perspective, it can be interpreted in the following way: the *effects* of time on processing in the system solely constitute the system's *model* of time, and thus time is no longer an engineering

[11] Brooks refers to subsumption's behavioural layers as abstraction barriers (Brooks, 1986).

appurtenance extraneous to the model. In the SRN, this is achieved through simple feedback connections between layers. According to Elman, "representations", formed in the hidden units of the SRN (ignoring, for the moment, how these are to be construed), are involved in mapping both current external stimuli and prior internal states to outputs. They thus intimately bind spatial and temporal patterns in a task-related manner. This connectionist ability to model simple, dynamic short-term memory (STM), at the level of granularity of its processing elements, differentiates neural networks from both symbolic AI systems and *nouvelle AI systems*. This is so although both, in some manifestations, and *at some level,* can also be viewed as dynamical systems of a kind (Franklin, 1995).

The distinction between traditional accounts and the one this thesis is moving towards is most clear at the dynamical systems level of description. The dynamic space of the standard, connectionist feedforward model is organised around local points or regions of attraction. The characteristic behaviour of this model encourages interpretations similar to the folk psychology account of stable referential relations between internal and environmental states rejected by Bickhard and Turveen (1995). However, in recurrent models of the kind advocated here, the dynamics may be characterised by trajectory attractors. In (Peschl, 1995) may be found support for the view that traditional representational misconceptions should be abandoned and replaced with a conception of representations in terms of such dynamic state trajectories[12] through the activation space of cognitive systems. In sum: the perspective opened up in this chapter is of systems and subsystems that model processes intrinsically (rather than ad

[12] If the cognitive system is modelled using a discrete time dynamical system the term is more correctly *itineraries*, but as is well known, continuous time dynamical systems can be approximated to an arbitrary degree of accuracy by controlling the granularity of the model, so the argument holds for connectionist models too.

hoc) dynamical. This opens the possibility of enabling autonomous agents to enjoy non-conceptual contents, that is contents-for-the-machine rather than for the observer / user. The next section explores the possibility that such models may be provided by connectionist networks, provided they are not construed merely as stateless input-output systems, and that they conform to one of the requirements of interactivism, namely that the system should interact with an environment so that past outputs affect subsequent inputs.

8.4 Summary

In this chapter the notion was developed that situated autonomous agents need to be embedded in time as a prerequisite for enjoying the representational contents necessary for the programmatic aims of AI. It was argued that the temporally impervious substrates of symbolic AI and *nouvelle AI* could not achieve such temporal embedding. In the next chapter some early work is presented that seeks to investigate the potential of recurrent neural networks in this respect.

CHAPTER 9

ARCHITECTURES AND STUDIES (III)

A FRAMEWORK FOR DELAYED-RESPONSE LEARNING

9.1 Introduction

In this penultimate chapter, the focus turns to the role recurrent neural networks might play in the simulation of a form of behaviour that represents a step beyond the essentially reactive kind studied in Chapter 6 and Chapter 7. A foundational position on the need for so-called temporally embedded systems was established in the previous chapter. In the first section of the present chapter, a new second-order architecture is described, together with a framework designed to support the study of delayed response in the context of SAB. This form of behaviour generally requires more than the single step memory previously investigated. In the second section, two further sets of studies are described that suggest how the new approach can be used to investigate the potential of various simple models of short-term memory, both extant and novel. These studies are offered as a preliminary step towards the longer-term aim of gaining insight into the temporal processes, held to be intrinsic to the grounding process (see Rylatt and Czarnecki, 1998; Rylatt and Czarnecki, 2000).

9.2 Architectures for delayed-response learning

In Chapter 4, the work of Ulbricht (1996) was briefly discussed. She described how her *input state network* could learn so-called *time-warped sequences*. Clearly, this behaviour belongs to the class of behaviour studied more generally in cognitive science under the heading of delayed response tasks, for example Guigon and Burnod (1996). In both these examples, sensorimotor encodings were at a high level of abstraction. Guignon and Burnod were most interested in demonstrating a model of neuronal circuits in the prefrontal cortex and in their model each specific stimulus and required motor response was represented by a single, dedicated input or output unit. Ulbricht, though depicting a more "naturalistic" setting, in practice relied on a traditional connectionist scheme of representing inputs by means of alphabetic symbols. Within these constraints, she showed that her network could learn this class of behaviour by relying on task-specialised features dedicated to the temporal processing of *input-level* information. Unfortunately, these same features implied a pre-processing and training approach unsuitable for simulations of sensorimotor activity that try to be more realistic. For the studies discussed later in this chapter, sensorimotor activity would be simulated at a somewhat lower (though still far from realistic) level than described in the earlier chapters (cf. Stein, 1992, Pal and Ker, 1996). Therefore, ways of focusing on input level information had to be found that avoided these problems. In this section it will now be described how:

- enhancements where made to the basic SRN (for the first set of studies);
- a new hybrid architecture was designed and implemented (for the second set of comparative studies).

9.2.1 An enhanced simple recurrent network

It was decided to investigate the possibility that the SRN could be enhanced for delayed response learning by inducing the hidden layer to provide more input-relevant information to be fed back as temporal context. The use of feedforward networks to compress input into hidden layer representations by training the network to reproduce its own inputs at the output layer (auto-association) is described, for example, in (Kadaba, Nygard, Juell and Kanga, 1990). An SRN was set up to perform the auto-association task concurrently with the task of associating percepts with motor outputs by adding extra output units equal to the number of inputs. In training, the inputs were to be used as the target vector for the additional outputs so that a compressed input representation would form part of the hidden layer's state, fed back as context. Additionally, in order to control the amount of state information processed recurrently, a constant "gain" was to the context units. Note that in Elman's original SRN the feedback was modelled as connections to the context layer with fixed weights of 1.0. However, short-term memory can be modelled generally in recurrent neural networks as the convolution of the input sequence with a so-called kernel function (Mozer, 1993):

$$\int_0^t K(t-\tau)X(\tau) \qquad (9.2)$$

where K is the kernel function and X the input function. Applying this idea to the SRN, the equation for the context layer activation can be written :

$$c_j = (1-\gamma)h_j(t) + \gamma c_j(t-1) \qquad (9.3)$$

where c_j is the j^{th} context unit, h is the j^{th} hidden unit and γ is the gain constant. If γ is set to zero, the effect is the same as that of the standard Elman context unit, that is, of *a high definition, low depth* memory unit, according to the terminology introduced by Mozer (1993). In this scheme, an increase in the gain should tune the context units by reducing the definition of the memory, at the same time increasing its depth. Convwersewly, a decrease in the gain should reduce the depth of the memory, at the same time improving its definition. No originality is claimed for this idea – it was an enhancement to the standard model that seemed strongly worth investigating as an independent variable that might be significant.

9.2.2 The hybrid second order input state network

The architecture shown in Figure 22 is a hybrid based on the second-order or quadratic network described by Pollack (1995) and the simple dynamic memory of (Port et al, 1995). It also has similarities with the self-adapting recurrent network (SARN) a hybrid of a second-order network and the SRN (Ziemke, 1996a). An explanation of its task-oriented features will be given later but first its general aspects need to be described. With further reference to Figure 22, it will be seen that this model has two levels of feedback (dotted arrows). Firstly, feedback occurs from the input layer to itself - indicated by showing an extra "layer" at a level before the input layer, similar to the context layer of an SRN. This re-conceptualisation may blur the

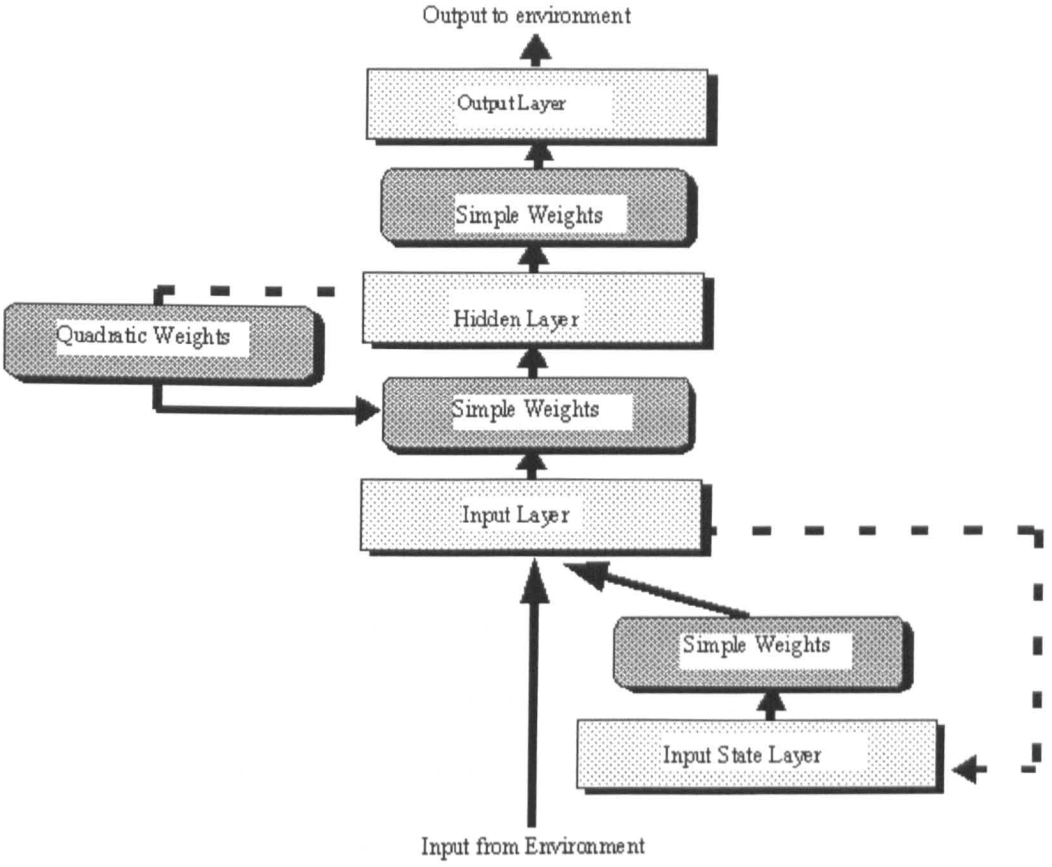

Figure 22: Hybrid second-order architecture with input state.

distinction between "layer feedback" and "unit feedback" drawn by Ulbricht (1996). But in fact it shows the fully interconnected (solid arrows) nature of the self-recurrent input layer quite clearly, conforming to the activation equation of (Port et al, 1995):

$$y_i(t+1) = squash[\alpha\, y_i(t) + \sum w_{ij} y_j + input + bias] \qquad (9.1)$$

In equation 9.1, α is a decay rate, y is the feedback and *input* represents direct sensory input (*bias* is optional). The second level of feedback is from the hidden layer and this incorporates the second-order aspects. The architecture is illustrated as a single network but it may be conceptualised as two separate nets in a master-slave relationship. Feedback, from the hidden layer at time $t - 1$, is construed as the input layer of the master network, which is then fed forward through a linear function to its output layer. The units of this output layer, in this model, represent the weights of the slave net hidden layer. The terms "quadratic" and "simple" reflect the multiplicative order of the connections. It will be seen that the master subnet will have a number of weights equal to the multiple of the simple weights between the layers it connects and the number of units in its input layer. The model is trained using backpropagation in the manner found to be successful by Elman (1990) and Pollack (1995) that effectively backpropagates error a single step in time. In this case, as in Ziemke (1996a), the hidden layer weight vector, adjusted by the slave net at time t - 1, serves as the target vector for the master net at time t. Thus, the master weights multiplex the activation function of the slave's hidden layer, giving richer representational potential.

9.3 Simulated adaptive behaviour studies using architectures for delayed response learning

The two studies described in this section were set in a framework based on a synthesis of ideas from the aforementioned work of Ulbrich (1996) and Guigon and Burnod (1996). The framework was designed to support investigation of delayed response learning in the kind of simulated mobile robot environment described in the

earlier studies in this thesis. It represents a first attempt at supporting the position established in the previous chapter by providing a basis for practical investigation.

The re-interpretation of Ulbricht's work in terms of Guigon and Burnod's neurophysiological laboratory-based indicates that traditional connectionism can still have relevance for SAB through a salient and mutual transfer of ideas between different levels of description. It also highlights some of the problems that emerge when levels of description change. In order to show this fully, it is necessary to recall Ulbricht's depiction of a scenario for so-called time-warped sequence learning (see Figure 23) and her scheme of representation.

Although the figure depicts an agent moving along what appears to be a road with a T-junction and a landmark or sign, the salient features were simply represented as streams of discrete input and output tokens. At this level of description the problem is related to those discussed by Port et al. (1995), and (less obviously) by Elman (1990), and Pollack (1995). This observation is important because the agent's percepts were modelled at the level of symbols; and the unfolding of its perceptions in time, by means of strings of those symbols. At some level of description, this broad area of enquiry is concerned with either the recognition or prediction of grammars composed of such strings. In Ulbricht's scenario, instead of being interpreted as "accept" states signifying the recognition of a string as a member of a particular grammar or "language", the outputs of the network represent motor commands, which might be

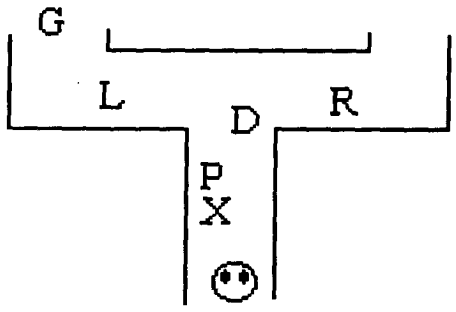

Figure 23: Sketch of the time-warped sequence learning problem showing the landmark (X), concealed goal (G), decision point (D). The gap between X and D is represented by P. Re-drawn from Ulbricht (1996).

interpreted in the following way (Figure 23): "go straight ahead while symbol X is perceived, continue straight ahead while symbol P is presented , turn right when symbol D is seen'", etc. X can be construed as some environmental feature that the agent is able to differentiate and use (as a memory trace) to make the control decision at D (maybe another environmental feature such as the perception of a fork in the route). P simply represents a unit interval following the last perception of X. During a sequence of Ps, the agent is not receiving input that is relevant to the decision at D (that is the symbol P has no information content relevant to D). Sequential presentations of these symbols can be interpreted as the constant sampling of a continuous signal, with repetitions of symbols representing varying rates of presentation. Two non-trivial problems investigated by Ulbricht were:

1. how to handle long-term dependencies (for example many repetitions of P before D); and

2. how to generalise across rates of presentation (that is, to ignore variations in rates so that, for example, the model can predict D following PPP having been trained only with P).

Reinterpreted at a lower level of abstraction, these became the foci of the two studies described here.

9.3.1 Behaviours and related sensors

In delayed response learning experiments, as described by Guigon and Burnod, a subject (typically an animal) in a laboratory setting is given an *instructional stimulus* (for example, one of two lights each spatially associated with a lever). This must be memorised in order to support an appropriate response (pulling the lever) when, subsequently, a decision is required (indicated by a third light, termed the *go-signal*). In Ulbricht's scenario, a correlate of the instruction stimulus was represented by a distinctive feature of the landscape "remembered" by the agent in passing. This was modelled by a single, suitably encoded alphabetic symbol. A correlate of the go-signal – the point at which the agent reaches a T-junction – was modelled in the same way.

In the new framework these ideas where synthesised and reinterpreted at a lower level of abstraction in terms of the available simulated sensors. Thus, the landmarks / instruction stimuli were conceived as objects with clearly distinguishable profiles that the robot would scan with its range finder sensors when moving in the simulated environment. The simulated robot designed for these studies had sensors oriented forwards. A problem associated with this configuration was that there would be a considerable time lapse between the robot's last perception of the stimulus and its first turning movement to negotiate a branch of the T-junction. This would be so even though the simulated robot was holonomic[13] - non-holonomic types would pose even

[13] A sufficient definition for the purposes of these experiments is: the ability to turn in place.

greater problems. As various researchers have noted (for example, Elman (1995), Linn, et al., 1996) existing and novel non-buffered ANN models of STM have difficulty in remembering context information more than a few steps into the past. Even Ulbricht's dedicated input state network seems to have been reliable only up to about six time steps. For these reasons a simplified version of the problem was represented in the simulation: the required delayed response would be a U-turn in a specified direction: each being a distinct temporally extended, motor control sequence. Rather than pre-wire the sequences it was decided to use supervised learning so that the representations responsible for them would be at the same level of granularity as the representations being memorised. This satisfied some of the requirements for structural coupling identified in the previous sections in a way that, for example, Colombetti and Dorigo (1994).

9.3.2 Simulated mobile robot environment

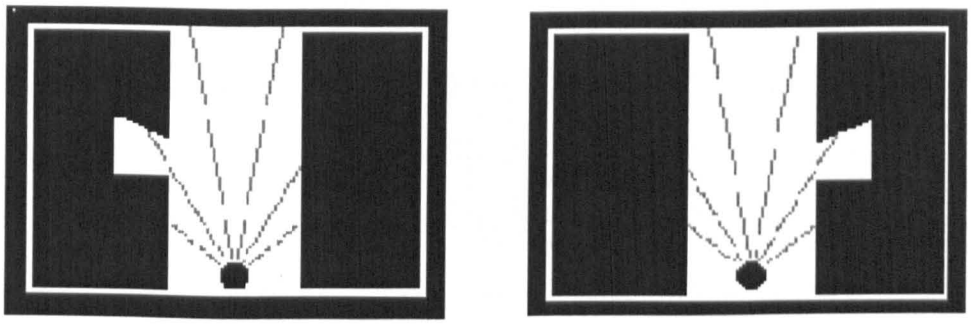

Figure 24: Mirror image environments.

In both studies, the instructional stimulus was modelled as a "notch" or recess in the "solid" wall (Figure 24). This delimited zone offers a distinctive profile to the

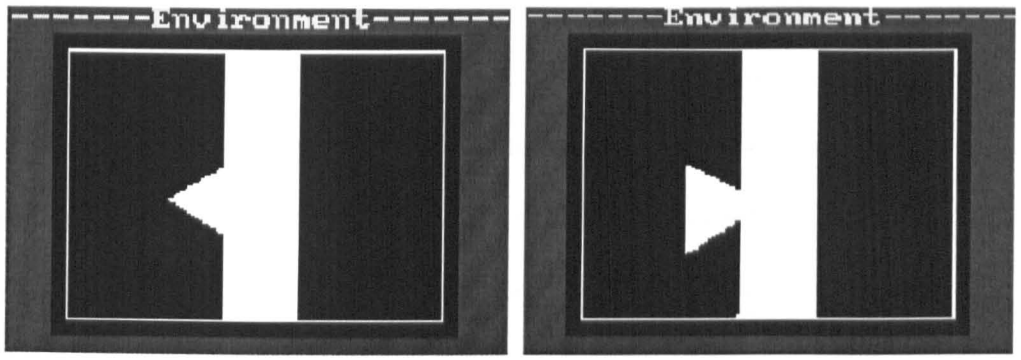

Figure 25: Instruction stimuli from second study.

simulated range finders, in contrast to the undifferentiated expanses on either side. In the first study, this feature was located either to the right or to the left of the central aisle. The delayed response task was to execute a U-turn. If the stimulus-object was passed on left hand side, then the turn should be to the right. Alternatively, if the stimulus-object was passed on right hand side, then the turn should be to the left.

Study 1

Two environments were created using IMRANN's screen painter; apart from the reflection about the vertical axis of symmetry, the two environments were identical in all respects. This procedure was devised to ensure that the neural network did not distinguish accidentally unbalanced features and use these as the basis for its decisions, rather than its memory of the distinct range finder profiles. Figure 24 shows two cropped screen dumps of these mirror image environments.

Study 2

In this study, more demanding recognition objects were used as the instructional stimulus (Figure 25): concavities with distinct contours situated on the same side of the aisle (that is, the instruction stimulus depended not on handedness but on

perceived shape). The investigation of generalisation across time was facilitated by incorporating a decision point into the simulations. This provided a dependent variable that could be adjusted in relation to the instructional stimulu. It took the form of a "go-signal", in this case a simulated light beam that could be flashed on and off at varying time intervals after the instructional stimulus was received. The robot's photoreceptors were the means of detecting this beam. This aspect of the simulation is illustrated in the cropped screen dump in Figure 26.

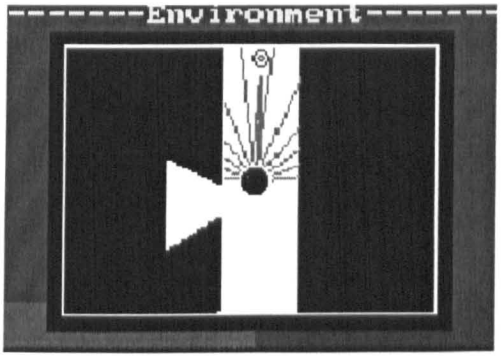

Figure 26: IMRANNS screen grab showing simulated robot receiving a go-signal

9.3.3 Simulated robot

Although similar in outline to the simulated robot used previously in these studies, there were some significant differences:

- bump sensors were not used;
- the simulated range-finders were positioned around the robot's 'front' hemisphere, the beam angle varying according to the number of sensors used;

- two simulated light-sensors were also positioned on the front hemisphere;
- the robot was able to turn in place through 45° in one movement;
- the motor commands were encoded over just two binary units in the output layer of the controller, giving four (2^2) alternatives: straight ahead, left, right and reverse.

As before, translations were simulated in single pixel steps. All sensors were placed symmetrically about the longitudinal axis. Each sensor was connected to an input unit on the input layer of the neural network controller. The only differences between studies were as follows.

Study 1

Six range finder sensors were used.

Study 2

More range finders were added to the robot for this task, making 10 in total, with the idea of improving the robot's ability to make and recall a reliable distinction between the profiles.

9.3.4 Neural network details

Study 1

This was carried out using the enhanced SRN described in subsection 9.2.1. The network, with one hidden layer, was configured as in table 7.

Table 7: Details of enhanced SRN architecture (study 1).

Architecture	Enhanced SRN			
Layer	Context	Input	Hidden	Output
Units	4	8	4	2 motor + 8 predictors
Connections (internal)	One-to-one with hidden layer units (feedback)		Fully interconnected with input and context units	All fully interconnected with hidden units
Connections (external)		One-to-one with sensors (6 rangefinders, 2 photoreceptors)		Motor outputs only: one-to-one with robot motor units
Activation Function			Logistic	Logistic
Bias units			No	No

Table 8: Details of hybrid architecture (study 2).

Architecture	Hybrid					
Module[1]	Slave				Master	
Net type	MLP				SLP	
Layer	Input state	Input	Hidden	Output	Input	Output
Units	12	12	4	2	4	48
Internal connections	One-to-one with inputs units (feedback)	Fully with input state layer	Fully with input layer	Fully with hidden layer	Fully with slave output layer	Full
External connections		One-to-one with robot sensors		One to-one with robot motor inputs	Fully with slave output layer	One-to-one with master first layer weights
Activation Function	Identity		Logistic	Logistic		Identity
Bias			No	No		No

[1]The modules are conceptual.

Study 2

This was a comparative study involving four different architectures. All networks had the same numbers of hidden layer units and motor output units as in study 1. The total number of sensory input units was increased to twelve, the additional four units being

connected to the extra range finder units on the simulated robot. Four architectures were compared. These were, the enhanced SRN used in Study 1 with an additional four predictor units on the output layer, together with the three architectures detailed in tables 7 to 10.

Table 9: Details of simple dynamic memory architecture (study 2).

Architecture	Simple dynamic memory		
Layer	Input state	Input	Output
Units	12	12	2
Internal connections	One-to-one with inputs units (feedback)	Fully with input state layer	Fully with hidden layer
External connections		One-to-one with robot sensors	One to-one with robot motor inputs
Activation Function	Identity		Logistic
Bias			No

Table 10: Details of NARX network (study 2).

ARCHITECTURE	NARX			
Layer	Plan	Input	Hidden	Output
Units	2(t-1), 2(t-2), 2(t-3) [1].	12	4	2 motor
Connections (internal)	One-to-one with output layer units (feedback)		Fully interconnected with input and plan units	Both fully interconnected with hidden units; both one-to-one with plan units
Connections (external)		One-to-one with sensors (8 rangefinders, 2 photoreceptors)		One-to-one with robot motor units
Activation Function	Identity		Logistic	Logistic
Bias units			No	No

[1]Feedback to plan units was subject to multiple delays.

9.3.5 Training and testing details

The special training scheme devised for this approach to delayed response learning had procedures that were common to both studies. Firstly, continuous samples of the robot's sensory inputs and corresponding motor responses were collected automatically. For this, the trobot was moved manually (that is, by using direction keys on the computer keyboard) along the centre of the aisle in each of the two associated environments (mirror-image in Study 1, pattern-differentiated in Study 2). Secondly, the controllers were trained initially (referred to as *first phase training*) on sets generated in each environment and tested in the same environment. This procedure was followed in order to confirm that the appropriate behaviour had been learned in response to each distinct stimulus, for example, turning right after a notch on the left-hand side. As expected, learning these sub-tasks proved to be a relatively trivial problem for all the controllers tested. Initial trials with the enhanced SRN led to successful parameter settings being adopted as common settings for all the architectures. During first phase testing, the learning rate was 0.3 and the momentum was 0.9.

The common procedure for training on the overall tasks (referred to as *second phase training*) was initiated by concatenating the two training sets obtained from the paired environments for a given trial. For example, the set for turning right after a notch on the left-hand side and the set for turning left after a notch on the right-hand side. The controller was then trained using this expanded training set. The initial trials with the enhanced SRN (see above) had showed that parameter setting was far less straightforward than in first phase training. Ultimately common settings were adopted for both studies (a learning rate of 0.07 and momentum of 0.15).

In both of the training phases in each of the two studies, the following common parameter settings and were adopted. All weights were randomly initialised in the range [-0.1, 0.1]. The enhanced SRN was trained and tested with five different settings of the feedback gain (0.1, 0.3, 0.5, 0.7 and 0.9). The decay rate α (equation 9.3) on the simple dynamic network and the hybrid input state network was zero in both training and testing.

After second phase training was completed, the controllers were tested in each of the underlying paired environments and their ability to perform the required delayed response was assessed.

Study 1

In this single architecture study, only the capacity for long-term dependency, or memory depth, of the STM was tested. As it did not involve a go-signal, the robot was turned about its axis following a number of steps, starting when the instruction stimulus was no longer being detected (in the judgement of the observer). The total number of steps before the turning point was varied across multiple training sets to support investigation of the effective depth of STM. The number of steps varied between one and twenty.

Study 2

This study involved a go-signal. There was one training set for each instructional stimulus (because the interval between the instruction stimulus and the decision point was fixed at the mean value derived from the first set of trials – see above). For each

architecture the testing was repeated twenty times. The step on which the go signal was administered was incrementally increased from one to twenty across the trials

During the training phase, the simulated light beam was "flashed" for one time step at a fixed number of steps following extinction of the stimulus in the training. The number of steps chosen was based on the mean value of five steps obtained in Study 1 based on just the enhanced SRN. This was considered to represent the optimum depth of memory on which to base investigations into the ability to generalise across time when the controller was subsequently tested.

9.3.6 Results

Study 1

The simulation was run and observations were made of the behaviour of controllers trained on each of the training sets with varying numbers of steps between instruction stimulus and turning point. Results, averaged across trials are summarised in the chart in Figure 27. As there would be a 50 per cent chance of turning in either direction, the results have been re-calibrated to set chance-level performance at zero. Figure 28 shows a sequence of screen shots recording the robot's behaviour when it passes the instructional stimulus on the left and subsequently executes a right U-turn. The first two shots (viewed across the rows) indicate the gap between the robot's last perception of the stimulus and its commencement of the U-turn behaviour, in this case four time steps. Successful test runs were repeated using different lateral starting points for the robot. It was found that, with the exception of starting points very close to the wall containing the stimulus object, the successful behaviour was repeated. This eliminated the possibility that any arbitrary features of the robot's path were

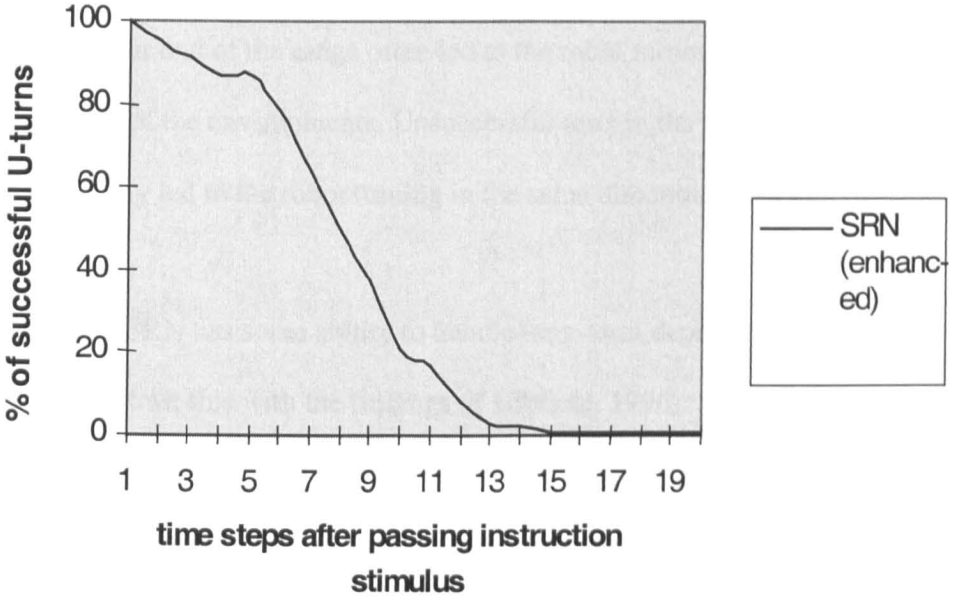

Figure 27: Graph showing performance of enhanced SRN architecture (study 1).

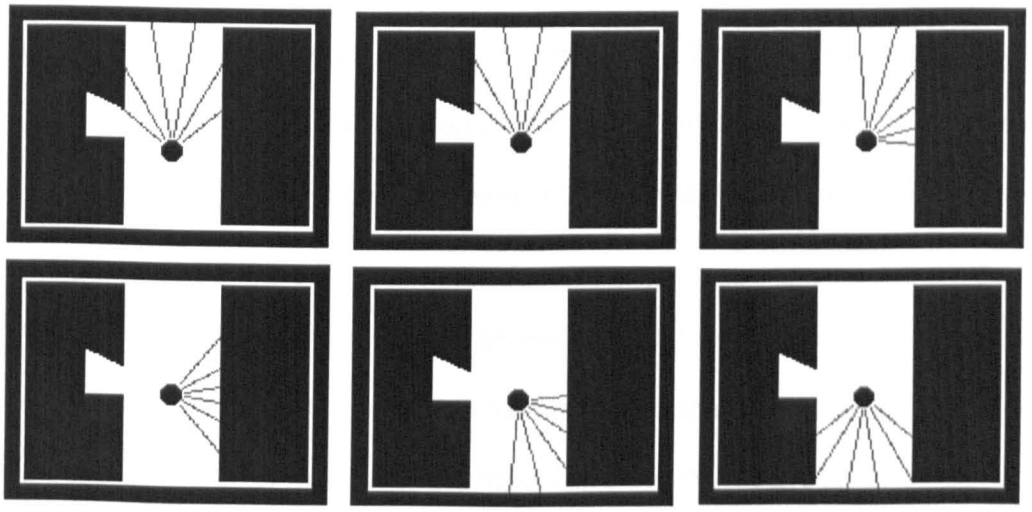

Figure 28: Sequence of turning movements (study 1)

responsible for its behaviour, and incidentally demonstrated the spatial generalisation capability of the ANN controller. It was observed that the longer the temporal dependency in the training set, the more erratic was the behaviour during testing. Tests at the far end of the range often led to the robot turning before it had passed the notch in one of the environments. Unsuccessful tests in the middle of the range usually merely led to the robot turning in the same direction in both environments.

Clearly, this SRN has some ability to handle long-term dependencies in a continuous domain (contrast this with the findings of Ulbricht, 1996).

Study 2

The comparative results are summarised, re-calibrated as before, in Figure 29. Using the go-signal to enable measurement of the ability to generalise across time, these experiments indicated that all the networks had some temporal generalisation ability. In the case of the enhanced SRN, a network gain setting of 0.3 produced the most noticeable effects but generalisation tended to disappear as the gain was increased. The comparative results shown are based on the SRN with this setting.

These results largely confirmed expectations that input level information is most useful to the agent, but the reasonable performance of the NARX network (with three output feedback delay lines) was surprising at least in the relative sense. However, the generalization ability overall was somewhat disappointing, particularly in the case of the hybrid network – certainly this was markedly inferior to the performance of Ulbricht's input state network (see section 4.2). It is possible that the highly specialized architecture, input pre-processing technique and training procedure

together with the conventional connectionist interpretive bias at the input level were responsible for the latter's superiority. Certainly, it was not possible to achieve the same level of performance, at the lower level of sensorimotor abstraction prevailing in these studies.

Although the results reported are interesting, they must be seen as little more than a suggestion of how the much longer-term aims envisioned in section 8.3 might be approached initially, with existing or hybrid models. Clearly, performance limitations that have been noted in more conventional connectionist domains remain a problem in SAB studies.

9.4 Concluding observations

It will be noted that in the studies described in this chapter the modular approach investigated earlier was abandoned. Instead, the studies focussed more minutely on

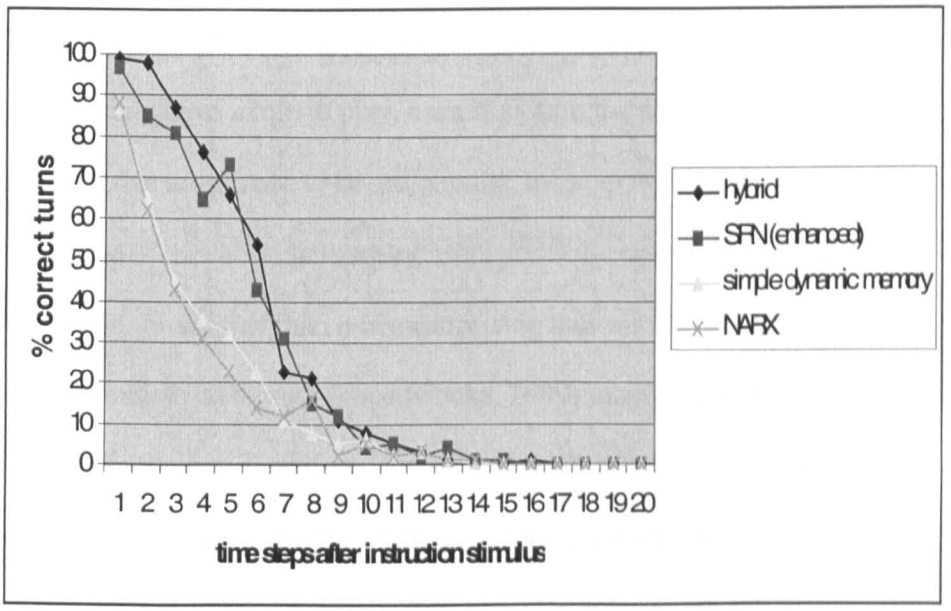

Figure 29: Results summarised for study 2.

the temporal processing capabilities of monolithic architectures rather than on their ability to fulfil a certain task-oriented requirement as in the action selection problem described in Chapter 6. However, this did not represent a philosophical or design position as in the case of Ziemke (1996a, 2000). The belief in the long-term need for a modular approach was retained (cf. Dorffner, 1997). The change of focus simply reflected the need to step back to look more closely at the nature of temporal processing itself and its essential role in the development of autonomous agents. For similar reasons, supervised learning was used as a temporary, enabling measure. The approach need not attract the criticism that it has received in the context of traditional connectionism (for example Bickhard and Terveen, 1995), nor the rather polemic, blanket disapproval of some workers (for example Gaussier and Zrehen, 1994). This kind of criticism seems to stem from a rather die-hard commitment to "situatedness" that brooks no degree of disconnection from "reality" even to facilitate the investigation of interesting mechanisms. That different learning rules, including supervised learning, probably coexist in real neural architectures is a view expressed by Ellis & Humphrey (1999). Consequently, it would seem premature to reject any that in principle may have a role to play, even if as here the training regimes employed in practice are certain to be quite unlike those in real neural systems. We are faced by some of the most challenging problems still confronting the cause of human knowledge. In view of this, it is unsurprising that we are nowhere near being able to build "complete creatures" (*pace* Brooks, 1989) in any sense that keeps the long term aims of AI alive. Therefore, any techniques that can be marshalled in the attempt to throw a little light on obscure processes may perhaps not be too sternly resisted on the grounds of purity or polemics, provided they do not raise overwhelming foundational concerns.

The initial experiments described in this section concentrated on the SRN in an attempt to determine whether its advantages in other contexts could be shown to persist in this version of Ulbricht's scenario. Despite her findings that it was unable to solve the two constituent problems, the conclusions were that, with some processing enhancements, it was able to demonstrate a limited ability to handle the recast version of these problems. This perhaps confirms her conclusion that an "inappropriate" training procedure may have been a factor in its failure, rather than its inadequacy as a general model for temporal processing.

9.5 Summary

In this chapter the notion was developed that situated autonomous agents need to be embedded in time as a prerequisite for enjoying the representational contents necessary for the programmatic aims of AI. It was argued that the temporally impervious substrates of symbolic AI and *nouvelle AI* could not achieve such temporal embedding. Drawing broad inspiration from neuroscience, the chapter focussed on the association between short-term memory processing areas and behaviour guided by internal representations. A framework for study was established in which recurrent neural network controllers could be compared using a simulated mobile robot to perform tasks that required a delayed response. Some studies where described to investigate existing models, an enhanced model and a model with some novel characteristics in simulations.

CHAPTER 10

CONCLUSION

10.1 Introduction

The inquiries into the simulation of adaptive behaviour reported in this thesis proceeded on two levels. On what may be termed the primary level, there were two distinct phases; the second level provided a bridged between them. In the first phase, the broad aim was to incrementally develop architectures integrating bottom-up learning and control for study in an essentially reactive setting. This was based on the opening discussion. It lent strength to the pragmatic view that ANNs offered a promising alternative to behaviour-based robotics in the attempt to surmount the problems of traditional AI (see also Rylatt et al., 1995 and Rylatt et al. 1998). The inquiries at the second, more philosophical level were seeded in that initial argument and germinated during the first phase of the inquiries. They took the form of a critique and statement of position on the question of how to move beyond reactive behaviour on a unified basis. From this position, the second phase of the inquiries went on to develop and investigate additional architectures for study in a delayed response setting.

10.2 Achievements and limitations

In the first phase, the aim was to address performance problems noted in subsection 2.4.5 using a modular approach on the lines of the pre-emption architecture (section

4.2) but based on reinforcement learning. The full goal of this phase was achieve a unified implementation of the sensorimotor and control levels within this scheme. Because of the special problems associated with the introduction of reinforcement learning, two distinct architectures with several variants were incrementally developed and studied. Firstly, a modular architecture (Crill) was demonstrated to achieve performance similar to the pre-emption architecture, avoiding the kind of cyclical behaviour afflicting subsumption–based agents in reactive control setting (Chapter 6, see also Rylatt et al, 1996). Underlying this demonstration, the principal advance was the application of CRBP learning in a modular architecture. However, this fell short of the full goal as it relied on an algorithm to provide overall control.

Incremental development of the first architecture led to MERGe, an architecture with the desired unified control approach (Chapter 7, see also Rylatt et al. 1997a and Rylatt et al. 1997b). This architecture was then studied in the same setting as the Crill architecture. Its substantially similar performance is considered to have satisfied the full goal of this phase. The noteworthy underlying advance was the integration of the SRN under the CRBP learning scheme into the existing ME model. At another level, the approach succeeded in restricting the influence of user/observer semantics on behaviour learning to the very coarse-grained definition of a modular framework within which sensory-modality related competences could emerge.

The second level of enquiry extended, and deepened the critique of *nouvelle AI* begun in section 2.5. In this, the foundational difficulties of behaviour-based robotics were more clearly identified as the reason why this approach cannot hope to cash the cheques written by Brooks (1990a, 1990b, 1992). The notion of embedding in time,

distinct from related arguments in focussing on the appropriateness of substrate, was developed as an extension of the idea of structural coupling from this critique (Chapter 8, see also Rylatt and Czarnecki, 1998). It represented a statement of position on the need for an appropriate, unified substrate to advance the simulation of adaptive behaviour significantly beyond reactive control. The conclusion drawn is that recurrent neural networks possessed features - not yet fully researched either in the context of the simulation of adaptive behaviour or of traditional connectionism - which made them promising as the mechanism for this substrate.

At the primary level of inquiry, the second and concluding phase stemmed from the critique and statement of position but also represented an incremental advance on the first phase in terms of the behaviour studied. It had the limited aim of establishing a framework for these new studies, in which the behaviour investigated would be delayed response. Additionally, a further new architecture was developed with characteristics expected to have particular advantages in the new setting. In the studies, it was compared with extant recurrent neural network architectures. It proved to have measurably superior performance (Chapter 9; see also Rylatt and Czarnecki, 1998; Rylatt and Czarnecki, 2000), though this was not sufficiently marked to remove the possibility that a radically different architecture might be needed in the end.

The foundational arguments advanced in Chapter 8 were intended to assist the development of the new AI paradigm based on ANNs free not only of traditional connectionist pseudo-symbolic bias but also of any lingering behaviour-based preconceptions. Without such foundational surety, this new AI, advocated in Dorffner

(1997), falls to critiques such as, for example (Bickhard and Terveen, 1995) which dismisses connectionism because it shares the foundational representational errors of encodingism. However, the issue of representation is still highly contentious. It is likely that an enormous amount of experimentation and philosophical debate will be required to even begin to settle the issue (for an up-to-date overview, from a perspective particularly relevant to the ideas in Chapter 8, see Ziemke, 1999). Linåker and Niklasson (2000) provide an example of some fascinating recent work in this area. Abandoning the principle of error minimisation in favour of so-called *change detection*, they show how compact representations of sensorimotor experiences can be extracted from the abstract sensory flows that emerge from agent-environment interaction. The work is based on the novel Adaptive Resource Allocating Vector Quantisation Network (ARAVQ). However, the conclusion that such representations should make it possible to solve the problem investigated in Chapter 9 (to which the authors refer), using reinforcement learning, is not actually demonstrated.

In conclusion, it will be clear that the overall contribution of these studies to knowledge cannot be neatly categorised as the construction of a single theory, application, artefact or methodology. It is offered in the context of the simulation of adaptive behaviour, an activity that is yet perhaps too diffuse to be considered a fully-fledged science. To this loose fascicle of knowledge, it has sought to contribute some new sticks of insight and some organically derived foundational glue. The integration of learning and control in one fine-grained medium, and the tentative identification of mechanisms therein that might support the grounding of representations, doubtless constitute only an infinitesimal part of the story that needs to unfold if the simulation of truly *cognitive* adaptive behaviour is to be demonstrated. In these studies, it only

remains to indicate how the work described might form the basis for further investigation.

10.3 Summary

The achievements and limitations of the work undertaken in these doctoral studies were summarised in this chapter. In the following final chapter, recommendations are made for further work.

CHAPTER 11

RECOMMENDATIONS

During the investigations, a number of interesting possibilities for further research were noted. The modular design of the MERGe controller described in Chaper 7 provides scope for further studies that might explore different network topologies within the architecture, for example SRNs as experts as well as in the gating network. An example of recent interest in this kind of approach should be noted in (Tani and Nolfi, 1998) where a hierarchical, recurrent mixture of experts architecture (RME) is described.

It might also be interesting to investigate the ability of a recurrent gating network to select appropriate types of expert for different tasks as in the experiments of Jacobs et al. (1991), with the added independent variable of temporal extension. Clearly, there is also scope for investigations that vary dimensionality, for example the number of units in various layers. It can however be noted in this respect that parametric investigations of this kind were undertaken by Meeden et al. (1994) using the monolithic Carbot architectures. These seemed fairly inconclusive, resulting in only very broad yardsticks – for example, "more is better". The dynamics of the recurrent networks require analysis so that observed behaviour can be more precisely attributed to activity at this level. However, analysis, in these terms, of other than very small networks is problematic. Experiments with such networks, however, may well be justified in order to carry out the "glass box" analysis that does not appear to be

applicable to networks with higher dimensionality (that is, hidden layers with more than three units). For examples of this kind of approach, with small hidden layers that permit 3D visual plots of activity, see Pollack (1995), Ziemke (1996a) and Sharkey et al. (1996). In particular, this approach might shed further light on the "modular versus monolithic" debate, in which the last named authors and at least Ziemke (1996b, 2000) appear to have an interest, by revealing the way in which the gating network partitions the control space. An alternative approach sometimes followed is principal components analysis (for example Elman, 1995 and Meeden et al, 1994), but it would appear to be best suited to more traditional connectionist systems (that is, where input representations have been pre-determined through a symbolic user-semantics).

The framework for study proposed in Chapter 9 could be used as the setting for further studies, based on existing and novel architectures. In particular, the suggestion of Linåker and Niklasson (2000) that a new approach not based on error minimisation would provide a more complete solution to this problem could be investigated.

BIBLIOGRAPHY

ACKLEY, D.H. and LITTMAN, M.L. (1990). Generalisation and scaling in reinforcement learning. In: D.S. Touretsky (Ed.) Advances in Neural Information Processing Systems 2. Morgan Kaufman, 551-558.

AITKEN, A.M. (1994). An architecture for learning to behave. *From Animals to Animats 3: Proc. of the Third Int. Conf. on the Simulation of Adaptive Behaviour*, 189-187, MIT Press, Cambridge, MA.

ALMASSY, N and VERSCHURE, P. (1992). Optimizing selg-organizing conttrol architectures with genetic algorithms: the interaction between natural selection and ontogenesis. *Proc. of the 2nd Conf. on Parallel Problem Solving from Nature.* Elsevier, 451-460.

ANDERSON. J.R. (1990). *Cognitive psychology and its implications.* W.H.Freeman, New York (3rd edition).

ARAUJO, E.G. and GRUPEN, R.A. (1996). Learning control composition in a complex environment. In P.Maes, M.J.Mataric, J-A Meyer, J.Pollack and S.W.Wilson (eds), *From Animals to Animats 4. Proc. of the Fourth Int. Conf. on the Simulation of Adaptive Behavior.* MIT Press, Cambridge, MA, 325-332.

ARKIN, R.C. (1990). Integrating behavioral, perceptual, and world knowledge in Reactive navigation. *Robotics and Autonomous Systems 6,* 105 –122.

ARKIN, R.C. (1998). *Behavior-based robotics.* MIT Press, Cambridge, MA.

BARTO, A.G. (1990). Some learning tasks from a control perspective. In L.Nade and D. Stein (eds.), *1990 Lectures in Complex Systems,* Santa Fe Institute Studies in the Science of Complexity, Lect. Vol. III, Addison Wesley, Redwood City, CA.

BEER, R.A. (1995). Computational and dynamical Languages for autonomous agents. In R.F. Port and T. van Gelder (eds), *Mind as Motion: Explorations in the Dynamics of Cognition*. MIT Press, Cambridge, MA.

BEER, R.D., CHIEL, H.J. and STERLING, L.S. (1990). A biological perspective on autonomous agent design. *Robotics and Autonomous Systems 6,* 169-186.

BICKHARD, M.H. and TERVEEN, L. (1995*). Foundational issues in artificial intelligence and cognitive science: impasse and solution.* North-Holland, Amsterdam.

BROOKS, R.A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2, 14-23.

BROOKS, R. A. (1989). How to build complete creatures rather than isolated cognitive simulators. In K. Van Lehn (ed.), *Architectures for Intelligence*, Erlbaum, Hillsdale, NJ, 225-239.

BROOKS, R.A. (1990). Elephants don't play chess. *Robotics and Autonomous Systems*, 6, 3-15.

BROOKS, R.A. (1991a). Intelligence without representation. *Artificial Intelligence*, 47, 139-159.

BROOKS, R.A. (1991b). Intelligence without reason. *Proc. of the Twelfth Int. Conf. on Artificial Intelligence* (IJCAI), vol. 1, 569-595.

BROOKS, R.A. (1994). Coherent behaviour from many adaptive processes. In: *From* Animals to *Animats 3: Proc. of the Third Internationa. Conf. on the Simulation of Adaptive Behavior.* MIT Press, Cambridge, MA, 22-29,

BROOKS, R.A. and MATARIC, M.J. (1993). Real robots, real learning problems. In: J.H. Connell and S. Mahadevan (eds.) *Robot Learning.* Kluwer Academic Publishers, 193-213.

BROOKS, R.A. and STEIN, L.A. (1994). Building bodies for brains. *Autonomous Robots,* 1:1, 7-28.

BUHLMEIER, A. (1994). Conditioning and robot control. *Proc. of the International Conf. on Artificial Neural Networks,* 1283-1286, Springer-Verlag, Berlin.

CHANDRASEKERAN, B. and JOSEPHSON, S.G. (1993). Architecture of intelligence: the problems and current approaches to solutions. *Current Science,* vol. 64, No. 6, 366-380.

CHESTERS, W. and HAYES, G. (1994). Connectionist environment modelling in a real robot. *From Animals to Animats 3: Proc. of the Third Int. Conf. on the Simulation of Adaptive Behaviour.* MIT Press, Cambridge, MA, 189-187.

CHOMSKY, N. (1957) *Syntactic structures.* Mouton, The Hague.

CHURCHLAND, P.M. (1996). Learning and conceptual change: the view from the neurons. In A.Clark and P.Millican (eds.), *Essays In Honour Of Alan Turing. Volume 2: Connectionism, Concepts And Folk Psychology,* Oxford University Press.

CHURCHLAND, P.S. and SEJNOWSKI, T.J. (1992). *The Computational Brain.* MIT Press, Cambridge, MA.

CLARK, A. (1993). *Associative engines: connectionism, concepts, and representational change.* MIT Press, Cambridge, M.A.

CLARK, A and WHEELER, M . (1998). Bringing Representation Back to Life. *From Animals to Animats 5: Proc. of the Fifth Int. Conf. on Simulation of Adaptive Behavior,* MIT Press, Cambridge, MA.

COLOMBETTI, M. and DORIGO, M. (1994). Training agents to perform sequential behaviour. *Adaptive Behaviour,* Vol. 2, No. 3, 247-275.

COPELAND, J. (1993). *Artificial Intelligence: A Philosophical Introduction.* Blackwell, Oxford.

COTTRELL, G.W. and TSUNG, F.S. (1991). Learning simple arithmetic procedures. In J.A. Barnden and J.B. Pollack (eds.), *High-Level Connectionist Models*, 305 – 321.

DORFFNER, G. (1997). Neural networks and a new AI - questions and answers, In G. Dorffner (Ed.), *Neural Networks and a New Artificial Intelligence*, International Thompson Computer Press.

DOYA, K. (1996). Recurrent networks: Supervised learning. In A. Arbib (ed.), *The Handbook of Brain Theory and Neural Networks.* MIT Press, Cambridge, MA.

EDELMAN, G. (1987). Neural Darwinism: the theory of neuronal group selection. Basic Books, New York

EDELMAN, G. (1992). *Bright air, brilliant fire.* Basic Books, New York.

ELLIS, R, and HUMPHREY, G.W. (Eds.) (1999). *Connectionist psychology: a text with readings.* Psychology Press, Hove, UK.

ELMAN, J.L. (1990). Finding structures in time. *Cognitive Science,* 14, 179-212.

ELMAN, J.L. (1995). Language as a dynamical system. In R.F. Port and T. van Gelder (Eds.) *Mind as Motion: Explorations in the Dynamics of Cognition.* MIT Press, Cambridge, MA.

FAGG, A.H., LOTSPEICH, D., and BECKEY, G.A. (1994). A reinforcement learning approach to reactive control policy design for autonomous robots. *Proc. of the IEEE Conf. on Robotics and Automation.*

FRANCHI, P., MORASSO and VERCELLI, G. (1994). A hybrid self-organizing architecture for autonomous mobile robots. *Proc. of the Int. Conf. on Artificial Neural Networks*, 1283-1286.

FRANKLIN, S. (1995). *Artificial Minds.* The MIT Press, Cambridge, MA.

FREEMAN, J. A. (1994). *Simulating neural networks.* Addison Wesley, Reading, MA.

FU, L. M. (1994) *Neural networks in computer intelligence.* McGraw-Hill, London.

GAUDIANO, P., ZALAMA, E., CHANG, C. and CORONADO, J.L. (1996). A model of operant conditioning for adaptive obstacle avoidance. In P.Maes, M.J.Mataric, J-A Meyer, J.Pollack and S.W.Wilson (eds), *From Animals to Animats 4, Proc. of the Fourth Int. Conf. on the Simulation of Adaptive Behavior.* MIT Press, Cambridge, MA, 325-332.

GAUSSIER, P. AND ZREHEN, S. (1994). A topological neural map for on-line learning. *From Animals to Animats 3: Proc. of the Third Int. Conf. on the Simulation of Adaptive Behaviour.* MIT Press, 275-281

GROSSBERG, S. (1971). On the dynamics of operant conditioning. *J. of Theoretical Biology,* 33, 225-255.

GUIGON, E. and BURNOD, Y. (1996). Short-term memory. In A. Arbib (ed.), *The Handbook of Brain Theory and Neural Networks.* MIT Press, Cambridge, MA.

HALLAM, B., HALPERIN, J.R.P., and HALLAM, J.C.T. (1994). *An ethological model of learning and motivation for implementation in mobile robots.* Research paper 629, Department of AI, Edinburgh University.

HARNAD, S.A. (1990). The Symbol Grounding Problem. *Physica* D 42, 335-346.

HARVEY, I., HUSBANDS, P. and CLIFF, D. (1993). Issues in Evolutionary Robotics. *From Animals to animats 2: Proceedings of the Second International Conference on the Simulation of Adaptive Behaviour.* MIT Press, 364-373.

HAYKIN, S. (1994). *Neural networks: A comprehensive introduction.* Macmillan, New York.

HEBB, D.O. (1949). *The organisation of behavior.* Wiley, New York.

JACOBS, R. A., JORDAN, M. I., NOWLAN, S. J. and HINTON, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, 3, 79-87.

JACOBS, R.A. (1988). Increased rate of convergence through learning rate adaptation. *Neural Networks*, 1(4), 295-308.

JACOBS, R.A. AND JORDAN, M.I. (1991). A competitive modular connectionist architecture. In: Tourezky, D.S. (Ed.) *Advances in Neural Information Processing 3*, Morgan Kaufmann, NY.

JAKOBI, N. (1998). The minimal simulation approach to evolutionary robotics, In *Proceedings of ER'98*, T. Gomi (ed.), AI Systems Books.

JORDAN, M.I. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. *Proc. of the 8th. Conf. on Cognitive Science*, 531-546.

JORDAN, M.I. and JACOBS, R.A. (1992) Hierarchies of adaptive experts. In J.E. Moody, S.J.Hanson, and R.P.Lippmann (eds.) *Advances in Neural Information Processing Systems* . Morgan Kaufmann, 985-992.

KADABA, N., NYGARD, K.E., JUELL, P.L. and KANGA, L. (1990). Modular backpropagation network for large domain pattern classification. *Proc. of the Int. Joint Conf. on Neural Networks (IJCNN)*, Vol. II., 551 - 554,.

KAELBLING, L.P. (1993). *Learning in embedded systems.* MIT Press, Cambridge, MA.

KAELBLING, L. P., LITTMAN, M. L. and MOORE A.W. (1996). Reinforcement learning: a survey. *Journal of Artificial Intelligence Research*, 4, 237-285

KHATIB, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research, 5:1, 90-98.*

KOHONEN, T. (1982). Self-organized formation of topographically correct feature maps. *Biological Cybernetics*, 43, 59-69.

KREMER, S.C. (1995). On the computational power of Elman-style recurrent networks. *IEEE Transactions on Neural Networks*, 6:4, 1000-1004.

LAKOFF, G. and JOHNSON, M. (1980). *Metaphors we live by*. Chicago.

LEMON, O. and NEHMZOW, U. (1998). The scientific status of mobile robotics: multi-resolution mapbuilding as a case study. *J. Robotics and Autonomous Systems*, 24, Vols. 1-2.

LIN, L-J. (1991). Programming robots using reinforcement learning and teaching. *Proc. of American Association for Artificial Intelligence Conf. (AAAI-91)*, 781-786.

LIN, L-J. (1993). Hierarchical learning of robot skills by reinforcement. *Proc. of IEEE Conf. on Neural Networks*, 181 - 187.

LIN, L.-J. and MITCHELL, T.M. (1994). Reinforcement learning with hidden states. *From Animals to Animats 2: Proceedings of the Second Int. Conf. on the Simulation of Adaptive Behaviour*. MIT Press, Cambridge, MA, 271 – 280.

LIN, T., HORNE, B.G., TINO, P. and GILES, L. (1996). Learning long-term dependencies is not as difficult with NARX recurrent networks. *IEEE Transactions on Neural Networks*.

LINÅKER and NIKLASSON. (2000). Extraction and inversion of abstact sensory flow representations. In J. Arcady-Meyer, A. Berthoz, D. Floreano, H. Roitblat and S.W. Wilson (eds.) *From Animals to Animats 6: Proc. of the Sixth Int. Conf. on the Simulation of Adaptive Behavior*. MIT Press, Cambridge, M.A.

LUDIK, J., PRINS, W., MEERT, K. and CATFOLIS, T. (1997). A comparative study of fully and partially recurrent Networks. *Proc. of the 1997 IEEE Int. Conf. on Neural Networks, Houston (ICNN97),* Texas, Vol. I, 292-297.

MACFARLAND, D. and BÖSSER, T., (1993). *Intelligent Behaviour in Animals and Robots* , MIT Press.

MAES, P. (1995). Modeling adaptive autonomous agents. In C.G. Langton (ed.) *Artificial Life: An Overview.* MIT Press, Cambridge, MA. 135-162.

MAES, P. and BROOKS, R.A. (1990). Learning to co-ordinate behaviours. In *Proc. of American Association for Artificial Intelligence Conf. (AAAI-90).* 796-802.

MAHADEVAN, J.H. and CONNELL, J. (1993). *Robot Learning.* Kluwer Academic Press, 1993.

MATARIC, M.J. (1990). A distributed model for mobile robot environment learning and navigation. *Technical Report 1228,* AI Lab, MIT.

MATARIC, M.J. (1992). Integration of behaviours into goal-driven, behavior-based robots. *IEEE J. of Robotics and Automation* 8, 3, 304-312.

MEEDEN, L. (1993*). Towards planning, incremental investigations into adaptive robot control.* Ph.D. Thesis, Department of Computer Science, Indiana University.

MEEDEN, L., McGRAW, G. and BLANK, D. (1994). Emergent Control and Planning in an Autonomous Vehicle. *Proc. of the Fifteenth Ann. Conf. of the Cognitive Science Society.*

MICHEL, O. (1996). *Khepera simulator version 1.0 – user manual.* University of Nice-Sophia Antipolis, Valbonne, France.

MILLAN, J. DEL R. (1994). Learning efficient reactive behavioural sequences from basic reflexes. From Animals to Animats 3: *Proc. of the Third Int. Conf. on the Simulation of Adaptive Behaviour.* MIT Press, Cambridge, MA, 266-274

MOZER, M.C. (1993). Neural net architectures for temporal sequence processing, In: A.Weigend and N.Gerschenfeld (eds.) *Predicting the Future and Understanding the Past.* Addison Wesley.

NEHANIV, C., DAUTENHAHN, K. and LOOMES, J.M.J. (1999). Constructive biology and approaches to temporal grounding in post-reactive robotics. In G.T. McKEE and P. SCHENBER (Eds.), *Sensor Fusion and Decentralised Control in Robotic Systems II* (Sept 19-20, 1999, Boston, Mass). *Proc. of SPIE*, Vol 38-39, 156-167.

NEHMZOW, U. (1992). *Experiments in competence acquisition for autonomous mobile robots.* PhD Thesis, University of Edinburgh, U.K.

NEHMZHOW, U. and McGONIGLE, B. (1994). Achieving rapid adaptations in robots by means of an external tutor. In: *From Animals to Animats 3: Proceedings of the Third International Conference on the Simulation of Adaptive Behavior.* MIT Press, 275-281

NEHMZOW, U. (1995). Flexible control of mobile robots through autonomous competence acquisition. *Measurement and control*, vol. 28, 48-54.

NEHMZOW, U., HALLAM, J. and SMITHERS, T. (1989). Really useful robots. In Kanade, T., Groen, F.C.A. and Hertzberger, L.O. (eds.) *Intelligent Autonomous Systems 2*, Elsevier, Amsterdam.

NEHMZOW, U., SMITHERS, T and MCGONIGLE, B. (1993). Increasing behavioural repertoire in a mobile robot. In : J.A. Meyer, H. Roitblat, and Wilson, S. (eds). *From Animals to Animats 2, Proceedings of the Second*

International Conference on the Simulation of Adaptive Behaviour. MIT Press, 291-297.

NEWELL, A. and SIMON, H.A. (1976). Computer science as empirical inquiry: symbols and search. In: Haughland, J. (ed.) *Mind Design: Philosophy, Psychology, Artificial Intelligence.* MIT Press, Cambridge, MA.

PAL, P.K. and KAR, A. (1996). Sonar-based mobile robot navigation through supervised learning in a neural net. *Autonomous Robots*, 3. 355-374.

PESCHL, M.F. (1995). Autonomy vs. environmental dependency in neural knowledge representation. In R.A. Brooks and Pattie Maes (eds), *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, 417-423.

PFEIFER, R. and SCHEIER C. (1994). From perception to action: the right direction? In: P.Gaussier and J-D. Nicoud, (Eds) *From perception to action.* IEEE Computer Society Press, 1-11.

POLLACK, J.B. (1995). The induction of dynamical recognisers. In: R.F. Port and T. van Gelder (eds), *Mind as Motion: Explorations in the Dynamics of Cognition.* MIT Press, Cambridge, MA.

PORT, R.F., CUMMINS, F. and MCAULEY, J.D. (1995). Naive time, temporal patterns and human audition. In: Port, R.F. and van Gelder, T. (eds), *Mind as Motion: Explorations in the Dynamics of Cognition.* MIT Press, Cambridge, MA.

RUMELHART, D.E., HINTON, G.E. and WILLIAMS, R.J. (1986). Learning internal representations by error propagation. In: D.E.Rumelhart, J.E.McLelland and the PDP Research Group, *Parallel Distributed Processing, (Vol. 1, Foundations).* MIT Press, Cambridge, MA, 318-362.

RYLATT, M. and CZARNECKI, C. (1998). Beyond physical grounding and naïve time: investigations into short-term memory for autonomous agents. In: R.Pfeifer, B.Blumberg, J-A. Meyer and S.W.Wilson (eds.) *From Animals to Animats 5: Proc. of the Fifth Int. Conf. on Simulation of Adaptive Behavior.* MIT Press, Cambridge, MA. 22-31.

RYLATT, R.M. and CZARNECKI, C. (2000). Embedding connectionist autonomous agents in time: the 'road sign problem'. *Neural Processing Letters*, 12:2. Kluwer Academic Publishers, 145-158.

RYLATT, M., CZARNECKI, C. and ROUTEN, T. (1998). Connectionist learning in behaviour-based mobile robots: a survey. *Artificial Intelligence Review*, 12:6, Kluwer Academic Publishers, 445-468.

RYLATT, R. M., CZARNECKI, C. A. and ROUTEN, T. W. (1995). A perspective on the future of behaviour-based robots. *Mobile robotics workshop notes of the Tenth Biennial Conference on AI and Cognitive Science*, Sheffield, U.K

RYLATT, R. M., CZARNECKI, C. A. and ROUTEN, T. W. (1996a). Learning behaviours in a modular neural net architecture for a mobile autonomous agent. *Proc. of the First Euromicro Workshop on Advanced Mobile Robots*, Kaiserslautern, 84-88.

RYLATT, R.M., CZARNECKI, C..A. and ROUTEN, T.W. (1997a). A partially recurrent gating network approach to learning action selection by reinforcement. *Proc. of the 1997 IEEE Int Conf. on Neural Networks, Houston (ICNN97)*, Texas, Vol. III , 1689-1698.

RYLATT, R.M., CZARNECKI C.A and ROUTEN T.W. (1997c). Towards the Autonomous Control of Mobile Robots by Connectionist Experts. *Proc. Of the 5th IEE Int. Conf. on Artificial Neural Networks*

SAUNDERS, G.M., KOLEN, J.F., and POLLACK, J.B. (1994). The importance of leaky levels for behaviour-based A.I. *From Animals to Animats 3: Proceedings of the Third International Conference on the Simulation of Adaptive Behavior*,. MIT Press, Cambridge, MA., 275-281

SCUTT, T. (1994). The five neuron trick. *From Animals to Animats 3: Proceedings of the Third International Conference on the Simulation of Adaptive Behaviour*, 365-369, MIT Press, Cambridge, MA.

SHARKEY, N.E., HEEMSKERK, J. and NEARY, J. (1996). Subsuming behaviours in neural network controllers. *Proceedings of ROBOLEARN96*, Florida, July.

SCHEIER, C. and PFEIFER, R. (1998). Exploiting Embodiment for Category Learning. *From Animals to Animats 5: Proc. of the Fifth Int. Conf. on Simulation of Adaptive Behavior* , MIT Press, Cambridge, MA.

SMOLENSKY, P. (1988). On the proper treatment of connectionism. *Behavioural and Brain Sciences*, 11, 1-73.

SPIER, E. and MCFARLAND, D. (1998). Learning To Do Without Cognition *From Animals to Animats 5: Proc. of the Fifth Int. Conf. on Simulation of Adaptive Behavior* , MIT Press, Cambridge, MA.

STEELS, L (1994). The artificial life roots of artificial intelligence. *Artificial Life Journal*, Vol. 1, 1, MIT Press, Cambridge, MA.

STEELS, L. (1995). When are robots intelligent autonomous agents? *J. of Robotics and Autonomous Systems*, 15, 3-9.

STEIN, L.A. (1995). Imagination and situated cognition. In: K.M. Ford, C.Glymour and P.J. Hayes (Eds.) *Android Epistemology*, AAAI Press, Menlo Park, C.A.

SUTTON, R.S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, vol. 3, 9-44.

SUTTON, R.S. (1996). Generalization in reinforcement learning: successful examples using sparse coding. *Advances in Neural Information Processing Systems 8.* MIT Press, Cambridge, MA., 1038-1044.

SUTTON, R.S. and BARTO, A.G. (1998). *Reinforcement learning: an introduction.* MIT Press, Cambridge, MA.

TANI, J. (1996). Model-based learning for mobile robot navigation from the dynamical systems perspective, *IEEE Transactions on Systems, Man, and Cybernetics (Part B: Cybernetics)*, 26:3.

TANI, J. and FUKUMURA, N. (1994). Learning goal-directed sensory-based navigation of a mobile robot. *Neural Networks,* 7:3, 553-563.

TANI, J. and NOLFI, S. (1998). Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems. In: R.Pfeifer, B.Blumberg, J-A. Meyer and S.W.Wilson (eds.) *From Animals to Animats 5: Proc. of the Fifth Int. Conf. on Simulation of Adaptive Behavior.* MIT Press, Cambridge, MA., 270-279.

THRUN, S. (1994). A lifelong learning perspective for mobile robot control, *Proc. of the IEEE Conf. on Intelligent Robots and Systems.* 12-16.

TOATES, F. (1994). What is cognitive and what is *not* cognitive. *From Animals to Animats 3: Proceedings of the Third International Conference on the Simulation of Adaptive Behavior.* MIT Press, Cambridge, MA., 275-281.

ULBRICHT, C. (1996). Handling time-warped sequences with neural networks. *From Animals to Animats 4: Proc. of the Fourth Int. Conf. on Simulation of Adaptive Behaviour.* MIT Press, Cambridge, MA., 180-192.

VARELLA, F.J., THOMPSON, E. and ROSCH, E. (1993). *The embodied mind: cognitive science and human experience.* MIT Press.

VERSCHURE, P.F.M.J. (1992). Taking connectionism seriously: the vague promise of subsymbolism and an alternative. *Proc. of the Fourteenth Annual Conf. of the Cognitive Science Society*. Elbaum, N.Y.

VERSCHURE, P.F.M.J. AND PFEIFER, R. (1993). Categorisation, representations and the dynamics of system-environment interaction: a case study in autonomous systems. *From Animals to Animats 2: Proceedings of the Second International Conference on the Simulation of Adaptive Behaviour*.MIT Press, Cambridge, MA, 210-217.

VERSCHURE, P.M. (1997). Connectionist explanation: taking positions in the mind-brain dilemma. In: G.Dorffner (Ed.) *Neural Networks and a New Artificial Intelligence*, International Thompson Computer Press, 134-187.

WATKINS, C.J.C.H and DAYAN, P. (1992). Q-Learning. *Machine Learning*, 8(3).

WILES, J. and ELMAN J. L. (1995). Learning to count without a counter: a case study of dynamics and activation landscapes in recurrent networks. *Proc. of the Seventeenth Annual Conf. of the Cognitive Science Society.* MIT Press, Cambridge, MA.

WILLIAMS, R.J. (1988). On the use of backpropagation in associative reinforcement learning. *Proc. of the IEEE Int. Conf. on Neural Networks*, 263-270.

WINOGRAD, T. and FLORES, F. (1986). *Understanding Computers and Cognition: A New Foundation for Design.* Ablex Publishing Corporation, Norwood, NJ.

ZIEMKE, T. (1996a). Towards adaptive behaviour system integration using connectionist infinite state automata. *From Animals to Animats 4: Proc. of the Fourth Int. Conf. on the Simulation of Adaptive Behaviour*, MIT Press, Cambridge, MA, 145-154

ZIEMKE, T. (1996b). Towards adaptive perception in autonomous robots using second-order recurrent networks. *Proceedings of the First Euromicro Workshop on Advanced Mobile Robots*, Kaiseslautern, 89-98.

ZIEMKE, T. (1999). Re-thinking grounding. In: A.Riegler, A.vom Stein and M.Peschle (Eds.) *Does Representation Need Reality?* Plenum Press, NY.

ZIEMKE, T. (2000). On 'parts' and 'wholes' of adaptive behavior: functional modularity and diachronic structure in recurrent neural robot controllers. In J. Arcady-Meyer, A. Berthoz, D. Floreano, H. Roitblat and S.W. Wilson (eds.) *From Animals to Animats 6: Proc. of the Sixth Int. Conf. on the Simulation of Adaptive Behavior.* MIT Press, Cambridge, M.A