Manuscript accepted for publication in *Robots and Autonomous Systems*, Final version, September 13, 1999. (Original submission: 31 October 1998).

ARBIB: an Autonomous Robot Based on Inspirations from Biology

R.I. Damper*, R.L.B. French* and T.W. Scutt†

*Image, Speech and Intelligent Systems Research Group,
Department of Electronics and Computer Science,
University of Southampton,
Southampton SO17 1BJ,
UK.

†Core Design Ltd.,
55 Ashbourne Road,
Derby DE22 3FS,
UK.

(emails: rid@ecs.soton.ac.uk, rlbf98r@ecs.soton.ac.uk and TomS@Core-Design.com)

Abstract

Simple artificial creatures (‘animats’), which operate as autonomous, adaptive robots in the real-world, can serve both as models of biology and as a radical alternative to conventional methods of designing intelligent systems. We describe the evolution and implementation of the autonomous robot ARBIB, which learns from and adapts to its environment. A primary goal was to test the notion that effective robot learning can be based on neural habituation and sensitization, so validating the suggestion of Hawkins and Kandel that (associative) classical and ‘higher order’conditioning might be based on an elaboration of these (non-associative) forms of learning. Accordingly, ARBIB’s ‘nervous system’ has a non-homogeneous population of spiking neurons, and learning is by modification of basic, pre-existing (‘hard-wired’) reflexes. By monitoring firing rates of specific neurons and synaptic weights between neural connections as ARBIB learns from its environment, we confirm that both classical and higher-order conditioning occur, leading to the emergence of interesting and ecologically valid behaviors.

1 Introduction

Ever since the pioneering experiments of Grey Walter [77, 78] on *Machina speculatrix* (an electromechanical ‘tortoise’) almost half a century ago, biologists, psychologists and systems scientists have sought to understand links between neural hardware and behavior by building and observing simple artificial creatures – ‘animats’ to use Wilson’s [81] term. In 1984, Braitenberg [10] published the small but now-famous book *Vehicles: Experiments in Synthetic Psychology*. In it, he described *gedanken* experiments in which simple robotic ‘creatures’ – in spite of their simplicity – displayed interesting, apparently intentional behaviors of light-seeking, obstacle-avoidance etc. Starting with a trivially simple vehicle with a single sensor of the external environment whose output drives a motor, hence propelling the vehicle, the abilities of such ‘creatures’ are incrementally expanded by the addition of more wheels and sensors, and adding features to the rudimentary ‘nervous system’ connecting them. Key ideas were the way that complex behavior could result from a very simple physical substrate, the possibilities for absorbing new concepts and generalizing to new situations offered by memory (*Mnemotrix*), and the extreme difficulty (if not impossibility) of understanding the “hidden mechanism” from its observed behavior. Braitenberg’s book has had an enormous impact on ideas about the way that an artificial intelligence might emerge through interaction of a machine with its environment. There is now widespread acceptance of the notion that, to quote McFarland and Bösser [49, p. 271] “intelligence requires a body” (see also [22]).

Bottom-up synthesis of animats holds much promise for achieving engineering systems of greater sophistication than can be designed by classical methods. Again, quoting McFarland and Bösser [49, p. 2]: “Recently, it has been realized that robots, which are still at a very primitive stage of their evolution, might better be designed along the zoological lines of primitive animals than along the traditional lines of autocratic control” (see [1] and [8] for some exemplary work in this vein). Just as important, animat studies can also provide models of emergent behavior in biological systems [69]. In particular, since the robot has been designed and built by the investigator, there is at least no mystery surrounding its mechanism.

We seek to extend the pioneering work of Braitenberg in two ways. First, no physical systems were ever built by him. This is an obvious omission, especially in light of his remark that observed behavior often "goes beyond what we had originally planned". (How does he know this?) There is currently significant research activity studying the behaviors of robots constructed according to the principles of 'synthetic psychology', with Brooks [11]–[15] probably the acknowledged flag-bearer of this field. Second, Braitenberg was not at all explicit about the relation of the mechanisms of learning in his vehicles to known details of nervous-system function. While it is "pleasurable and easy to create little machines that do certain tricks" [10], it is important in our view for these creations to bear some systematic relation to the realities of biology and psychology if they are to tell us anything about natural, intelligent systems. In this view, we are influenced by Barlow's [7] famous 'neuron doctrine', or 'psychophysical linking principle', that "... active high-level neurons *[might]* directly and simply cause the elements of our perception", so that it makes eminent sense to examine relations between the activity of single neurons (or small neuron assemblies) and learned behavior. Hence, we try to make these relations explicit in our implementation(s) so that we effectively model the robot's nervous system at a much lower (i.e. biologically faithful) level than in other animat studies. Learning behavior is based on neural modeling of habituation, sensitization and classical conditioning – simple forms of learning which have long been recognized by biologists and psychologists. The result is ARBIB: an **A**utonomous **R**obot **B**ased on **I**nspirations from **B**iology. Our robot is named in honor of cyberneticist Michael Arbib, well known for his seminal book [3].

Against this background, the specific goals of this work are:

1. to test the notion that effective robot learning can be based on neural habituation and

sensitization. Furthermore, we seek to validate the suggestion of Hawkins and Kandel [35, 34] that (associative) classical conditioning – and ‘higher’ forms of learning – might be based on the same neural mechanisms as (non-associative) habituation and sensitization.

2. to observe the emergent behavior of the ARBIB robot when provided with the potential to learn from its environment using these neural mechanisms.

3. in line with the neuron doctrine, to relate any interesting emergent behaviors back to details of ARBIB’s neural circuitry.

Accordingly, the remainder of this paper is structured as follows. In Section 2 immediately following, we justify the level of abstraction at which we simulate and implement the learning behavior of ARBIB. Section 3 describes the implementation of the neural simulator and, in Section 4, we then describe the biological motivations for the forms of learning embodied in the robot. Section 5 details ARBIB’s nervous system arrived at on the basis of simulation, before describing three physical implementations of the animat (Section 6) and their resulting behavior (Section 7). Finally, Section 8 concludes with discussion of the implications of this work for embodied artificial intelligence (AI) and the emergence of intelligent, purposeful behavior in autonomous systems.

2 Levels of Abstraction in Robot Learning

In this section, we consider the most appropriate level of abstraction at which to design and implement the nervous system of an autonomous robot. We should make it clear that the field is by no means mature enough for there to be any inalienable rights or wrongs in these matters. Unavoidably, therefore, what follows has something of the flavor of a manifesto.

2.1 Physiologically Faithful Modeling

One obvious approach to modeling a robot nervous system would be to incorporate most, if not all, known details of single-nerve function, and to wire up the resulting single-nerve models in accordance with real nervous systems. The problems with this physiologically faithful approach (exemplified by MacGregor [45]) are many and daunting. The complexity of even single nerve cells, when considered at a biochemical or electrophysiological level, means that interconnections of only a very few neurons can be simulated in present-day real-time embedded systems – making such models a tool for the physiologist rather than for the system scientist. Also, the required neuroanatomical knowledge of detailed patterns of wiring is mostly absent. Another problem, pointed out by Sejnowski, Koch and Churchland [65], is that: "As the model is made increasingly realistic by adding more variables and parameters, the danger is that the simulation ends up as poorly understood as the nervous system itself." For these reasons, we do not pursue this approach here, although it remains attractive for the future (given the rapid technological development of computing machinery).

2.2 Parallel Distributed Processing

In recent years, parallel distributed processing (PDP) or 'connectionism' [58] has become the majority paradigm in cognitive modeling and (non-symbolic) AI. It must, therefore, be taken seriously as an approach to learning in autonomous robots. For our purposes, we will consider PDP systems to be characterized by modeling at the level of activation signals (rather than spikes or some lower level still). Also, a homogeneous population of neurons is typical (but not universally so) in PDP systems. Of course, real systems of neurons are not homogeneous, but have different neuron types specialized to particular functions [50].

2.2.1 Supervised learning

PDP's greatest successes obviously lie with supervised back-propagation learning [21, 59]. In our view, however, conventional feed-forward artificial neural networks trained on back-propagation are of limited use for developing adaptive mobile robots for several reasons.

First, they rely on an external training signal ('teacher') of uncertain origin: Crick [24] describes back-propagation as "rather far-fetched as a biological learning mechanism". Second, as stated by Sharkey and Heemskerk [68, p. 177] "... real-time neural control systems that make no distinction between a learning and a performance phase are promising candidates for the control of adaptive behavior", yet back-propagation requires a prior training phase after which learning cannot easily be modified incrementally. Autonomous agents should not need to be pre-trained: they should discover their world and adapt 'on-line' to it, by themselves. Finally, back-propagation models are typically passive, open-loop and ungrounded (see [33] for the classic statement and discussion of the problem) in the sense that they are unattached to the environment: output from a network generally has no effect on the network's environment. Thus, their semantic interpretation can only be left to a human observer or homunculus (see the recent work of Billard and Dautenhahn [9]). Compare this with a simple invertebrate nervous system or the 'nervous system' of a Braitenberg vehicle. In these cases, the output of the network makes a difference to the world, and changes the input that the network receives. Learning typically involves associating two inputs rather than associating an input with an output, and the input training set (if it can be called that) is determined by both the environment and the motor output of the network.

These objections have not, however, prevented the use of back-propagation in training mobile robots (e.g. [53]). At worst, this has led to the faintly ridiculous scenario (as far as autonomous agents are concerned) of the teacher/experimenter following the robot around and pressing a button when it makes an error! Back-propagation has also been used to train subsumption architectures [36, 68], which were pre-programmed or fixed in the original work of Brooks [11].

2.2.2 Unsupervised learning

Many of the above objections can be removed if robot training is unsupervised, particularly if it is also incremental so that learning and performance phases can be integrated on-line rather than being separate. Self-organizing feature maps [42] have proved popular (e.g. [5, 54]) in view of their ability to extract features from raw sensory data. However (because of the way that the winning-neuron neighborhood is varied during learning), incremental training is still not directly achieved.

Gaudiano and coworkers [29, 30] have also described mobile robot control using unsupervised neural network learning based on the earlier biological sensory-motor control models of Bullock and Grossberg [17] and Gaudiano and Grossberg [28]. This work addresses the question of how an autonomous system can learn ("by doing") about its own sensory-motor endowments and their grounding in the complex and changing external world. Although having stronger biological motivations than typical PDP work, the control models are still (to borrow a phrase from Hawkins and Kandel [35, p. 376]) "hypothetical ... " rather than "based ... on known neural mechanisms" at such a low level as habituation and sensitization.

2.3 Computational Neuroscience

Computational neuroscience [65] attempts to exploit the relative strengths of the two paradigms just mentioned – neurobiological modeling and parallel distributed processing – by linking some of the principles of connectionism with data from experimental neurophysiology. This is the approach we favor, as it allows an appropriate trade to be made between biological fidelity and computational expediency. Thus, we aim to model ARBIB's nervous system at a level of abstraction intermediate between the PDP style (which largely abandons biological fidelity almost at the outset) and the physiologically faithful modeling of real neural systems. A primary goal was to model the learning at a level sufficiently close to neurophysiology that the links between ARBIB's adaptive behavior and its neural endowments might have some very direct and obvious relevance to real biological creatures, in line with Barlow's neuron doctrine.

Certain simple lower animals with only very rudimentary nervous systems still exhibit interesting behaviors, adapting to and surviving in their environment. Studies of small systems of neurons in such animals have revealed much about the relationship between neurophysiology and behavior [40]. The simplicity of these small systems allows the observer to see how each individual neuron in a circuit contributes to behavior. However, the behaviors so far examined in this way form only a tiny fragment of an extensive and complex repertoire. The gill-withdrawal reflex in the marine snail *Aplysia*, for example, has been extensively studied [18, 40] (and even modeled in CMOS VLSI circuitry [20]) but this animal is also involved in activities of locomotion, finding food, eating, escaping from predators (including inking), mating, etc. Enough has been learned, however, to make it worthwhile to link some of the principles of connectionism with data from experimental neurophysiology, in accordance with the computational neuroscience paradigm.

Our approach has been to draw inspirations eclectically from across a range of biological systems. There is no attempt to base ARBIB specifically on either a vertebrate or invertebrate animal. Rather we aim only to use general biological principles, noting that habituation and sensitization are ubiquitous processes in vertebrate and invertebrate nervous systems.

3 Implementation of Neural Simulator

We have previously developed a program capable of simulating small systems of neurons [62]–[64] at various levels of abstraction. The neural simulator is called Hi-NOON (**Hi**erarchical **N**etwork of **O**bject-**O**riented **N**eurons). In Hi-NOON, synapses, neurons and networks are all represented as objects within an object-oriented hierarchy. This section briefly describes the methodology of its design and details its working.

In line with the computational neuroscience approach, we have used the membrane potential (strictly, transmembrane potential difference) as the observable parameter in the network model. This is a much lower-level approach than the use of ‘activation’ values roughly corresponding to the spike or action potential (AP) rate of individual neurons or collections of neurons as in PDP models. By contrast, Hi-NOON allows us explicit knowledge of the information processing occurring in the cell, much of which is lost in the traditional connectionist approach. More important, Hi-NOON facilitates use of a non-homogeneous population of neurons.

The original program was written in object-oriented Pascal, but has subsequently been rewritten in C using the disciplines of object-oriented programming (OOP) [23, 27]. C was used (rather than C++ with its explicit support of OOP features) to maximize portability among the various hardware/software realizations of ARBIB. The benefits of the OOP approach are two-fold. First, the ability for objects to inherit properties from other objects means that it is easy to define more physiologically exact neurons in terms of simpler neurons. (In fact, the system allows a simple threshold unit as the most basic type of object.) More complex objects inherit certain properties from this object (e.g. the fact that it has weighted connections to other objects). The second benefit of OOP is polymorphism. This means that the network may contain many different types of neuron, at many levels of complexity, without the programmer having to be concerned with this.

3.1 Neuron Parameters

Basic neurophysiology suggests those attributes a model spiking neuron must have. The fixed parameters `BaseMP`, `Threshold` and `TimeConst` correspond to the resting potential, threshold and time constant of the neuron, respectively. There are also dynamic parameters MP, SynPot and fired (a 1/0 predicate) which model the actual membrane potential as it varies in time, accumulate the weighted sum of synaptic inputs which influence the updating of MP at the next time step, and indicate if the object is in the process of firing, respectively. This parameter system allows us easily to describe differences between neurons and to keep track of the changing states of neurons over time.

3.2 Networks, Neurons and Synapses as Objects

The neural network is held as a list of objects, where each such object corresponds to a single neuron and holds all the information about its state (see below) and about subsidiary objects. The information held in the neuron object is comprised of:

- a set of parameters which defines the neuron;
- a set of data structures which defines the 'axon terminals' for the neuron, each of which is itself an object and has its own parameters;
- a set of methods – pointers to functions, i.e. pieces of code which access and alter parameter values and so determine exactly how the neuron functions.

The top-level list corresponds to the network object. This possesses two methods (called h_access and add) for accessing network objects and adding further objects onto the list, respectively. Simulation run length is handled by a global object. This stores the simulation and concurrent socket interface 'housekeeping' data, including a counter whose original value specifies the length of simulation. It decrements after each evaluation of the network object, and the simulation halts when the counter reaches zero.

As synapses are also objects, they too have fixed and dynamic parameters much like the neurons. Thus, BaseWeight is the 'normal' or default weight of the synapse and is a constant; Weight holds the present synaptic strength and is variable during simulation; Recovery is a constant (within each synapse) which determines how quickly Weight returns to `BaseWeight`. To prevent synaptic weights growing without limit, `Weight` is bounded during simulation. This models the finite size of stores of neuro-transmitter in the synaptic terminals of real biological neurons. Also, according to Sutton and Barto [71, p. 161]:

> "If it is assumed that synaptic strength slowly decays in the absence of a reinforcement signal, then a bound on weight size is imposed that is a function of reinforcement level and the decay rate. . . . In system theoretic terms, the adaptive element has *definite memory*: it cannot remember anything that occurred arbitrarily far in the past."

Sutton and Barto call this *autonomous decay*. This decay mechanism is also implemented here (via `Recovery`).

3.3 Neuron Types

Hi-NOON allows a non-homogeneous population of neurons to be simulated – reflecting the fact that neurons have specialized functions in real neurobiological systems – at the most appropriate level of abstraction. Modeling individual neurons at the level of membrane potential allows sub-threshold and spiking behaviors to be simulated at low computational cost. The fixed parameters cater for differences between neurons which, in this work, are of the following types:

basic: tells its synapses to fire when its membrane potential crosses threshold from below.

noisy: similar to **basic**, but has an additional internal noise component determining the weighted synaptic input, and hence influencing the membrane potential at the next time step.

ramp: similar to **noisy**, but has ability to ramp up spike generation rate. It is used as a test signal source in network development.

burst: similar to **noisy** but produces a short burst of spikes when its membrane potential crosses threshold.

sensor: similar to **basic**, but acts as a sensory neuron with ARBIB's on-board sensors.

motor: similar to **basic**, but acts as a motor neuron with ARBIB's motor drives.

3.4 State System

Each neuron is treated as being in one of a number (or occasionally more than one) of six states depending on the present membrane potential, cell threshold and whether or not the cell has just fired, etc. For example, if the membrane potential of the basic cell is above threshold, and the cell has not just fired, then the neuron will start to generate a spike and will initiate synaptic transmission.

*** FIGURE 1 ABOUT HERE ***

Figure 1 (taken from a Hi-NOON simulation) shows the states passed through by a neuron during firing. In the case illustrated, the minimum, resting and peak potentials of the neuron are set at −69, −60 and +45 mV respectively, and the threshold value was −45 mV. Note that actual values will under/overshoot these settings before state can change at the next iteration of simulation. The states are:

A: `MP` above resting potential and below threshold

B: above threshold and below peak

C: at peak

D: post-firing

E: at minimum

F: hyperpolarized

The equations governing the membrane potential in each of these states and the synaptic weights are given in Appendix A. The state system is not ‘pure’ – there is some overlap between states. For instance, a neuron may be sub-threshold, but `fired` may still be true, indicating that the membrane potential is undergoing its post-firing hyperpolarization.

The use of a state system for controlling the membrane potential facilitates the addition of new features to the program; it is only necessary to identify which of the states may trigger this feature and to add a procedure call at that particular state. This, coupled with OOP’s inheritance, allows models to be developed and altered relatively easily.

3.5 Axonal and Synaptic Transmission

Our neurons model sub-threshold behaviour but sub-threshold potentials are not propagated (from axon hillock to terminal fibers) in real neurons, only action potentials are. This, however, is not a serious concern because the model’s behaviour depends entirely on how pre-synaptic activity is transformed into post-synaptic activity. It is only in supra-threshold states B, C and D (see Figure 1 and Appendix A.2) that synaptic communication can take place. Hence, it is irrelevant that we are, in some sense, modeling sub-threshold behaviour incorrectly. An alternative view is that we are not modeling axonal transmission, i.e. we have ‘point’ neurons as is common in neural modeling [45, pp. 21-4].

To facilitate the replication of our work by others, code for the C version of the Hi-NOON simulator is made available by anonymous ftp to `mun.ecs.soton.uk` from directory `pub/users/rid/hinoon`.

4 Biologically Motivated Learning

In this section, we detail the biologically-motivated approach to learning implemented in ARBIB, and briefly compare this with other, more behaviorally-based approaches to robot learning.

4.1 Associative and Non-Associative Learning

Learning in ARBIB is modeled on *habituation* and *sensitization*, the most basic forms of non-associative learning, and associative *classical conditioning*. A full and complete review of these extremely-well studied topics would be an enormous undertaking and is well beyond the scope of this paper. Hence, we concentrate here on the basic points which have influenced our model. The reader is referred to [25, 43, 46, 47] for extensive, additional background on associative and non-associative learning.

According to Brooks [14, p. 298]:

> "Two classical types of learning that have been little used in robotics are habituation and sensitization. Both ... seem to be critical in adapting a complex robot to a complex environment. Both are likely to ... be critical for self-adaptation of complex systems."

Habituation occurs when an animal learns to ignore a weak repetitive stimulus whose consequences are neither rewarding nor noxious. In neuronal terms, this means that if neuron A synapses onto neuron B, and A fires repeatedly, the synaptic strength decreases and B's response lessens accordingly. This is modeled in Hi-NOON by lowering the `Weight` of a habituating synapse every time it fires, but `Weight` then returns towards `BaseWeight` with a time constant determined by `Recovery`. The synaptic weight equations for an habituating synapse are given in Appendix A.2.1. Habituation is non-associative in the sense that no other stimulus is involved.

In sensitization, an animal learns to respond more vigorously to a variety of stimuli after it has received a noxious stimulus. Unlike habituation, it occurs in response to a stimulus different from that originally received. Although the term is potentially confusing to those outside the field of psychological learning theory, this is also universally referred to as *non*-associative learning because there is no specific pairing of the noxious and sensitized stimuli: the effect occurs for a variety of sensitized stimuli and for a range of relative timings between them. Sensitization is modeled using a facilitatory interneuron I with synapse-on-synapse connection to an $A \to B$ synapse, as depicted in Figure 2(a). This is essentially the neural model presented by Hawkins and Kandel [35, Fig. 1] for the gill withdrawal reflex in *Aplysia*. When I fires, it increases the strength of the latter synapse. Recovery is as for habituation. The synaptic weight equations for a sensitizing synapse are given in Appendix A.2.2.

*** FIGURE 2 ABOUT HERE ***

Classical conditioning has been intensively studied ever since the pioneering work of Pavlov [55]. It is a form of associative learning in which a pre-existing *unconditioned* reflex (UR) undergoes modification. (According to McFarland and Bösser [49, p. 259], Pavlov used the terms *conditional* and *unconditional* rather that *conditioned* and *unconditioned*: the latter were an error of translation from the original Russian. These authors state that modern practice is to return to Pavlov's original terminology. We prefer, however, to retain the terminology now established in the English language.) If the conditioned stimulus (CS) consistently precedes the unconditioned stimulus (US) by a short time interval, then the CS strengthens the UR and eventually becomes effective in eliciting it in the absence of the US – so called *stimulus substitution* (cf. p. 139 of Sutton and Barto [71] and pp. 168–171 of Lieberman [43]). It is referred to as *associative* learning because the pairing of CS and US is highly specific and embodies a fairly strict timing relative constraint.

How might the mechanisms of these different forms of learning be related? As pointed out by Byrne [19, p. 479]:

> “Classical conditioning and sensitization share some common features. In both, the effectiveness of a test stimulus in eliciting a response is modified by a rein-

> forcing stimulus. They differ, however, in that classical conditioning requires a close temporal relation between the CS and the reinforcer (UR)."

In an intriguing paper, Hawkins and Kandel [35] argue that the cellular mechanism underlying classical conditioning in *Aplysia* is an extension of the mechanism underlying sensitization. They go on to state [p. 375]: "This finding suggests that the mechanisms of yet higher forms of learning may similarly be based on the mechanisms of ... simple forms of learning" referring to this as "an elementary cellular alphabet of learning" [p. 376]. This is a key hypothesis for this paper, in that we attempt to build classical conditioning (and higher-order features such as generalization and second- or higher-order conditioning) on the basic neural model of habituation/sensitization. Hence, our success (or otherwise) has a direct bearing on the plausibility of the notion of a "cell biological alphabet". We emphasize, however, that we make no attempt at this stage to model the full range of conditioning phenomena (e.g. blocking): only the basic aspects are considered.

Accordingly, classical conditioning is modeled in a similar way to sensitization (Figure 2(b)), except for the direct synaptic connection from the interneuron to the US neuron: the strength of the facilitation depends upon the length of time since the last firing of the CS → UR synapse, so that CS and US form a paired association. In our work, the facilitatory effect is greatest when this period is 0.5 s and the CS will not, by itself, strengthen the synapse. Again, recovery is as for habituation. This is effectively the neural model of Lieberman [43, Fig. 13.10] which rests on earlier suggestions of Byrne [19] concerning the cellular mechanisms of learning in *Aplysia*. The synaptic weight equations for a conditioning synapse are given in Appendix A.2.3. Important work on neural models of conditioning has also been described by Grossberg and Schmajuk [32], Schmajuk and DiCarlos [61] and Raymond et al. [56].

4.2 PDP Approaches to Conditioned Learning

As we have seen, the computational neuroscience paradigm attempts to link some of the principles of connectionism with data from experimental neurophysiology. It remains somewhat arbitrary, however, precisely where we draw the line between the two. In particular, several workers [66, 71, 72] have considered modifications to the kind of synaptic updating rules used in connectionist learning to allow the simulation of conditioning behavior. Thus, rather than modeling this behavior at the level of cell parameters within spiking neurons as we do in Hi-NOON, it is modeled at a much higher level within synaptic weight updating equations for PDP type neurons. It is an open question how much practical difference there will be in the outcome between the two approaches. Indeed, it is worth noting Sutton and Barto's [71] observation that the delta rule (of which back-propagation is a generalization) is near identical in its mathematical formulation to Rescorla and Wagner's [57] model of classical conditioning.

To our knowledge, there have been no previous attempts to use classical conditioning in conjunction with spiking neurons in robotic studies. By contrast, the PDP approach outlined in the previous paragraph has been occasionally used as a learning paradigm for robotics. Bühlmeier and Manteuffel [16] and Arkin [4, pp. 322–325] have both recently reviewed this work. According to Bühlmeier and Manteuffel [p. 204] "there are three different directions in the research of conditioning models: attentional models, expectancy or comparator-based models and operant behavior networks" but they also [p. 206] briefly consider classical conditioning. This, they say, "can be interpreted as a mechanism enabling a system to adaptively anticipate protective actions . . . This hard-wired knowledge might be used to adapt the robots behavior". Finally, Andreae [2] has described "associative learning" as a basis for robot intelligence and presented a neural model [p. 21] for this. However, this is done at a relatively high level – essentially within the PDP paradigm.

4.3 Reinforcement Learning

Supervised training is inherently unsatisfactory as a learning technique for autonomous systems. As an alternative to the provision of a teacher, it is generally possible to derive some kind of reinforcement signal from the environment. According to Kaelbling [38, p. 5], this is “a mapping from each state of the environment into a scalar value encoding how desirable that state is for the agent”. Learning then acts to maximize the *reward* or *return* associated with or predicted from the reinforcement signal over time. The many variants of this form of learning (especially the Q-learning paradigm of Watkins [79, 80]) have been very popular in animat studies (e.g. [37, 39, 44, 48, 52, 70]).

Reinforcement learning is similar to the S-R theory of conditioning [43, p. 350] which posits a direct link between the conditioned stimulus and response, in contrast to Pavlovian S-S theory which posits association of two stimuli – the CS and the US. Thus, although we have not done so, a detailed comparison of our Pavlovian approach with reinforcement learning could be potentially revealing from the point of view of psychological learning theory (see discussion in section 8).

5 Robot Nervous System

In this section, we describe the neural system of ARBIB arrived at on the basis of Hi-NOON simulations.

5.1 Central Pattern Generator

Central pattern generators (CPGs) are simple oscillatory neural circuits that underlie rhythmic behaviors in animals such as breathing, walking or swimming [41, 67]. There is no single pacemaker cell, and CPGs can operate autonomously without control from the brain or external feedback. They are found in both invertebrates and vertebrates: in the latter, CPGs are often sited in the spinal cord.

*** FIGURE 3 ABOUT HERE ***

Figure 3 shows the simple hard-wired CPG which provides the basic driving force for ARBIB, in the form of a wandering behavior. Although not based on any specific animal, this CPG uses principles from identified neural circuitry, e.g. the *Clione* swim network [60]. It consists of a four-neuron loop, C1-C2-C3-C4, with mutual inhibition between two of the neurons, C1 and C3, so that there is a cyclical or oscillatory pattern of firing. There is an excitatory connection from neuron C1 to the forward motor, and from C2 and C4 to the right and left steer motor neurons respectively. The synapses to the steer neurons are of type **noisy**, which means that the probability of the synapse firing depends on the activity of the parent, pre-synaptic neuron, thus allowing for deviation from the robot's generally straight-line path when wandering over long distances. The CPG does not start firing spontaneously; an excitatory connection from the rear bumper sensory neuron (Fig. 3) activates C1, thus starting the oscillatory pattern of firing of the CPG. This is a practical expedient that allows ARBIB to be 'kick-started' by pressing its back bumper.

5.2 Reflex Responses

Classical conditioning assumes that there are genetically-provided (or 'pre-wired') behaviours, or reflexes, which have evolved to help the species' survival in its environment These are then modified by an individual's experience of the environment. The obvious question then arises: How do we decide what should be hard-wired in ARBIB? Clearly, this shapes what the robot can and cannot learn. In their seminal 1984 paper, Hawkins and Kandel write [35, footnote 1]:

> "… we believe that the neural connections for most or all possible stimulus-response associations are prewired, and training merely alters the strengths of

> those connections, in some cases bringing the response from below threshold to above threshold."

This is exactly the point of view adopted here, and gives a way to interpret *emergence*. An emergent behavior is one which has been brought above threshold by interaction with the environment.

ARBIB has two (Left/Right) hard-wired reflex responses as depicted in Figure 4. These are used as the unconditioned stimuli for classical conditioning and form the starting point for learning in the robot. Each of the two front bump sensors makes an excitatory connection to the ipsilateral reflex interneuron. These reflex neurons in turn connect to the reverse motor neuron and the ipsilateral steer motor. If, for example, the left bump sensor is activated, then ARBIB will reverse away from the object that it collided with while turning at the same time. Hence, once the reflex has been completed, ARBIB will no longer be facing toward the object with which it collided. This has an obvious analogy with the 'withdraw-from-pain' reflex that all animals possess.

*** FIGURE 4 ABOUT HERE ***

There is one final reflex that plays no part in the adaptive behavior. An inhibitory connection from the rear bumper sensory neuron to the reverse motor neuron means that the vehicle will stop reversing if it backs into something, allowing it to wander forward again under the control of the CPG.

5.3 Classical Conditioning

Figure 5 illustrates the addition of conditioning circuitry (Figure 2(b)) to the basic reflex circuitry. Interneurons feature both because monosynaptic reflexes are uncommon in biology [35, pp. 376–7] and because this gives a natural, structured way of introducing higher-order conditioning (see below). Note that the circuit is 'over-provided' with conditioning synapses in accordance with the quotation from Hawkins and Kandel above, i.e. ARBIB has the potential to learn ‘wrong’ things but should be prevented from doing so by the environment.

*** FIGURE 5 ABOUT HERE ***

We now outline these operational principles in a little detail. Each bump sensory neuron makes conditioning connections to the synapses connecting (from the infrared sensory neurons) to the ipsilateral infrared interneuron (*IIL* and *IIR*, respectively). Each infrared sensory neuron (*IR*) makes connections to both of the infrared interneurons (see Figure 5): thus, the network is initially wired up neutrally. (That is, to avoid explicitly pre-determining its behavior, ARBIB is over-endowed with the potential to learn.) The connections between the *IR* sensory and interneurons are very weak initially; even if both sensory neurons fire, the *IR* interneurons will not. This means that, to begin with, ARBIB essentially ignores the signals from the *IR* sensory neurons – they have no effect on its behavior. However, as it explores its world, the connections from the *IR* sensory neurons become strengthened, i.e. it increasingly uses the information from its *IR* neurons to avoid hitting obstacles. In fact, IR signals lead to reflex behavior just as if ARBIB had struck an obstacle. How does this happen?

Imagine that ARBIB is wandering around when it approaches an obstacle on its left hand side. As it draws close to the obstacle, the left *IR* sensory neuron begins to fire (because an IR sensor is active when it picks up ARBIB’s active IR signals reflected back off nearby objects). However, although the *IR* sensory neuron sends signals to both *IR* interneurons (*IIL* and *IIR*), the connection is too weak for the interneurons to fire, so that ARBIB continues forward. A second or two later, the left bump sensor hits the object. The immediate effect is that the left reflex interneuron is triggered and ARBIB begins its reflexive withdrawal behavior. The other effect is the *possible* conditioning of the connections from the two *IR* sensory neurons to *IIL*. However, only the left *IR* sensory neuron has fired recently, so only the connection from that neuron to *IIL* will be facilitated – its weight will be increased. In future (perhaps after just one or two more 'trials' depending on the exact parameters used), the connection from the left *IR* sensory neuron and the *IIL* is strong enough that the IR signal on its own is enough to trigger the left reflex interneuron and cause the backing-off behavior. ARBIB has learned to use a different sensory modality to avoid the 'pain' of colliding with obstacles – as it approaches an object, its *IR* neuron fires and it reverses away.

Once these conditioned connections have been formed, the next stage of adaptation takes place.

5.4 Chained ('Second-Order') Conditioning

The task of learning to avoid obstacles through the association of objects with shadows occurs in a similar manner to the association of bump and IR detection previously described. However, the light detecting resistors (LDRs) used in this case generate a graded signal (between 0 and 255) rather than the binary 0 or 1 output of the bump sensors and IR sensors. Also, we wished to avoid explicit pre-programming of light-seeking ability. Accordingly, ARBIB was provided with a means of sensing changes in light rather than detecting absolute levels of light *per se*.

ARBIB is equipped with light sensing circuitry inspired from the visual ability of the scallop *Pecten maximus* [73]. Light, shadow and movement detection in this animal is provided through a compound of approximately 60 eyes. Each consists of a cornea, a large lens, sense cells arranged as proximal and distal retinae, a reflecting argentae and a layer of screening pigment cells. For our purposes, this is simplified to two 'eyes' and four sensory (or Δ) cells: two proximal and two distal. Proximal cells respond to an increase in light intensity by generating APs until a constant illumination has been detected, at which point spike generation decays. Discharge ceases with a decrease in light intensity. By contrast, distal cells respond to a decrease in light intensity, showing no activity during constant illumination or an increase in light intensity [73, p. 244–5].

In ARBIB, two LDRs give a simple view of the left and right half-fields of the scene ahead. These feed to the (proximal) $\Delta{+}L$ and $\Delta{+}R$ cells and the (distal) $\Delta{-}L$ and $\Delta{-}R$ cells as shown in Figure 6. Light intensity is sampled and compared with the previous value. If the difference is positive and the cell is proximal, this difference is added to the membrane potential. If the difference is negative and the cell is distal, the difference is again added to the membrane potential. Here, thresholds are very close to the resting potential of the sensory cells, making them very sensitive to changes in light.

As depicted in Figure 6, the four Δ neurons make weak connections to the set of light interneurons ($LI-$, $LI+$, $RI+$ and $RI-$) so that each Δ neuron connects to the ipsi- and contralateral neuron with the same sign. These connections can be conditioned by the synapse-on-synapse connections from the infrared interneuron(s), but initially are not strong enough to cause the light interneurons (and therefore the reflex) to fire.

*** FIGURE 6 ABOUT HERE ***

As for the learning, suppose ARBIB enters a patch of shadow behind an obstacle to its left. This has no immediate effect on the behavior, but does cause the $\Delta{-}L$ neuron to fire. A moment later, the left IR sensor picks up the obstacle and fires the *L.IR* sensory neuron, causing *IIL* to fire. This not only triggers the learned reflex behavior, but also strengthens the connection from a Δ neuron to the left light interneurons if that Δ neuron has recently fired. In this case, $\Delta{-}L$ has recently fired so that connection is strengthened. It typically takes two to three incidents such as this before ARBIB starts to back away from shadows even without either *IR* sensory neuron firing.

Notice that the initial wiring of the vehicle does not make assumptions about the world (except for the fact that touch is a shorter-range modality than IR which, in turn, is shorter-range than light). No attempt was made to implement any kind of range sensing algorithm:

ARBIB was simply given the potential to sense range. The theoretical/empirical issue is how it learns to use these potentialities, given its (limited) neural endowments. For example, if we were to put reversing glasses on ARBIB or if obstacles were bright rather than dark, the neural circuit would still adapt correctly to its environment. Thus, its behavior can truly be described as emergent intelligence, albeit at a relatively low level (of ‘animal’ as opposed to ‘human’ intelligence).

6 Implementation of the Autonomous Robot

Various implementations of ARBIB have been constructed operating on the above principles.

6.1 Z180 Implementation

Figure 7 shows this implementation, which has two independent driving wheels, and a rear trailing wheel. It was constructed at low cost making extensive use of sheet metal for the bodywork.

*** FIGURE 7 ABOUT HERE ***

The electronic hardware is divided into analog and digital subsystems, realized on separate printed circuit boards. The collection and refinement of sensory information is done by the analog subsystem, while the digital hardware is responsible for executing the neural network code, monitoring sensory information from the analog-to-digital converter, and controlling the motors driving the wheels. The digital hardware is based on a Zilog Z180 microprocessor operating at 6.144 MHz, with 1 Mbyte addressing capability, built-in support for direct memory access, and various other refinements which allow a highly compact design. Technical information on the Zilog Z180 microprocessor is available on the WorldWide Web from URL:

http://www.zilog.com/frames/fz80_180micro.html

The system has 8 Kbyte ROM containing bootstrapping code and 128 Kbyte RAM containing the neural network code. Three programmable logic devices provide address decoding for the memory systems, control of the input/output decoding and pulse-width modulation signals to control the motor speeds and directions, respectively. The IR sensors are L14G1 phototransistors and the LDRs are NORP-12's from EG & G Vactec. The latter are cadmium sulphide LDRs with a peak response at 530 nm and a dark resistance of 1 MΩ, falling to 400 Ω at 1000 lux.

After simulation to verify correct operation, the neural software was rewritten in a Small C subset of the C programming language, implementing most of the C language constructs but restricting data types to integers and single-dimension arrays. The one used here was designed for use with Zilog Z80 systems and works via a Z80 cross-assembler and linker. The monitor program in the robot's ROM was designed to recognize the output files from the linker. The software was developed incrementally: first the CPG was coded; then the bump-sensor reflex was added; finally, the IR and LDR sensors and *IR* and *LDR* sensory neurons and conditioning synapses were added. (Note the parallel to Brook's layered, subsumption architecture [11].) The developing system was extensively tested at each stage.

On reset, the ROM code initializes the system and sets up a serial link to a PC. This is the only physical link to/from ARBIB: we intend to dispense with this in the near future. Neural network code is then downloaded to RAM from the PC, and the robot starts to interact with – and learn from – its environment.

6.2 ARM Implementation

The original implementation based on the 6.144 MHz Z180 processor appeared to make many 'mistakes' which we interpreted to be failures to learn *in time* for the learning to be useful, i.e. before a collision. This was thought to be a consequence of the slow clock rate and limited processing ability of the Z180. Hence, a subsequent implementation used a much faster 40 MHz 32-bit ARM710 processor from Advanced RISC Machines Limited. The ARM710 has an 8 Kbyte cache with write buffer and memory management unit on-chip. Fuller details can be found at URL:

`http://www.armltd.co.uk/Documentation/UserMans/#ARM7TDMI`

Software for the ARM implementation was written in C using the ARM Software Development Toolkit. The bodywork has been rebuilt using Lego.

In practice, the ARM implementation reacts *too* fast – in that the processor emulates ARBIB's neural circuitry considerably faster than the environment changes. Thus, after hitting an obstacle, the robot would back off just enough to stop the bump sensor from making contact, but no more. This is a consequence of the absence of a global clock or other device for imposing realistic timing on the simulation: the neural system merely runs as fast as the processor allows. The problem was cured by adding delay loops to the program code so as to match its speed of operation to the time scale of significant events in its environment.

This is an interesting observation. The problem of matching the timescales of internal computation to those of external events has recently been comprehensively discussed by Ballard et al. [6]. A central tenet of their work is that "intelligence has to relate to interactions with the physical world, meaning that the particular form of the ... body is a vital constraint in delimiting many aspects of intelligent behavior". They propose that "pointing movements are used to bind objects in the world to cognitive programs" – so-called "deitic computation" – as a solution to this problem. Deitic computation reflects a particular level of hierarchical organization (and therefore of abstraction) and so operates on a characteristic timescale – about 1/3 second for the human sensory-motor and cognitive systems. According to this notion, the 'slow' speed of neurons viewed as computing elements becomes a virtue/necessity – contributing to the matching of internal and external timescales – rather than an engineering disadvantage or 'paradox' [31].

6.3 Khepera Implementation

ARBIB has also been implemented on a Khepera miniature robot. Technical details of this product are available at:

`http://lamiwww.epfl.ch/lami/robots/K-family/Khepera.html`

and at:

`http://lamiwww.epfl.ch/Khepera/#khepera`

It has a cylindrical shape, 55 mm in diameter and 30 mm in height. A serial link allows master-slave operation of the unit from a workstation, or alternatively, application programs can be uploaded to Khepera for execution on its onboard MC68331 processor. Eight infrared proximity devices (Siemens SFH900s) are located at its perimeter, sensing distance as well as light level. Khepera is equipped with two motors providing mobility through its environment, each motor giving feedback to the onboard processor through an incremental encoder mounted on its drive shaft.

7 Results

Here, we report on experiments designed to monitor changes in ARBIB's behaviour as it adapts and learns from its environment. This presents something of a difficulty because the various implementations have insufficient on-board data-logging capability to do this totally autonomously. However, the Khepera simulator [51] gives a way of capturing and processing real-time data from the Khepera implementation of ARBIB via an umbilical link. In this case, it also makes sense to simulate ARBIB's nervous system remotely (but in real-time) on the PC running the simulator. We emphasize that the data being collected are in no way 'simulated': they come from a real implementation of ARBIB. The Khepera simulator is merely used as an expedient means of data logging.

Evidence of ARBIB's ability to adapt to an environment was acquired over an extended period, as described below. Adaptation was monitored by recording neural and synaptic activity at key points within Arbib's nervous system during its exploration of the test environment. ARBIB was implemented on a Khepera robot in mode 3 (controlled by serial link protocol at 38400 Baud). An 80486 Linux PC ran the Khepera simulator (version 2), augmented with Hi-NOON capability. Physically, the Khepera robot does not possess bump sensors. Hence, the bump sensory neurons are actually fed with infrared signals at positions 6 and 7 of Khepera (see Figure 3 of [51]) as a convenient way of detecting objects at short range.

Because ARBIB's behavior is probabilistic, results are averaged over several 'runs' . Also, it must be given sufficient exposure to the environment for learning to be effective. Thus, each 'run' consists of several 'mini-trials', starting from approximately the same start point. Weights are reset to baseline values only at the beginning of a new run: they are not reset at the beginning of each mini-trial. In this way, extended periods of learning can be studied even with a relatively compact test environment.

ARBIB was initially positioned at the start location, chosen such that IR sensors at positions 6 and 7 (see above) are active (and capable of kick-starting the CPG). Directly ahead, at a distance of 28 cm (measured center to center), a cylindrical object of 8 cm in diameter and 9 cm in height was placed. (A red mug was used for this purpose with the handle facing away from the robot.) Viewing the scene from the start location, a light source (a bench power supply unit feeding a 4.8 V, 0.5 A MB3 filament bulb) was positioned 8 cm to the left, suspended 16.5 cm above and 9 cm behind the mug. The robot laboratory (which is windowless) was darkened as much as possible so that the mug threw a distinct shadow (which did not encompass the start position).

Each mini-trial commenced by placing ARBIB at the start location (as described above) and invoking the simulator. The robot was allowed to move towards the mug, reacting to the obstacle (possibly colliding) until finally it left the area of its shadow, away from any potential collision. The mini-trial was then stopped and ARBIB replaced at the start location – ready for the next mini-trial. After six such mini-trials, constituting a complete run, the simulator was terminated and data logged for subsequent analysis. Six runs were completed, accumulating data for 36 mini-trials, lasting a total of approximately 1.7 hours and yielding 54584 KB of data.

To avoid the dangers inherent in presenting results from single runs, action potentials and synaptic weights have been averaged over all six runs. Because of the probabilistic behavior, the different stages of learning and the fact that the start point is only approximately the same, the time scales over each mini-trial (and, therefore, over each run) actually differ. Unfortunately, there is no obviously best way to present the results of learning. Should we take averages over real time or over some normalized time scale? In what follows, we have decided to present results in terms of real time. This means that data are truncated to the shortest time scale of the six runs (approximately 850 s or just over 14 minutes). This shortest time scale was then divided into bins of 50 s and APs were counted in each bin before ensemble averaging the bin values across the six runs.

*** FIGURE 8 ABOUT HERE ***

Figure 8 shows the resulting plots of average AP count for the left and right bump sensory neurons. In interpreting this (and subsequent) plots, the reader should recall that each run is a composite of six mini-trials. Accordingly we would expect firings to cluster into six groups, more or less equally spaced across the time extent of the plot, because collisions can only take place towards the end of a mini-trial. As can be seen, however, activity decays with time in both cases – left bump and right bump. This is evidence of sensory substitution through classical conditioning. That is, ARBIB learns to rely upon a different sense modality for proximity detection than the one involved in its hard-wired reflexes (i.e. bumping into an object).

As shown in Figure 9, right and left infrared sensory neuron activity also decays, albeit at a lower rate. Again, this is evidence of ARBIB learning through chained conditioning. In time, it starts to use a different sense modality than that previously learned – the infrared sense is being substituted by the light sense (using the delta light sensory neurons) as a source of collision detection.

*** FIGURE 9 ABOUT HERE ***

The activities of the infrared interneurons, depicted in Figure 10, show signs of increasing before reducing. This results from strengthening of synaptic weights between the infrared sensory and infrared interneurons before the establishment of chained conditioning. After this is established, however, the activities reduce as light sensing takes over from the infrared sense in evoking the withdrawal reflex.

*** FIGURE 10 ABOUT HERE ***

Consider next how synaptic weights change during this experiment. It was observed that two classes of behavior emerged: when avoiding the mug, ARBIB either consistently passed it on its right (runs 2, 4, 5 and 6) or consistently passed it on its left (runs 1 and 3). Because the pattern of learning is different in the two cases, results are presented separately for the two behaviors. Figure 11 shows weights developing between infrared sensory and interneurons for the first case (runs 2, 4, 5 and 6). Clearly, the left infrared ipsilateral and right infrared contralateral synapses have been considerably strengthened by conditioning, whereas the left infrared contralateral and right ipsilateral synapses have not. However, in the second case (runs 1 and 3) in which ARBIB learns to pass the mug on its left, the situation is reversed (Fig. 12). That is, the right infrared ipsilateral and left infrared contralateral synapses have been strengthened, but the right infrared contralateral and left ipsilateral synapses have not.

*** FIGURE 11 ABOUT HERE ***

*** FIGURE 12 ABOUT HERE ***

Activity in the delta light interneurons is depicted in Figure 13. Firings are clearly biased to later in the run, indicating the strengthening of their pre-synaptic weights only *after* the establishment of the infrared interneurons as collision detectors as a result of chained conditioning. Finally, Figure 14 shows the average activity of the reflex interneurons over the six runs. Both left and right interneurons show signs of reducing activity with time. This indicates that ARBIB has acquired information that allows it to reduce the frequency of collisions (i.e. 'pain') that it experiences while exploring its environment. So, although ARBIB has no explicit *value system* hard-coded into its nervous system (as in the case of [76] for instance), it has still established a behaviour that reduces the chances of an aversive stimulus occurring.

*** FIGURE 13 ABOUT HERE ***

*** FIGURE 14 ABOUT HERE ***

The various implementations were also observed informally as they learned from their environment. Observed behavior was entirely consistent with the more quantitative description above.

8 Conclusions and Discussion

We have detailed the evolution and embodiment of a simple robot creature, ARBIB, situated in the real world. In order to exploit the specific advantages of each approach, we aim to model ARBIB's nervous system at a level of abstraction intermediate between the PDP style and the physiologically faithful modeling of real neural systems, in accordance with the tenets of computational neuroscience. This imposes a necessity to decide where the dividing line between 'neuroscience' and 'PDP modeling' should lie. In this work, we attempt to maintain biological fidelity by modeling at the level of neuron membrane potential. Thus, ARBIB's nervous system was initially designed and debugged using the authors' Hi-NOON neural simulator in which synapses, neurons and networks are all represented as objects within an object-oriented hierarchy. This allows a non-homogeneous population of neurons to be implemented, reflecting the fact that neurons have specialized functions in real neurobiological systems.

Various physical instantiations of ARBIB have been constructed and observed to adapt to the real world using simple forms of non-associative and associative learning – exhibiting shadow- and collision-avoidance behavior. This behavior is both ecologically valid and emergent: it was not programmed into the simulation but results from learned modification to pre-existing, hard-wired reflexes. Admittedly, there is a sense in which behavior is "pre-wired" but this is not in any deliberate way. That is, the robot's nervous system is *over-provided* with conditioning synapses, so that it has the potential to learn all sorts of things. But, of course, it does not learn just anything. Its learning is shaped by the environment.

The work has similarities to other situated-systems studies within the 'new AI' paradigm, which emphasizes the importance of embodiment and learning from the environment, but has deliberately greater biological fidelity. Perhaps the closest study to ours is that of Verschure, Kröse and Pfeifer [75]. These authors also describe how classical conditioning (based on the "adaptive fields" of Verschure and Coolen [74]) can modify pre-existing reflexes to produce collision-avoidance and goal-seeking behaviors in autonomous robots.

There are, however, several important differences. First, Verschure et al. merely simulate an autonomous robot rather than building one. Second, they model the robot's nervous system at a significantly higher level than we do. Rather than using non-homogeneous spiking neurons, the so-called CS, US and UR fields contain homogeneous linear threshold units with binary activations. Conditioning is modeled by the effects of the CS field on the US^+ and US^- fields as opposed to our synapse-on-synapse mechanism, and [p. 185] "connections between the US-fields and UR-fields are prewired and are not modifiable". Third, there is nothing as low level (from the biological perspective) as our central pattern generator to provide the motivational force for exploring the environment. In place of our CPG are "command neurons" and "specific motor responses". Finally, Verschure et al. seem to have a different view of reinforcement learning to ours. They state [p. 185] that their classical conditioning approach "would fall under the paradigm of reinforcement learning" whereas we have preferred to emphasize the distinctions (section 4.3). In short, Verschure et al.'s approach is behaviorally-based while ours is biologically-based. As pointed out by Donegon, Gluck and Thompson [26], there seems much to be gained from attempting to integrate these two approaches. Verschure et al.'s work has since evolved considerably [76], principally by the construction of "a behaving real world artifact" which confirms and extends their ideas. However, the approach remains behaviorally-based.

In the realm of psychological learning theory, controversy has raged for decades on the precise relation between classical conditioning and reinforcement learning. For instance, Lieberman [43, p. 355] says:

> "Reinforcement and classical conditioning are extraordinarily similar in their basic principles, which suggests that they involve the same learning mechanism, but neither form of learning is able to account for the other: R-S* and S-S* contingencies have different effects on behavior, and the effects of one cannot be reduced to the other. Another way to account for their similarity is to assume that both rely on a common system for detecting relationships between events."

and

> "Any analysis that claims that reinforcement and classical conditioning are based on the same learning process needs to be able to explain why classical conditioning influences autonomic responses more that skeletal responses whereas the

> reverse is true for reinforcement . . . ”

There seems every prospect that robotic work of the sort described in this paper could contribute to the resolution of this debate, by implementing each in the same artifact and comparing their behavior under a range of experimental conditions.

A key question for future work is the extent to which the approach described here – which leads to simple ‘animal-level’ behaviors – can scale and generalize in the direction of more complex ‘human-level’ intelligence. The issue has been discussed by Wilson [82] and, more recently, by Brooks [14]. Wilson advocates [p 20] “. . . maintaining the holism of the situation of real animals in real environments, while progressively . . . increasing animat complexity only as necessary” in order to cope with increasingly complex environments. This is one direction we intend to follow. In this regard, we should not lose sight of the extreme simplicity of ARBIB in its current instantiation. Given this, it is remarkable that ARBIB’s behavior is as complex as it is. There is clearly some way to go in following Wilson’s “animat path to AI”.

Acknowledgements

Preliminary versions of this paper were presented at the 1997 IEEE International Conference on Systems, Man and Cybernetics, Orlando, FL, at the 1998 IEEE International Symposium on Circuits and Systems, Monterey, CA, and in seminars by author RID to the Department of Cybernetics, University of Reading, UK and the Institute for Human and Machine Cognition, Department of Computer Science, University of West Florida, Pensacola, FL. Thanks to questioners and discussants whose perceptive comments on those occasions prompted us to improve the presentation and content of this manuscript.

References

[1] T. L. Anderson and M. Donath. Animal behavior as a paradigm for developing robot autonomy. *Robotics and Autonomous Systems*, 6:145–168, 1990.

[2] J. H. Andreae. *Associative Learning for a Robotic Intelligence*. Imperial College Press, London, UK, 1998.

[3] M. A. Arbib. *Brains, Machines and Mathematics (2nd edition)*. Springer-Verlag, Berlin, 1987.

[4] R. C. Arkin. *Behavior-Based Robotics*. MIT Press, Cambridge, MA, 1998.

[5] N. R. Ball and K. Warwick. Using self-organizing feature maps for the control of artificial organisms. *IEE Proceedings, Part D: Control Theory and Applications*, 140:176–180, 1993.

[6] D. H. Ballard, M. M. Hayhoe, P. K. Pook, and R. P. N. Rao. Deitic codes for the embodiment of cognition. *Behavioral and Brain Sciences*, 20:723–767, 1997. (Including open peer commentary).

[7] H. B. Barlow. Single units and sensation: A neuron doctrine for perceptual psychology? *Perception*, 1:377–394, 1972.

[8] G. A. Bekey. Biologically inspired control of autonomous robots. *Robotics and Autonomous Systems*, 18:21–31, 1996.

[9] A. Billard and K. Dautenhahn. Grounding communication in autonomous robots: An experimental study. *Robotics and Autonomous Systems*, 24(1–2):71–79, 1998.

[10] V. Braitenberg. *Vehicles: Experiments in Synthetic Psychology*. MIT Press, Cambridge, MA, 1984.

[11] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 1:14–23, 1986.

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

[13] R. A. Brooks. Human level cognition in embodied robots. In *Proceedings of International Joint Conference on Neural Networks*, volume 2, pages 1079–1084, Nagoya, Japan, 1993.

[14] R. A. Brooks. From earwigs to humans. *Robotics and Autonomous Systems*, 20:291–304, 1997.

[15] R. A. Brooks and L. A. Stein. Building brains for bodies. *Autonomous Robots*, 1:7–25, 1994.

[16] A. Bühlmeier and G. Manteuffel. Operant conditioning in robots. In O. Omidvar and P. van der Smagt, editors, *Neural Systems for Robotics*, pages 195–225. Academic, San Diego, CA, 1997.

[17] D. Bullock and S. Grossberg. Neural dynamics of planned arm movements: emergent invariants and speed-accuracy properties during trajectory formation. *Psychological Review*, 95(1):49–90, 1988.

[18] J. H. Byrne. Comparative aspects of neural circuits for inking behavior and gill withdrawal in *Aplysia californica*. *Journal of Neurophysiology*, 45:98–106, 1980.

[19] J. H. Byrne. Neural and molecular mechanisms underlying information storage in *Aplysia*: implications for learning and memory. *Trends in Neurosciences*, 8:478–482, 1985.

[20] H. C. Card and W. R. Moore. Silicon models of associative learning in *Aplysia*. *Neural Networks*, 3:333–346, 1990.

[21] Y. Chauvin and D. Rumelhart, editors. *Backpropagation: Theories, Architectures and Applications*. Lawrence Erlbaum, Hillsdale, NJ, 1995.

[22] A. Clark. Being there: Why implementation matters to cognitive science. *Artificial Intelligence Review*, 1:231–244, 1987.

[23] P. Coad and E. Yourdon. *Object Oriented Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1991. Second Edition.

[24] F. Crick. The recent excitement about neural networks. *Nature*, 337:129–132, 1989.

[25] A. Dickinson. *Contemporary Animal Learning Theory*. Cambridge University Press, Cambridge, UK, 1980.

[26] N. H. Donegon, M. A. Gluck, and R. F. Thompson. Integrating biological and behavioral models of classical conditioning. In R. D. Hawkins and G. H. Bower, editors, *Computational Models of Learning in Simple Neural Systems*, pages 109–156. Academic, San Diego, CA, 1989.

[27] A. Eliëns. *Principles of Object-Oriented Software Development*. Addison-Wesley, Wokingham, UK, 1994.

[28] P. Gaudiano and S. Grossberg. Vector associative maps – unsupervised real-time error-based learning and control of movement trajectories. *Neural Networks*, 4(2):147–183, 1991.

[29] P. Gaudiano, F. H. Guenter, and E. Zalama. The neural dynamics approach to sensory-motor control: Overview and recent applications in mobile robot control and speech

production. In O. Omidvar and P. van der Smagt, editors, *Neural Systems for Robotics*, pages 153–194. Academic, San Diego, CA, 1997.

[30] P. Gaudiano, E. Zalama, and J. L. Coronado. An unsupervised neural network for low-level control of a wheeled robot: Noise resistance, stability and hardware implementation. *IEEE Transactions on Systems, Man and Cybernetics*, 26(3):485–496, 1996.

[31] W. Gerstner, R. Kempter, J. L. van Hemmen, and H. Wagner. A neuronal learning rule for sub-millisecond temporal coding. *Nature*, 383:76–81, 1996.

[32] S. Grossberg and N. A. Schmajuk. Neural dynamics of attentionally modulated Pavlovian conditioning: Conditioned reinforcement, inhibition, and opponent processing. *Psychobiology*, 15:195–240, 1987.

[33] S. Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.

[34] R. D. Hawkins. A biologically based computational model for several simple forms of learning. In R. D. Hawkins and G. H. Bower, editors, *Computational Models of Learning in Simple Neural Systems*, pages 65–108. Academic, San Diego, CA, 1989.

[35] R. D. Hawkins and E. R. Kandel. Is there a cell biological alphabet for simple forms of learning? *Psychological Review*, 91:375–391, 1984.

[36] J. N. H. Heemskerk and N. E. Sharkey. Learning subsumptions for an autonomous robot. In *IEE Colloquium on Self Learning Robots*, number 026 in Colloquium Digest Series 96, pages 8/1–8/3, London, 1996. Institution of Electrical Engineers.

[37] L. P. Kaelbling. Foundations of learning in autonomous systems. *Robotics and Autonomous Systems*, 8:131–144, 1991.

[38] L. P. Kaelbling. *Learning in Embedded Systems*. MIT Press, Cambridge, MA, 1993.

[39] M. Kaiser and R. Dillman. Hierarchical refinement of skills and skills acquisition for autonomous robots. *Robotics and Autonomous Systems*, 19:259–271, 1997.

[40] E. R. Kandel. Small systems of neurons. *Scientific American*, 241:61–70, 1979.

[41] D. Kleinfield and H. Sompolinsky. Associative neural networks for central pattern generators. In C. Koch and I. Segev, editors, *Methods in Neuronal Modeling: From Synapses to Networks*, pages 195–246. MIT Press, Cambridge, MA, 1989.

[42] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

[43] D. A. Lieberman. *Learning: Behavior and Cognition (2nd edition)*. Brooks/Cole, Pacific Grove, CA, 1993.

[44] L.-J. Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8:293–321, 1992.

[45] R. J. MacGregor. *Neural and Brain Modeling*. Academic, London, UK, 1987.

[46] N. J. Mackintosh. *The Psychology of Animal Learning*. Academic Press, New York, NY, 1974.

[47] N. J. Mackintosh. *Conditioning and Associative Learning*. Oxford University Press, Oxford, UK, 1983.

[48] M. J. Mataric. Reinforcement learning in the multi-robot domain. *Autonomous Robots*, 4:73–83, 1997.

[49] D. McFarland and T. Bösser. *Intelligent Behavior in Animals and Robots*. Bradford Books/MIT Press, Cambridge, MA, 1993.

[50] C. Miall. The diversity of neuronal properties. In R. Durbin, C. Miall, and G. Mitchison, editors, *The Computing Neuron*, pages 11–34. Addison-Wesley, Redwood City, CA, 1989.

[51] O. Michel. Khepera Simulator version 2, User Manual. Downloadable from URL `http://diwww.epfl.ch/lami/team/michel/khep-sim/index.html`, March 1, 1996.

[52] J. D. Millan. Reinforcement learning of goal-directed obstacle-avoiding reaction strategies in an autonomous mobile robot. *Robotics and Autonomous Systems*, 15:237–246, 1995.

[53] U. Nehmzow and B. McGonigle. Achieving rapid adaptations in robots by means of external tuition. In D. T. Cliff, P. Husbands, J.-A. Meyer, and S. W. Wilson, editors, *From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pages 301–308. Bradford Books/MIT Press, Cambridge, MA, 1994.

[54] U. Nehmzow, T. Smithers, and J. Hallam. Location recognition in a mobile robot using self-organising feature maps. In G. Schmidt, editor, *Information Processing in Autonomous Mobile Robots*, pages 267–277. Springer-Verlag, Berlin, 1991.

[55] I. P. Pavlov. *Conditioned Reflexes: An Investigation of the Activity of the Cerebral Cortex*. Oxford University Press, London, UK, 1927. (G.V. Anrep, Translator).

[56] J. L. Raymond, D. A. Baxter, D. V. Buonomano, and J. H. Byrne. A learning based on empirically-derived activity-dependent neuromodulation supports operant conditioning in a small neural network. *Neural Networks*, 5:789–803, 1992.

[57] R. A. Rescorla and A. R. Wagner. A theory of Pavlovian conditioning: The effectiveness of reinforcement and non-reinforcement. In A. H. Black and W. F. Prokasy, edi-

tors, *Classical Conditioning II: Current Research and Theory*, pages 64–69. Appleton-Century-Crofts, New York, NY, 1972.

[58] D. E. Rumelhart and J. L. McClelland (Eds.). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition (2 Volumes)*. Bradford Books/MIT Press, Cambridge, MA, 1986.

[59] D. E. Rumelhart, G. E. Hinton, and R. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[60] R. A. Satterlie. Reciprocal inhibition and postinhibitory rebound produce reverberation in a locomotor pattern generator. *Science*, 229:402–404, 1985.

[61] N. A. Schmajuk and J. J. DiCarlo. Stimulus configuration, classical conditioning, and hippocampal function. *Psychological Review*, 99(2):268–305, 1992.

[62] T. W. Scutt. *Synthetic Neural Networks: A Situated Systems Approach*. PhD thesis, Department of Electronics and Computer Science, University of Southampton, UK, 1995.

[63] T. W. Scutt and R. I. Damper. Computational modelling of learning and behaviour in small neuronal systems. In *Proceedings of International Joint Conference on Neural Networks*, pages 430–435, Singapore, 1991.

[64] T. W. Scutt and R. I. Damper. Designing a nervous system for an adaptive mobile robot. In A. Browne, editor, *Neural Network Perspectives on Cognition and Adaptive Robotics*, pages 220–250. Institute of Physics Press, Bristol, UK, 1997.

[65] T. J. Sejnowski, C. Koch, and P. S. Churchland. Computational neuroscience. *Science*, 241:1299–1306, 1988.

[66] T. J. Sejnowski and G. Tessauro. The Hebb rule for synaptic plasticity: Implementations and applications. In J. H. Byrne and W. O. Berry, editors, *Neural Models of Plasticity*, pages 94–103. Academic Press, San Diego, CA, 1989.

[67] A. I. Selverston. A consideration of invertebrate pattern generators as computational databases. *Neural Networks*, 1:109–117, 1988.

[68] N. E. Sharkey and J. N. H. Heemskerk. The neural mind and the robot. In A. Browne, editor, *Neural Network Perspectives on Cognition and Adaptive Robotics*, pages 169–194. Institute of Physics Press, Bristol, UK, 1997.

[69] L. Steels. Towards a theory of emergent functionality. In J.-A. Meyer and S. W. Wilson, editors, *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, pages 451–461, Cambridge, MA, 1991. Bradford Books/MIT Press.

[70] R. S. Sutton. Reinforcement learning architectures for animats. In J.-A. Meyer and S. W. Wilson, editors, *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, pages 288–296. Bradford Books/MIT Press, Cambridge, MA, 1991.

[71] R. S. Sutton and A. G. Barto. Towards a modern theory of adaptive networks: Expectation and prediction. *Psychological Review*, 88:135–170, 1981.

[72] G. Tessauro. Simple neural models of classical conditioning. *Biological Cybernetics*, 55:187–200, 1986.

[73] U. Thurm. Eyes specialized for dark responses. In W. Reichardt, editor, *Processing of Optical Data by Organisms and Machines*, pages 236–255. Academic, New York, NY, 1969.

[74] P. F. M. J. Verschure and A. C. C. Coolen. Adaptive fields – distributed representations of classically-conditioned associations. *Network – Computation in Neural Systems*, 2:189–206, 1991.

[75] P. F. M. J. Verschure, B. J. A. Kröse, and R. Pfeifer. Distributed adaptive control: The self-organization of structured behavior. *Robotics and Autonomous Systems*, 9:181–196, 1992.

[76] P. F. M. J. Verschure, J. Wray, O. Sprons, G. Tononi, and G. M. Edelman. Multi-level analysis of classical conditioning in a behaving real world artifact. *Robotics and Autonomous Systems*, 16:247–265, 1995.

[77] W. Grey Walter. An imitation of life. *Scientific American*, 182:42, 1950.

[78] W. Grey Walter. A machine that learns. *Scientific American*, pages 60–63, August 1951.

[79] C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, University of Cambridge, UK, 1989.

[80] C. J. C. H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.

[81] S. W. Wilson. Knowledge growth in an artificial animal. In *Proceedings of 1st International Conference on Genetic Algorithms and their Applications*, pages 16–23, Hillsdale, NJ, 1985. Lawrence Erlbaum.

[82] S. W. Wilson. The animat path to AI. In J.-A. Meyer and S. W. Wilson, editors, *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, pages 15–21. Bradford Books/MIT Press, Cambridge, MA, 1991.

Appendix A

A.1 Neurons

A.1.1 Basic neurons

The basic neuron type has the state system functionality which is subsequently embedded in all derivatives such as the sensory and motor cells. Updating equations for the membrane potential (MP – in millivolts) are:

$$\begin{aligned}
&\text{state A:} && MP(t+1) = MP(t) - \tau + S(t)\\
&\text{state B:} && MP(t+1) = MP(t) - \alpha + S(t)\\
&\text{state C:} && MP(t+1) = h + S(t)\\
&\text{state D:} && MP(t+1) = MP(t) - \mu + S(t)\\
&\text{state E:} && MP(t+1) = l + S(t)\\
&\text{state F:} && MP(t+1) = MP(t) + \tfrac{BaseMP - MP(t)}{\eta} + S(t)
\end{aligned}$$

where:

$$S(t) = \sum_i w_i \kappa \left(MP_i(t) - BaseMP_i\right) \quad \text{is the synaptic potential (}\texttt{SynPot}\text{)}$$

and:

i is a counter which counts over active pre-synaptic cells

w_i is the synaptic weight from a pre-synaptic neuron

τ is the neuron time constant

$\eta = 1.5$ is the post-undershoot increment rate

$\mu = 25$ is the post-action potential peak-MP decrement

$\kappa = 1/450$ is a heuristically-set learning constant

$\alpha = 20$ is the post-threshold attack increment

$h = 45$ is the post-threshold maximum MP

$l = -69$ is the pre-undershoot minimum MP

Certain of the above parameters (e.g. τ, η) are time-dependent and have been set empirically to suit a range of processor speeds and implementations. However, they may be inappropriate in some circumstances (see, for example, comments on the ARM implementation in Section 6.2).

A.1.2 Sensory neurons

The sensory neurons react to changes in light level and to object proximity.

1. Changes in light level: The LDR sensors feed delta sensory neurons with a sampled value $\lambda(t)$. There are two modes of operation: *proximal* and *distal*. Distal mode operation simply inverts the measurement taken from the environment. However, distal mode depolarizes the MP in response to measured decreases.

Proximal mode:

$$MP(t+1) = \begin{cases} MP(t) + \Delta\lambda & \text{if } \Delta\lambda > 0 \\ MP(t) & \text{otherwise} \end{cases}$$

Distal mode:

$$MP(t+1) = \begin{cases} MP(t) + \Delta\lambda & \text{if } \Delta\lambda < 0 \\ MP(t) & \text{otherwise} \end{cases}$$

where:

$$\Delta\lambda = k[\lambda(t+1) - \lambda(t)]$$

and k is an empirically set scaling factor whose value depends on the particular physical sensor employed.

2. Proximity sensing: IR sensors feed IR sensory cells with a sampled value (approximately) proportional to object range, R, at time t, whence:

$$MP(t+1) = MP(t) + \lfloor sR(t) \rfloor$$

where $\lfloor\ \rfloor$ represents the floor function and s is an empirically set scaling factor whose value depends on the particular physical sensor employed.

A.1.3 Motor neurons

These close the loop between the nervous system and the environment. Motor drive activity is given as:

$$A(t) = \left\lfloor \frac{\pm \gamma MP(t)}{h} \right\rfloor$$

where γ is a scaling constant for the robot hardware (10 for Khepera) and its sign is determined by the requirement for forward or reverse drive.

A.2 Synapses

The basic synapse has functionality which is subsequently embedded in all derivatives such as the habituating, sensitizing and conditioning types.

$$w(t) = \begin{cases} w(t) - \beta & \text{if } w(t) > w_{\text{base}} \\ w(t) + \beta & \text{if } w(t) \leq w_{\text{base}} \\ w_{\text{max}} & \text{if } w(t) > w_{\text{max}} \\ w_{\text{min}} & \text{if } w(t) < w_{\text{min}} \\ w_{\text{min}} & \text{otherwise} \end{cases}$$

where β is the MP recovery parameter and w_{base} is the base weight (typically 0). These are individually set (together with w_{min} and w_{max}, typically ±16) for each neuron.

Noise-free synapse:

$$\texttt{fired}(t) = \begin{cases} \textsc{true} & \text{if state B, C, D} \\ \textsc{false} & \text{otherwise} \end{cases}$$

Noisy synapse:

$$\texttt{fired}(t) = \begin{cases} \textsc{true} & \text{if (state B, C, D)} \ \wedge \left(\frac{MP_p - \theta_p}{h - \theta_p} \times 100 \geq \text{rand mod100}\right) \\ \textsc{false} & \text{otherwise} \end{cases}$$

where p denotes a parent (pre-synaptic) neuron.

A.2.1 Habituating type

$$w(t+1) = \begin{cases} w(t) - d & \text{if state C} \\ w(t) & \text{otherwise} \end{cases}$$

where d is a constant decrement (typically ~ 1).

A.2.2 Sensitizing type

$$w(t+1)_{\text{target}} = \begin{cases} w(t)_{\text{target}} + w(t)_{\text{sos}} & \text{if } \texttt{fired}_{\text{target}} \wedge \texttt{fired}_{\text{sos}} \\ w(t)_{\text{target}} & \text{otherwise} \end{cases}$$

where ‘target’ denotes the synapse to be sensitized (the CS → UR connection in Fig. 2(b)), and ‘sos’ denotes the synapse-on-synapse influence.

A.2.3 Conditioning type

$$w(t+1)_{\text{target}} = \begin{cases} w(t)_{\text{target}} + w(t)_{\text{sos}} \frac{nT}{\psi} e^{\left(\frac{-nT}{\varsigma}\right)} & \text{if } \texttt{fired}_{\text{target}} \wedge \texttt{fired}_{\text{sos}} \\ w(t)_{\text{target}} & \text{otherwise} \end{cases}$$

where nT is a count of sample periods initiated by encountering state C for the target neuron, ψ (= 250) is an empirically-set scaling factor and ς (= 500) is a constant chosen to maximize the effect of conditioning when CS precedes US by 0.5 s.

Author Biographies

Bob Damper was born in Tunbridge Wells, England, in 1948. He obtained his MSc in biophysics in 1973 and PhD in electrical engineering in 1979, both from the University of London. He also holds the Diploma of Imperial College, London, in electrical engineering. He was appointed Lecturer in electrical engineering at the University of Abertay Dundee in 1976, Lecturer in electronics at the University of Southampton in 1980, Senior Lecturer in electronics and computer science in 1989 and Reader at Southampton in 1998. He has also spent periods as Guest Researcher, KTH, Stockholm (1990/91) and Visiting Research Professor, Oregon Graduate Institute, Portland, OR (1996/97). He has wide research interests including speech science and technology, neural computing and systems neuroscience, cognitive modeling, pattern recognition and intelligent systems engineering, and heads the Image, Speech and Intelligent Systems (ISIS) Research Group at Southampton. He currently teaches artificial intelligence, neural computing and electronic design. He was recently Guest Editor for a special issue of the *International Journal of Systems Science* on “Emergent Properties of Complex Systems”. Dr. Damper has published approximately 200 research articles and authored the undergraduate text *Introduction to Discrete-Time Signals and Systems*. He is a Chartered Engineer and a Fellow of the UK Institution of Electrical Engineers, a Chartered Physicist and a Fellow of the UK Institute of Physics, and a Senior Member of the IEEE.

Richard French was born in Newport, Gwent, in 1967. He studied with the Open University while working as a technician with industrial computer systems from 1984 to 1994. He obtained his BSc in computer science with artificial intelligence in 1998 from the University of Southampton, where he is currently studying towards a PhD in biologically-motivated learning in autonomous robots in the Image, Speech and Intelligent Systems (ISIS) Research Group.

Tom Scutt was born in Bognor Regis, England, in 1965. He obtained his BA in philosophy with cognitive studies in 1987 from the University of Sussex, and his PhD in artificial intelligence in 1994 from the University of Southampton. In 1991, he was appointed Lecturer in artificial intelligence at the University of Nottingham and taught for five years in the Department of Psychology before moving to the ICL Institute of Information Technology at Nottingham. Dr. Scutt has published research articles on neural modeling, language acquisition, computational neuroethology and philosophy of mind. In 1998, he left academia to work for Core Design Ltd., where he is currently the artificial intelligence programmer for the best-selling *Tomb Raider* series of computer games.

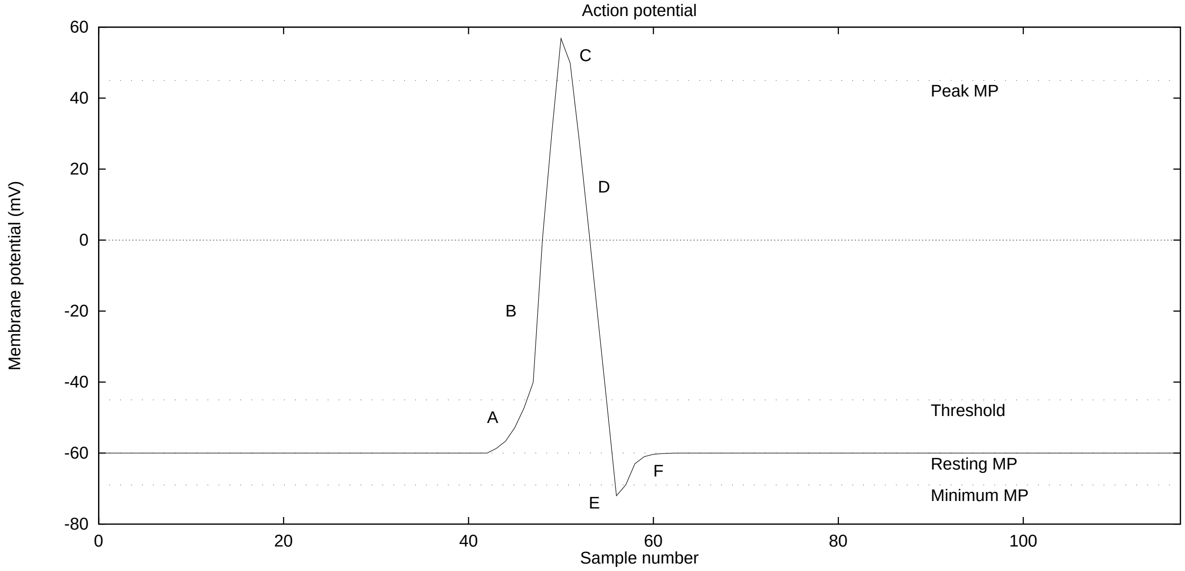


Figure 1: Time evolution of typical action potential (spike) of a **basic** neuron in a Hi-NOON simulation. See text for specification of the states (A..F) passed through by a neuron during firing. Here, the sample period is approximately 4 ms (this varies with the machine on which the simulation runs.)

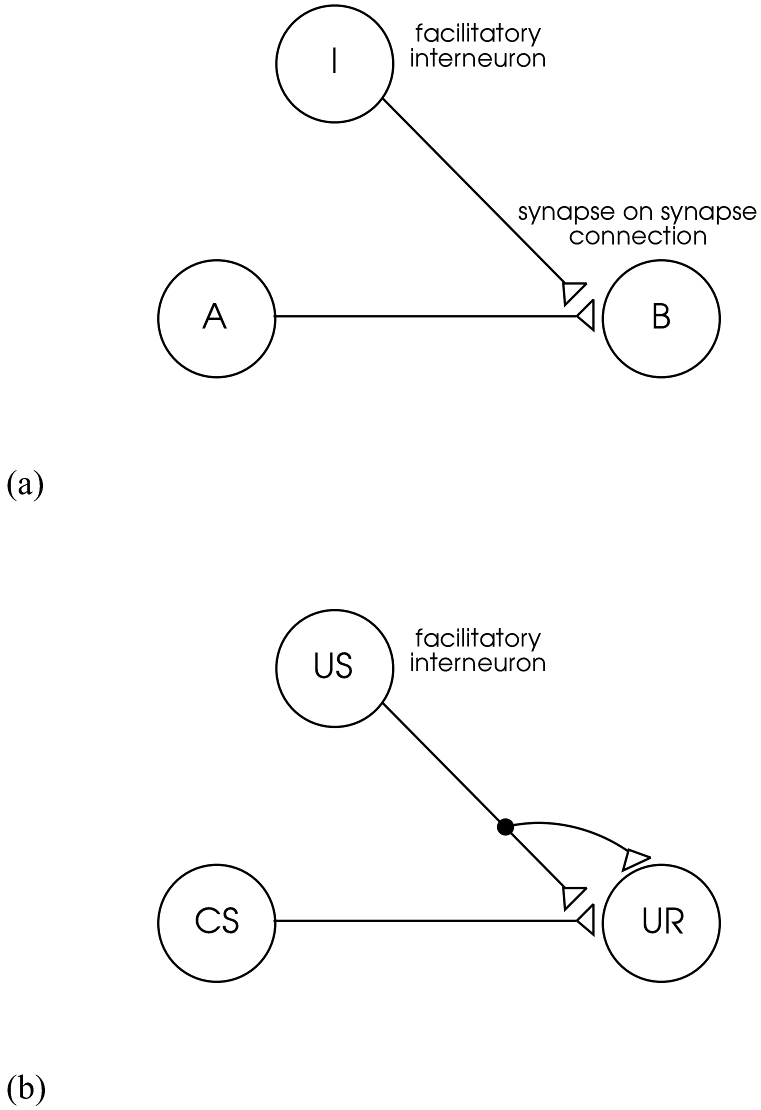


Figure 2: (a) Sensitization is modeled using a facilitatory interneuron I with synapse-on-synapse connection to an $A \rightarrow B$ synapse. (b) Classical conditioning is modeled in a similar way, except for the direct synaptic connection from the interneuron to US and the strength of the facilitation depends upon the length of time since the last firing of the CS → UR synapse.

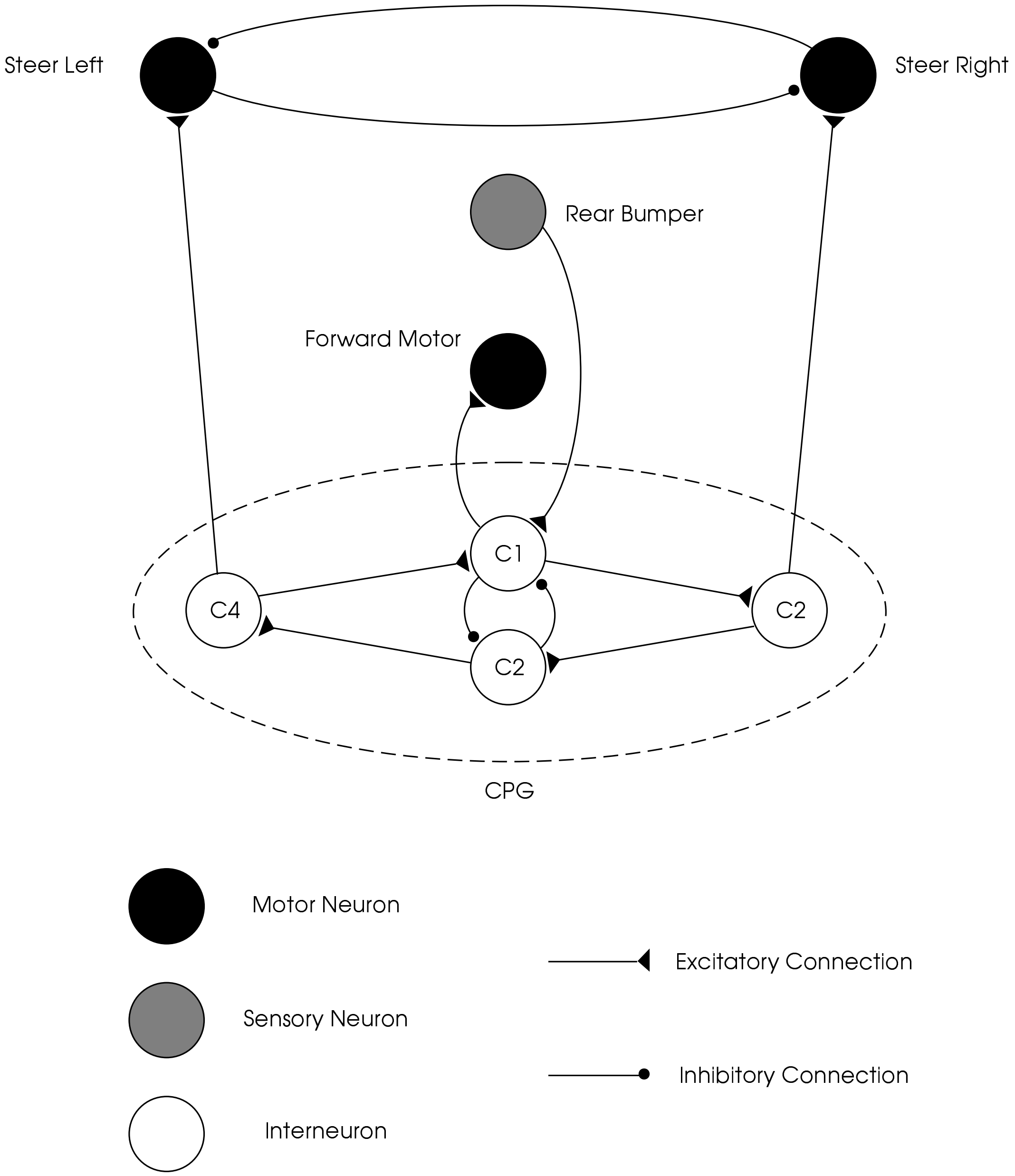


Figure 3: The central pattern generator (CPG) consists of four neurons (C1-C4) with mutual inhibition between two of the neurons, C1 and C3, so that there is a cyclical or oscillatory pattern of firing. There is an excitatory connection from neuron C1 to the forward motor, and from C2 and C4 to the right and left steer motor neurons respectively. The synapses to the steer neurons are of type **noisy** synapses so promoting wandering behavior

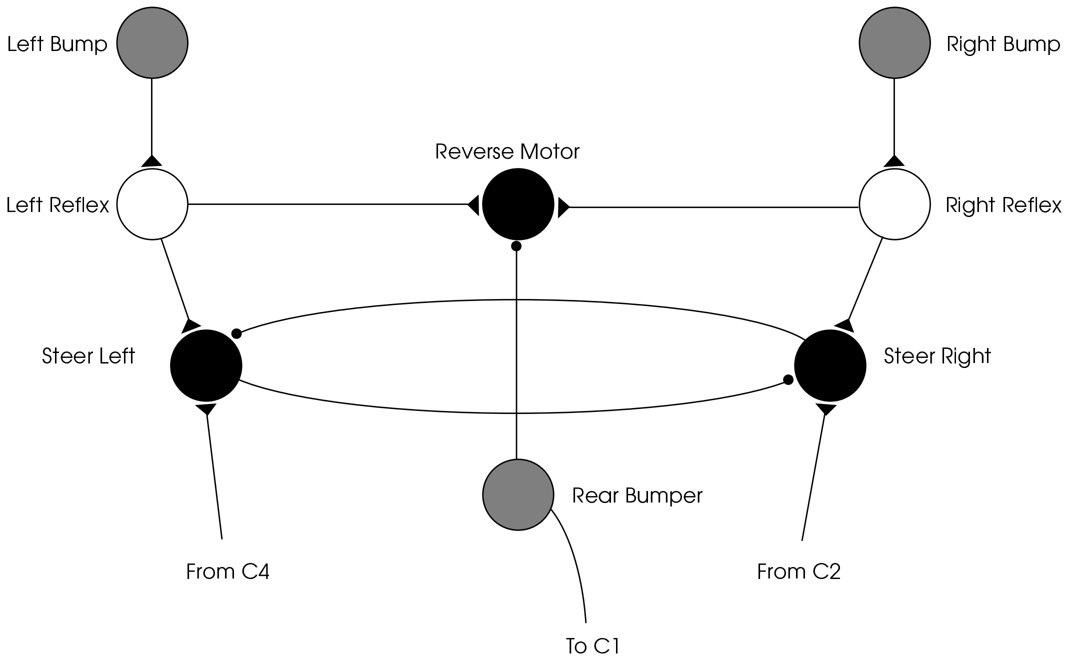


Figure 4: Basic reflex behavior is provided by Bumper Left and Right sensory neurons which are connected to the Reflex Left and Right interneurons respectively. (The actual number of bump sensors depends upon the specific implementation.) Each Reflex interneuron is connected to the Reverse Motor neuron and the ipsilateral Steer neuron. This causes ARBIB to reverse while turning away from the object that it hit before resuming its wandering behavior. (Key as for Figure 3.)

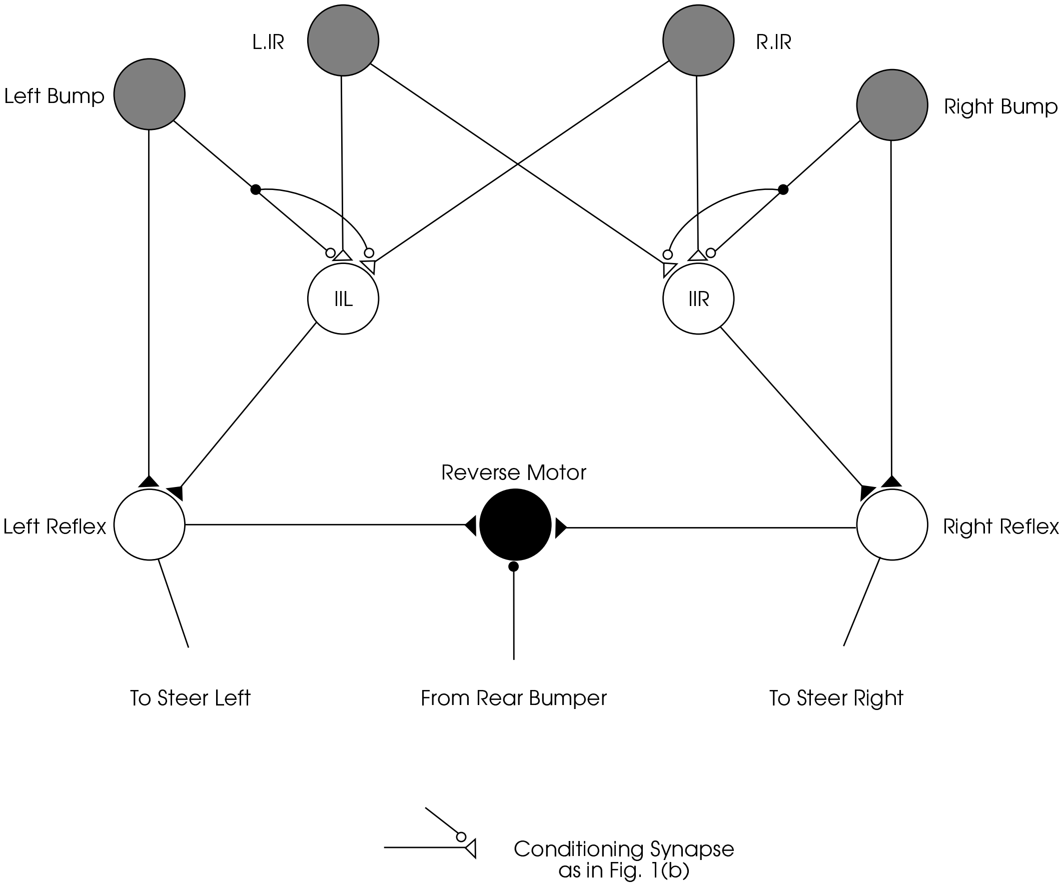


Figure 5: Adding conditioning to the network. Each bump sensory neuron makes conditioning connections to the synapses connecting (from the infrared sensory neurons) to the ipsilateral infrared interneuron (*IIL* and *IIR*, respectively). (The actual number of IR sensors depends upon the specific implementation.) Each infrared sensory neuron (*IR*) makes connections to both of the infrared interneurons: thus, the network is initially wired up neutrally.

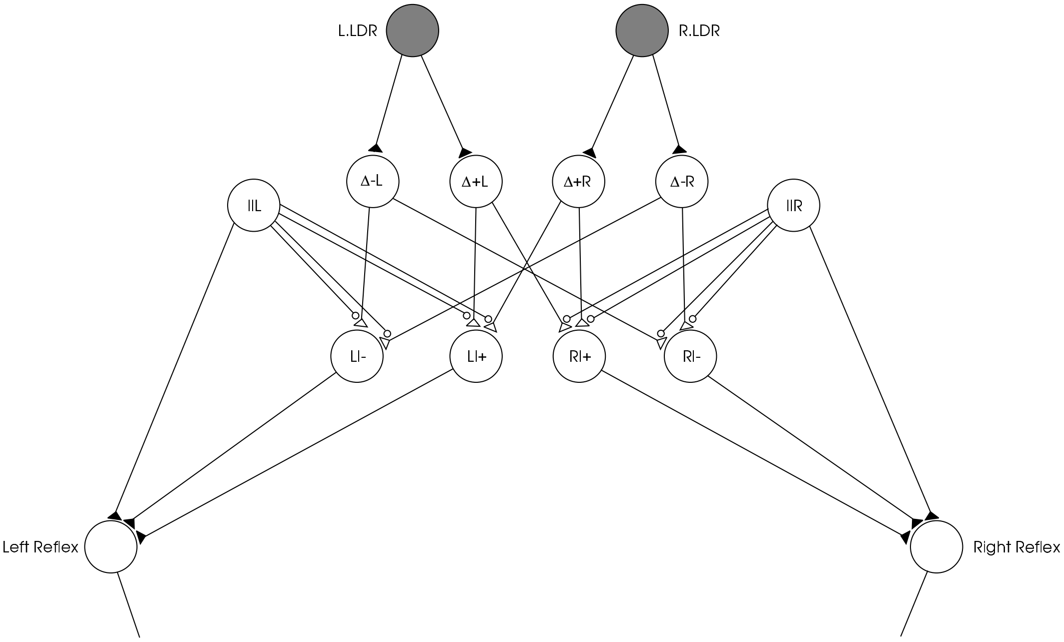


Figure 6: Adding light-dependent behavior to the network. Four Δ neurons make weak connections to the set of light interneurons (*LI*−, *LI*+, *RI*+ and *RI*−) so that each Δ neuron connects to the ipsi- and contralateral neuron with the same sign. These connections can be conditioned by the synapse-on-synapse connections from the infrared interneuron(s), but initially are not strong enough to cause the light interneurons (and therefore the reflex) to fire.

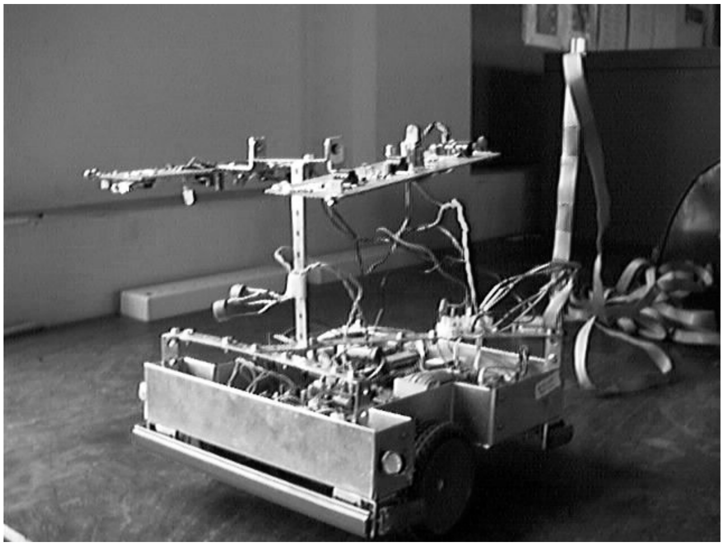


Figure 7: The ARBIB robot (Z180 implementation). Bump sensors are mounted at either end (left and right) of the front bump bar. IR sensors are mounted just above, with the LDRs mounted high on printed circuit boards (where they can be angled down as convenient).

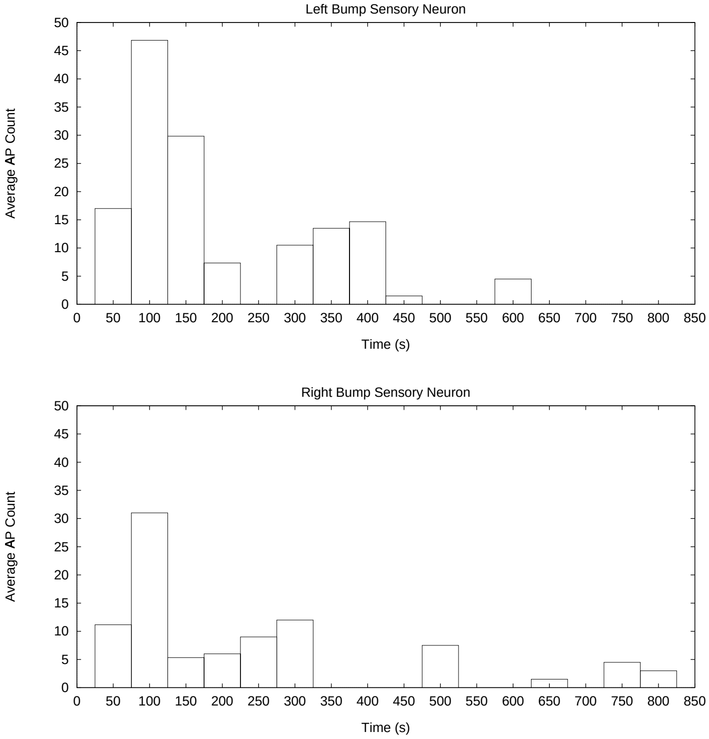


Figure 8: Average counts of action potentials from left and right bump sensory neurons as ARBIB learns in an environment where a hard object (a mug) casts a shadow. Activity reduces with time, providing evidence of sensory substitution through classical conditioning.

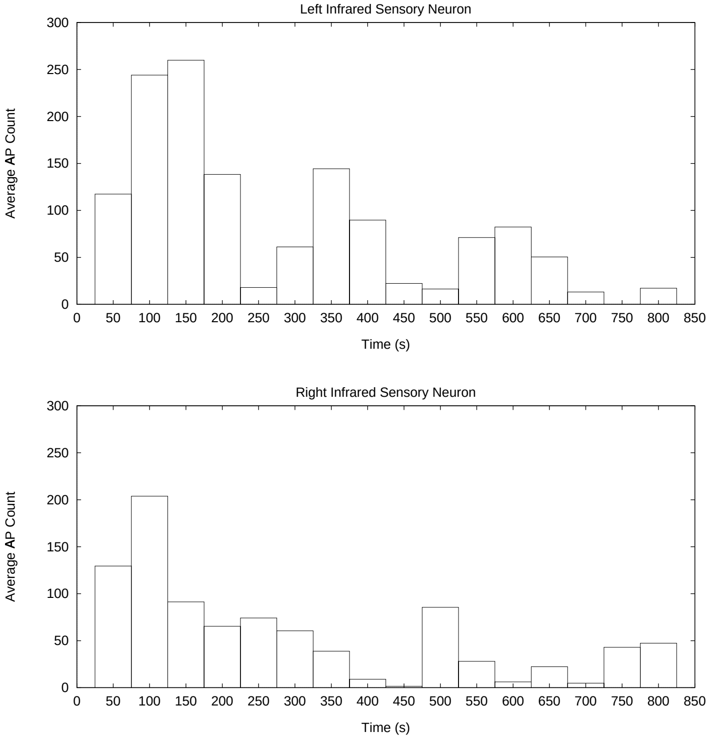
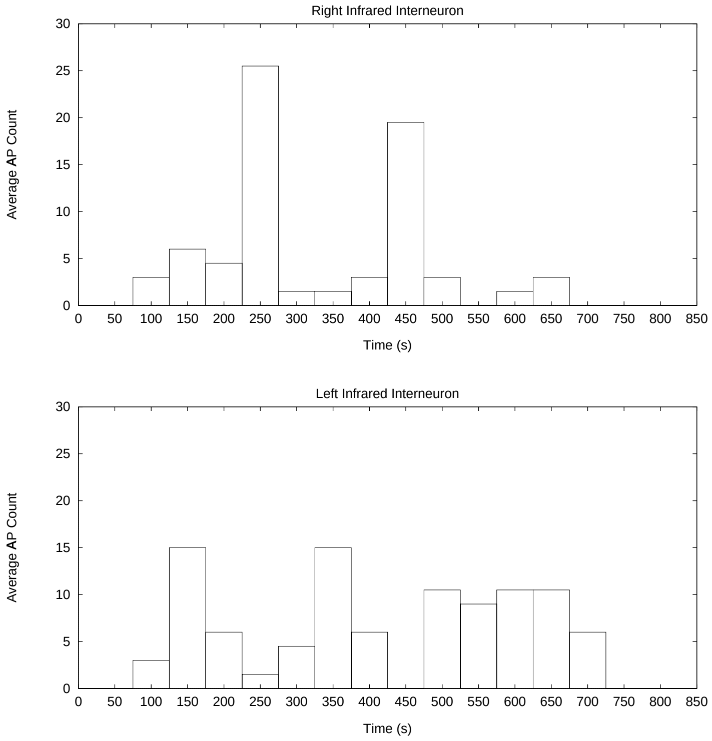


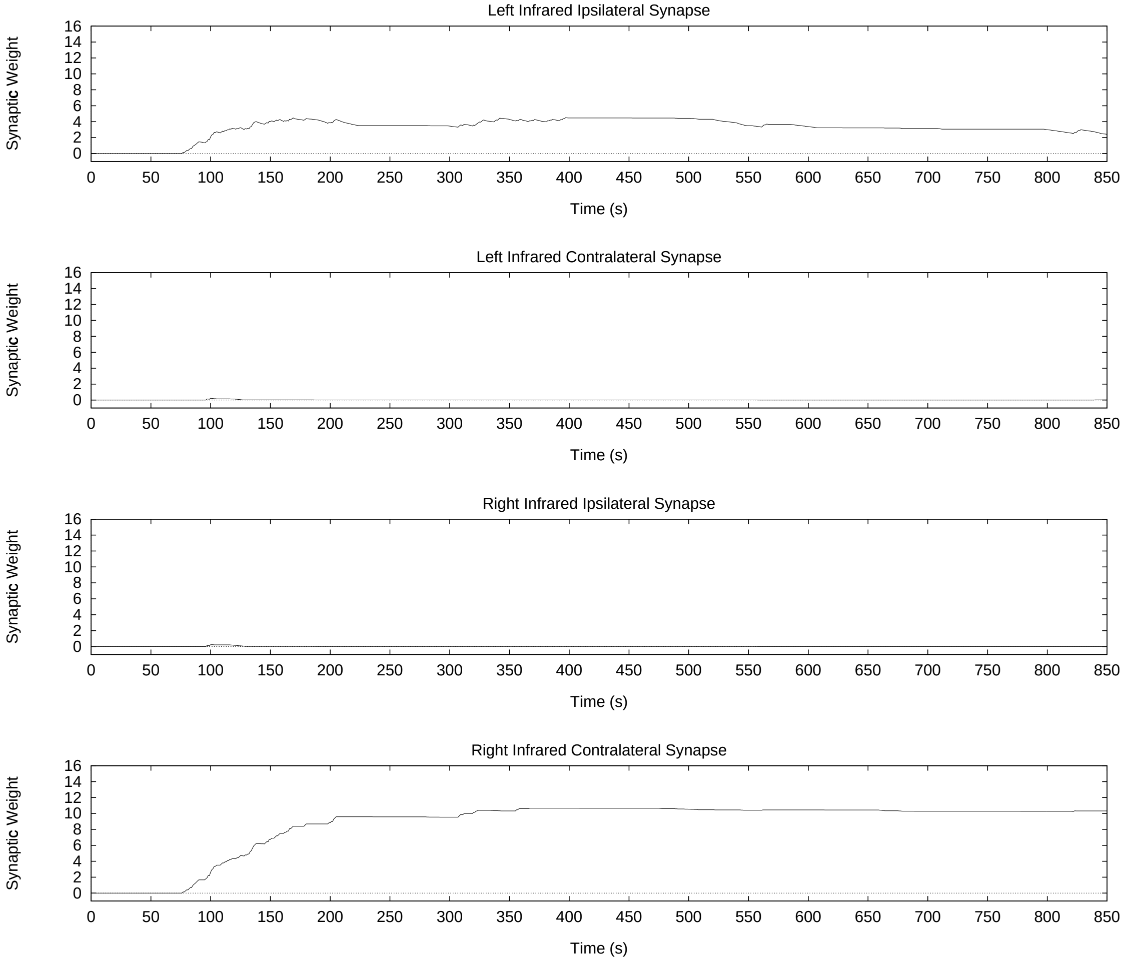
Figure 9: Average counts of action potentials from left and right infrared sensory neurons as ARBIB learns in an environment where a hard object (a mug) casts a shadow. Activity reduces with time, providing evidence of sensory substitution through chained conditioning.



Right Infrared Interneuron

Left Infrared Interneuron

Figure 10: Average counts of action potentials from left and right infrared interneurons as ARBIB learns in an environment where a hard object (a mug) casts a shadow. Activity initially increases with time, providing evidence of the infrared sense substituting for bump sense, but then decreases as the light sense starts to substitute for the infrared sense.

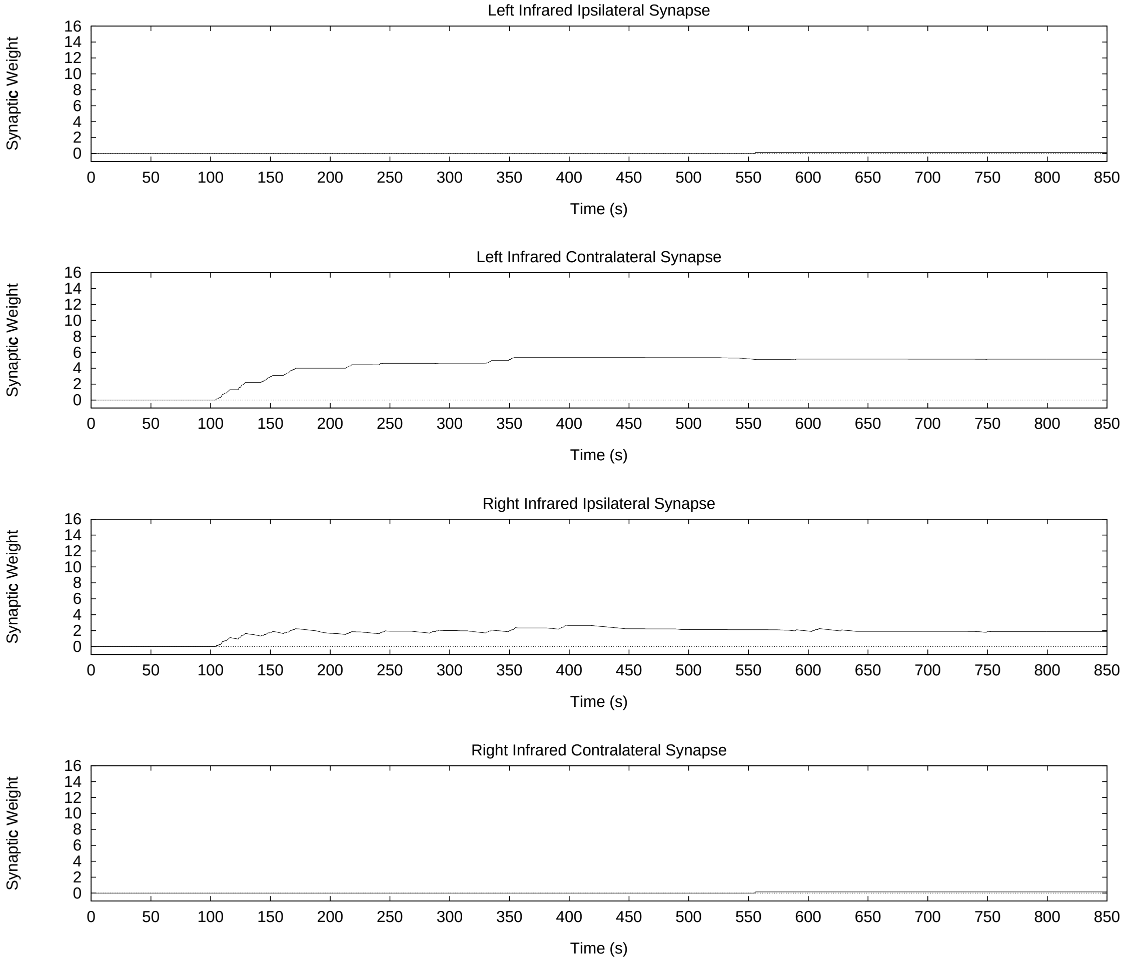

Left Infrared Ipsilateral Synapse

Left Infrared Contralateral Synapse

Right Infrared Ipsilateral Synapse

Right Infrared Contralateral Synapse

Figure 11: Synaptic weight changes between infrared sensory and interneurons for emergent behavior in which ARBIB learns to pass an obstruction on its left side. Only left ipsilateral and right contralateral synapses are strengthened. Weight parameters were: $w_{\mathrm{min}} = -16$, $w_{\mathrm{base}} = 0$ and $w_{\mathrm{max}} = 16$.

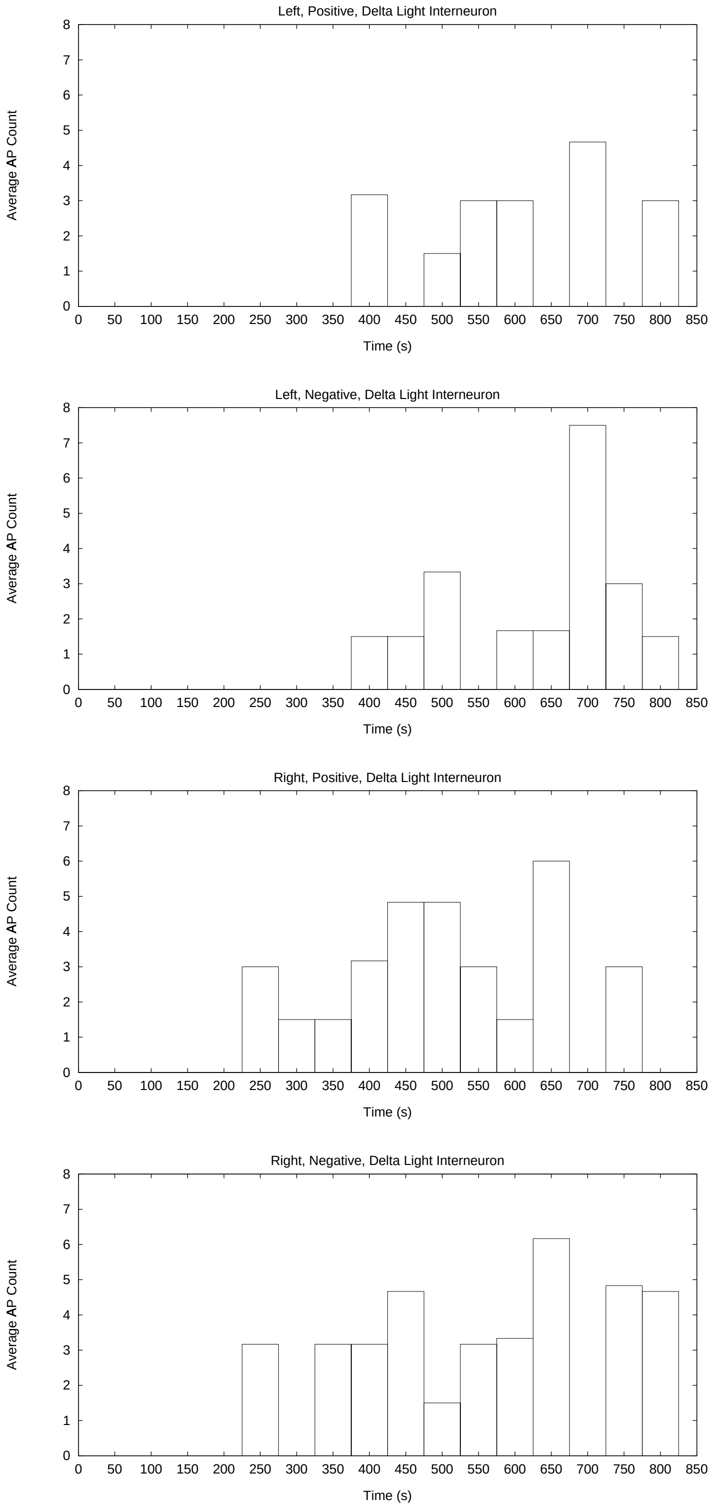

Left Infrared Ipsilateral Synapse

Left Infrared Contralateral Synapse

Right Infrared Ipsilateral Synapse

Right Infrared Contralateral Synapse

Figure 12: Synaptic weight changes between infrared sensory and interneurons for emergent behavior in which ARBIB learns to pass an obstruction on its right side. Only left contralateral and right ipsilateral synapses are strengthened. Weight parameters were: $w_{\text{min}} = -16$, $w_{\text{base}} = 0$ and $w_{\text{max}} = 16$.


Left, Positive, Delta Light Interneuron

Left, Negative, Delta Light Interneuron

Right, Positive, Delta Light Interneuron

Right, Negative, Delta Light Interneuron

Figure 13: Activity in the four delta light interneurons is biased to later in the run, indicating the strengthening of their pre-synaptic weights *after* the establishment of the infrared interneurons as collision detectors.

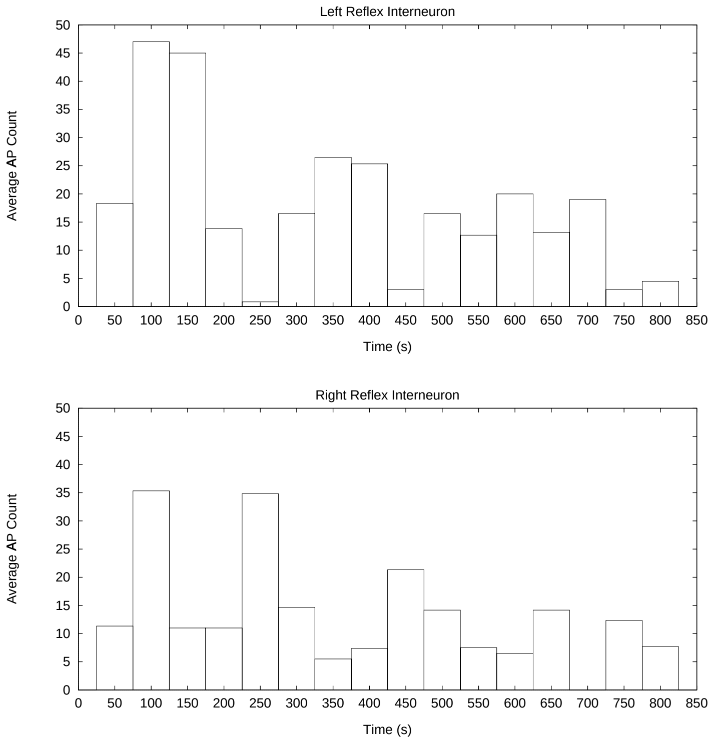


Figure 14: Average activity of the reflex interneurons reduces over time as ARBIB learns to avoid collisions.