An Ethological Model of Learning and Motivation for Implementation in Mobile Robots

†B.E. Hallam, <bridget@aifh.ed.ac.uk>
‡J.R.P. Halperin, and <janh@zoo.toronto.edu>
†J.C.T. Hallam <john@aifh.ed.ac.uk>

†Department of Artificial Intelligence,
The University of Edinburgh,
5 Forrest Hill,
Edinburgh EH1 2QL,
SCOTLAND

‡Department of Zoology,
University of Toronto
Toronto
Ontario M5S 1A1
CANADA

Keywords: learning, motivation, mobile robot, behavioural model

Running title: Robot Learning and Motivation

Abstract

This paper describes a neuroethological model of learning and motivation which is biologically plausible, accounting for many of the behavioral phenomena observed in animals. Unusual predictions from the model have been tested and demonstrated in laboratory Siamese fighting fish. Because the model is mathematically well-defined, it is implementable in a robot or in computer simulation. A trial implementation in a mobile robot was carried out as part of this work. This paper describes the full model, the simplified version programmed into the robot, and the preliminary experiments carried out.
This model is important both to robotics and to ethological research. The study of robot autonomy should be enhanced through an understanding of a complex and realistic model of animal autonomy; and ethological research should benefit from a supply of guaranteed 'naive' agents on which to further study this ethological model.

Roboticists tend to dismiss biological models of behavior as irrelevant to their work or too poorly specified to be implementable. While the latter criticism has undoubtedly been true in the past, recently biologically plausible models have been proposed which are more tightly specified. The model described in this paper is one such.

Halperin's Neuro-Connector model of learning and motivation is based on an unusual interpretation of Hebb's insight that synapses should strengthen when the connected neurons fire together [Hebb, 1949], which is that it is only the *ends* of firing which are significant. This is combined with Sinclair's 'rest' principle [Sinclair, 1981] which states that consolidation of synaptic strength change is interfered with if the neuron does not rest after a bout of activity.

The model was produced as a result of analyzing accounts of observed behavior in many different species. It proposes a mechanism for behavior such as learning, conditioning, and priming of responses. These features are discussed in more detail in [Halperin, 1990]. When the model was completed, further analysis showed that it made some unexpected predictions about behavior, including a new form of conditioning called 'postponed conditioning'. These predictions were subsequently tested and found to be observable in Siamese fighting fish. This model is therefore biologically plausible in that it explains and correctly predicts actual behavior in real vertebrate animals.

The model is an unsupervised connectionist network with unidirectional links. It differs from most connectionist nets by operating with dynamic input and dynamic output and having rather more complex and realistic neurons than those used in most neural network simulations (*e.g.* the Perceptron described by McCulloch and Pitts [McCulloch and Pitts, 1943] and named by [Rosenblatt, 1962]). Another important difference is that each individual input neuron reflects the state of a particular aspect of the environment; the net responds to the firing of particular input neurons rather than the overall pattern of firing.

The model assumes that some form of perception is already done and that the agent is capable of performing the actions necessary for it to accomplish whatever task it has been given. A multiplicity of recognizable sensory states is assumed in order to provide resources allowing the model to choose and refine which sensory states should trigger which actions.

The requirement for many sensory states (which is not the same as many sensors) combined with the limitations imposed by the computational cost of updating weights means that current robots are not capable of the range and detail of behavior with which the model can cope. This contrasts with many currently used computational models of action selection which can typically cope with a smaller range of actions than the robots they are controlling can achieve [Tyrrell, 1993].

In this paper we describe a simplified version of the neuro-connector model in computing terms, with some indication of the extensions to the full model. The features of animal behavior explained by the model are discussed. The robot used and the experiments carried out on the model are described. The results show that even the simplified version of the model is capable of causing some of the behavioral phenomena observable in animals. We are currently investigating a less simplified version which preserves more of these features.

1 Description of the Neuro-Connector Net

Three main types of neurons are distinguished in the Neuro-Connector model — sensory (S), releaser (R), and behavior (B). These neurons are organized into three layers (one for each neuron type) with adjustable-weight connections between the S and R layers, fixed-weight feedback connections between the R and B layers, and fixed-weight inhibitory connections between B neurons, as illustrated in Figure 1.

The structure illustrated in Figure 1 is only part of the net. The behavioral repertoire of vertebrates is extensive so hundreds of B neurons are postulated, each with its own R connected to hundreds of S neurons.

Sensor input is pre-processed to various degrees by hand or by a self-organizing net and is presented (as a sensor value or, more commonly, as logical combinations of sensor or perceptual propositions *e.g.* 'front infra-red on and bumper off' or 'I can see a cup but not a bottle') to S neurons. The amount and temporal sequence of each type of input is used to determine which S neurons should fire. Output from S neurons is combined with information encoded in the current synapse weights, reflecting the previous experience of the agent, and passed to the R neurons. It is these S–R synapses which have adjustable weights and so allow the net to learn.

When a releaser neuron accumulates enough input to fire, its behavior neuron is stimulated. Ethologists should note that non-plastic excitation of the B neurons is used instead of inhibition of an inhibitory link. The B neuron will only fire if the stimulation it receives is enough to overcome inhibition from the B neuron currently active. B neurons inhibit each other so that only one behavior happens at a time; thus it is here that behavior selection occurs.

Each B neuron causes the agent to carry out some behavior equivalent to part of a 'fixed action pattern' in ethological terms. For a robot, a B neuron firing causes some code to run. This code will cause something that one could describe as a 'behavior' under Brooks' terminology [Brooks, 1986b]. This behavior may be dependent upon another sensor reading *e.g.* the robot may turn towards a noise; or it may be a single motor command; or a sequence of small actions; or it may depend on the history of the agent's previous activity; or ...

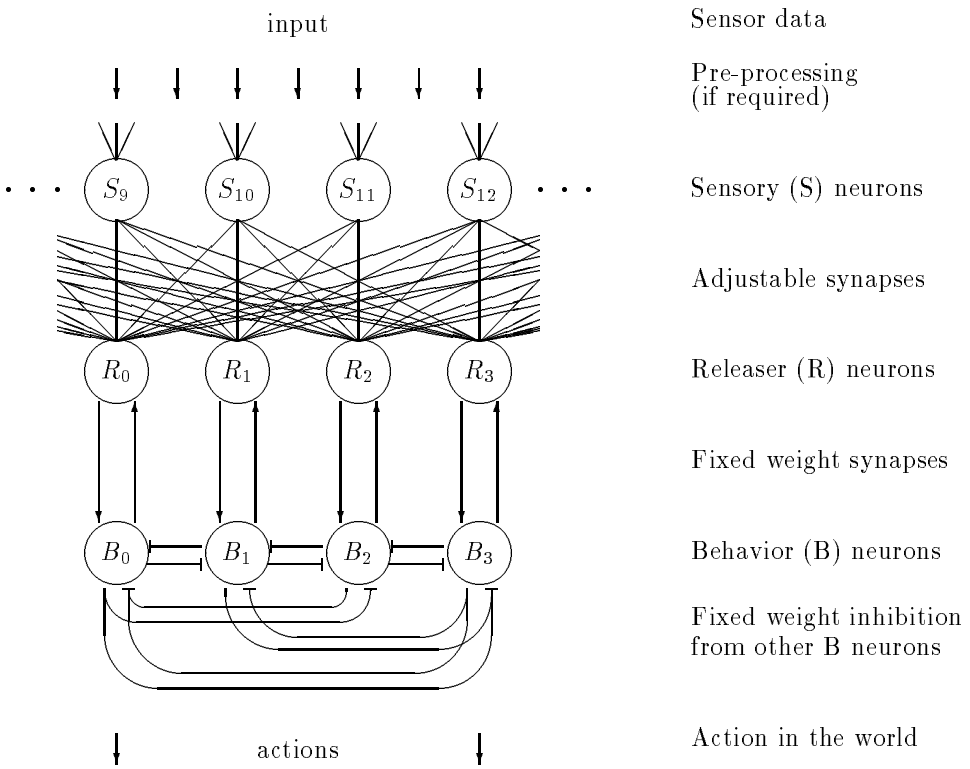

Figure 1: Part of a Neuro-Connector Net

For many behaviors, *e.g.* shutting the fingers, the time taken to complete them successfully is fairly constant. When the 'shut fingers' behavior fires correctly, *i.e.* when the hand is poised with an object between the fingers, the time taken for the hand to grasp the object is roughly constant. When the behavior fires wrongly, *e.g.* the robot identified a picture of the object as the object itself, then the attempt at grasping continues until the robot gives up, which is typically a rather longer time than a successful grasp would take. By measuring how long this sort of behavior takes, it would be possible to have a good idea of whether the behavior was successful or not. In fact, measuring time is the only way that this model has of deciding whether a behavior fired correctly or not and so whether the synapses involved should be strengthened or weakened.

However, for some behaviors, *e.g.* approaching a moving object, the time that the behavior takes is expected to vary considerably, depending upon the distance at which the object it detected and whether it is moving towards or away from the robot. So measuring the length of time that the behavior operates would not help in this case. Instead the synapses in this model measure the time between the S (pre-synaptic) neuron going off and the R (post-synaptic) neuron going off. The S neuron is responding to the state of the environment. The R neuron is switched on by environmental cues but, if its B neuron is selected, it then remains on until the B neuron goes off. The significant sensory state for our example would be something like "suitable object more than 50cm away". The behavior would be something like "move towards specified object until within 50cm". So we could then expect the behavior to finish very shortly after the significant stimulus went off. Since we have managed to constrain our expected time, we can again compare this expected time with the real time difference between the S going off and the R/B going off and obtain a measure of the success of the operation. It is always possible to arrange that the behavior takes an approximately constant time after a particular stimulus disappears, with a little thought.

It is important that the time between the end of firing in the S neuron and the end of firing in the R neuron is fairly constant, as the comparison between this expected time for the behavior and the actual time taken is the only information available to the learning part of the model.

1.1 Neurons

The internal features of the neurons used in this model are illustrated in Figure 2. Note the two registers which contain a 'memory' of recent input and recent activity; and the dynamic nature of the threshold.

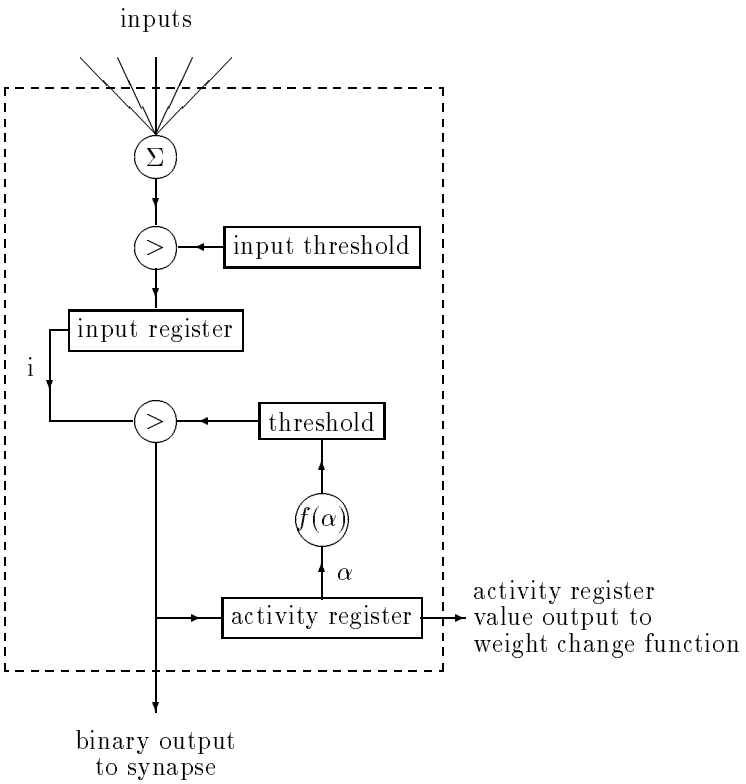

Figure 2: Inside a Neuron

Input to a neuron is summed and this value minus the input threshold value becomes the target value for the exponential decay of the **input register**. These exponential decays are also known as simple first order lag systems, decaying average filters, or smoothing units, and are represented as:

$$\frac{dy}{dt} + \gamma\, y \;=\; \gamma \;\times\; \text{target} \tag{1}$$

where y is the current value of the register, t is time and γ is the time constant or gain of the function. The up gain may be different from the down gain. Note that the target value for the function is not a constant for any of the exponential decays in this model, which makes them unusual versions of their kind. Since the value in the input register reflects not only the current input value but also contains a component of past input values, this register can be said to contain a memory of past input values, weighted towards the current input value by an amount which depends on the size of the gain.

If the value in the input register exceeds the current threshold value for this neuron, then this neuron fires. Each neuron is essentially binary in that it is either on or off at any given time.

Neurons also contain an **activity register** which is adjusted according to a similar exponential function but with 1 or 0 as the target value depending on whether the neuron is on or off. Therefore it contains a memory of the proportion of time that this neuron has been on recently. The value in this register is passed to the synapse weight change function which is discussed in Section 1.3.

The **threshold** value which determines what stimulus intensity is required before the neuron

will fire also changes with time. While the neuron is quiescent the threshold falls so that the neuron becomes more likely to fire. A preset minimum for this threshold value stops the neuron firing in response to sensor noise. When the neuron fires, the threshold is temporarily raised to make it more difficult for the neuron to fire subsequently. The function used is unspecified by the model; for the experiments described here we have used a step function which flips from minimum to maximum after a preset 'adaptation time' and flips back after a preset 'refractory time'.

1.2 Synapses

Synapses consist of a weight, a note of whether this weight should be adjustable, and a record of which is their pre- and post- synaptic neuron. Each S–R synapse also has as a parameter the expected time difference, t_{exp}, between its S neuron finishing firing due to the stimulus disappearing and its R neuron finishing firing due to the successful completion of the behavior released by that R. They also keep track of the actual difference in firing, t_{obs}, between their S and R neurons by recording the current activity register value in their R neuron and the previous activity register value (t_{exp} seconds ago) of the S neuron. The expected time t_{exp} is set such that when a behavior is successful there should be excellent correlation between observed and expected differences. When a behavior is unsuccessful the B neuron and therefore its R neuron generally stay on too long, so the correlation is not so good. Correlation is also bad if either neuron fails to fire.

Most excitatory S–R synapses have adjustable or plastic weights which are initialized to low values; however, some S–R synapse weights are initialized high or even fixed during setting up in order to provide default behaviors under given circumstances. Fixed-weight synapses are normally initialized to high values and are used for inhibitory connections, the R–B feedback loop and some essential reflexes.

Plastic weights can be fixed later if it is desirable for the 'learning phase' to finish, but normally the agent learns continuously.

1.3 Weight Adjustment

Only S–R synapses have adjustable weights, so only these change.

The value in the activity register of the S neuron t_{exp} ago is compared with the value currently in the R neuron activity register and a **coactivation** value returned. The coactivation function is described in detail later. The coactivation value is used as the target for another exponential function of the type shown in equation 1. The resulting value is called the **command register** value. The command register can be considered to contain a memory of the correlation in activation between the S and the R neuron. This command register is used as part of the *gain* for the weight change function:

$$B \;=\; (1-s)^{30} \times\; (1-r)^{30} \;\times \text{the command register value}$$

where s is the value in the S neuron's activity register t_{exp} seconds ago, and
r is the current value in the R neuron activity register

The change in weight is calculated according to:

$$\frac{dw}{dt} \;=\; \begin{cases} Bw & \text{if} \quad B < 0 \\ 2B(1-w) & \text{if} \quad B \geq 0 \end{cases}$$

Raising the values to the power 30 (an arbitrary number) ensures that there is virtually no weight change unless s and r are tiny, *i.e.* both neurons have been off for a while. This implements Sinclair's 'Rest Principle' [Sinclair, 1981].

The values in the S and R neuron activity registers depend entirely upon what has been seen and done — we have no control over these. So in order to make the weights change as we

wish, the command register value has to be correct. Therefore considerable thought has gone into the coactivation function. The desired behavior of this function is as follows.

1. If S and R are well correlated then the coactivation function should be positive long enough for the command register value to become strongly positive. Since the degree of correlation is measured entirely by the values in the activation registers it is biased heavily towards recent activity. Thus when the neurons have been off long enough for the synapse weight to be affected it is the ends of firing in the two neurons which is most significant, and the down gains in the activation registers.
2. If the R goes on without the S the command register value should be near zero. This is so that the same behavior, *e.g.* 'flee', can be triggered by different stimuli.
3. If the S goes on without the R the command register value should be negative, since it this would indicate that the current situation is unrelated to this R's behavior.
4. If both neurons go on but are not well correlated then the command register value should be negative.

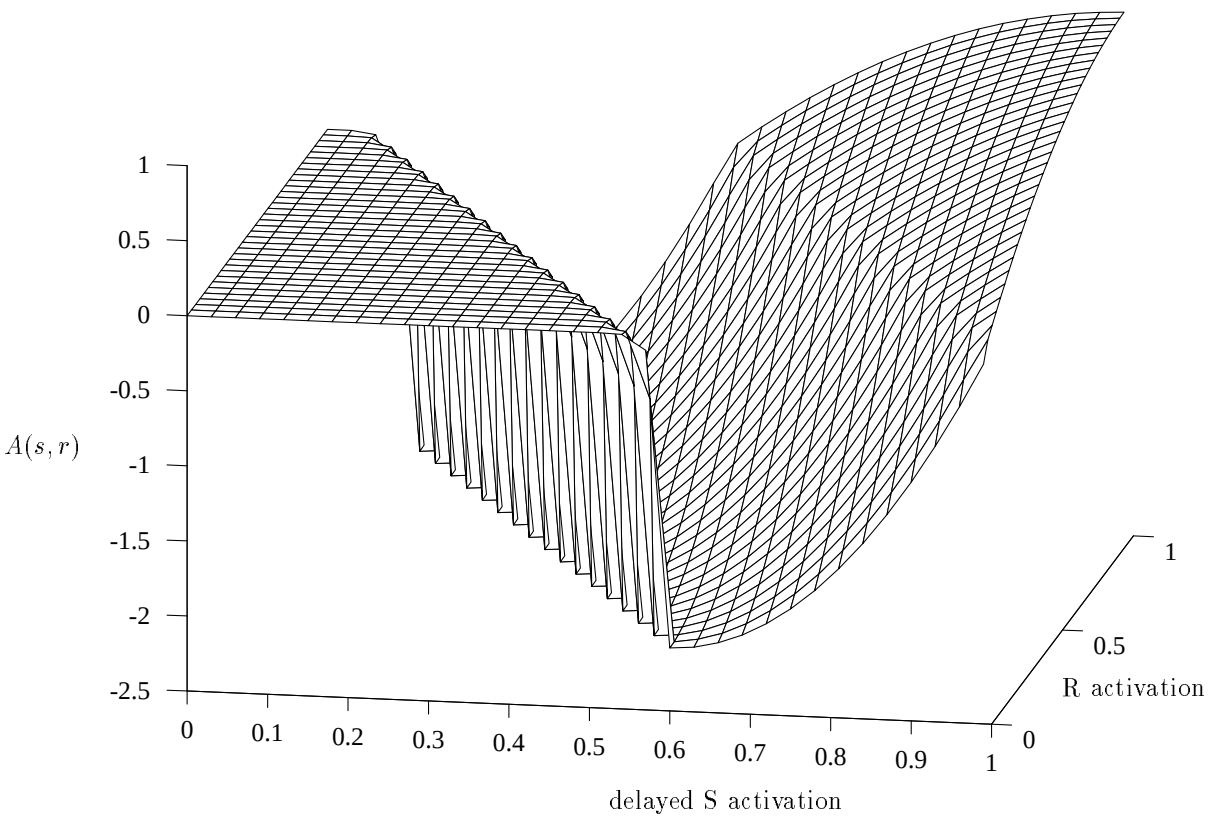

Figure 3: The Control Surface used for Synapse Weight Adjustment $A(s,r)$

The surface chosen to implement these requirements is illustrated in Figure 3. It is described by the equation:

$$A(s,r) = g\left(\frac{2s + r - 1}{2}\right)$$

where s is the value in the S neuron's activity register t_{exp} seconds ago,
r is the current value in the R neuron activity register, and

$$g(u) = \begin{cases} -2.08e^{100(u-0.1)} & \text{if } u < 0.1 \\ 13(u+0.3)(u-0.5) & \text{if } 0.1 <= u <= 0.5 \\ -4(u-0.5)(u-1.5) & \text{if } u > 0.5 \end{cases}$$

Since the activity register functions are continuous functions, the coactivation also changes in a continuous manner. Therefore, if we plot the coactivation value from any series of consecutive moments for any one synapse, we find we have drawn a piece of continuous path across the control surface. It is the precise trajectory followed by this path which determines the value in the command register when weight changes start to occur, and therefore the size and direction of the weight change.

1.4 The Full Model

The Neuro-Connector net is considerably more complicated than has been presented here, as should be expected considering the wide range of vertebrate learning behaviors for which the model claims to account. In this section brief descriptions of most of the extensions are given, a fuller treatment of some of them is given in [Halperin, 1990].

1.4.1 Perceptual Layers

The full model includes extra layers of S neurons which produce the sensor combinations required to make meaningful world-state descriptions from raw sensor data. It is proposed that a self-organizing net may be able to do this in a robot.

1.4.2 Neuronal Pools

The single neurons illustrated in Figure 1 should really be pools of neurons which allow the intensity of behaviors to vary depending upon the number of sub-B neurons firing. R and B neuron pools are disjoint, S neuron pools overlap to some extent within an S layer. There also exist special deficit-detecting R neuronal pools in which each sub-R has a slightly different input threshold. Since these R neurons do not adapt out, the level of firing in the whole pool in effect displays the level of deficit of the substance being measured.

Each pool also has internal structure, being composed of non-overlapping neighborhoods. Inhibitory links between neurons in different S neighborhoods both within a single pool and between adjacent layers of S neurons allow this model to account for stimulus-stimulus conditioning such as was seen in Pavlov's dogs (J Halperin, pers. comm.).

1.4.3 Motivational Layers

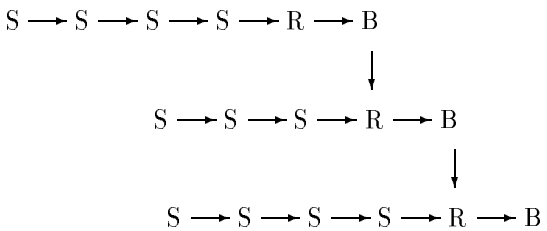

Figure 4: A Vertical Slice Through a Neuro-Connector Net

The whole structure described so far (including extensions) is repeated three times in a hierarchy, connected as illustrated in Figure 4. Only the bottom level generates behavior directly,

the other levels act through their B neurons potentiating R neurons in the level below. The top level of B neurons corresponds to the *motivational* level in ethological terms. Inhibition between B neurons is very weak at this level, to allow multiple motivations to affect the choice of behavior.

1.4.4 Neuron Types

Different types of neurons can be included in this model by adjusting the threshold change function. For *tracking* (or *tonic*) neurons the threshold may be non-adapting or is only raised slowly, so that in the continued presence of the appropriate stimulus the neuron will take some time to switch off. For *burst* neurons the threshold increases quickly causing the neuron to stop firing soon after it starts. These neurons fire as the stimulus appears and periodically throughout the time that the stimulus remains present. For *onset* (or *phasic*) neurons the threshold is adjusted according to the input register instead of the activity register, allowing the neuron to respond more quickly to changes in stimuli. These neurons also have a refractory time which depends on the input register value, preventing the neuron firing again while the stimulus persists.

1.5 Features of this Model

This model employs an unsupervised connectionist or neural net which **learns continuously.** This means that the robot should continue to operate despite changes in circumstances, even where these changes were unforeseen.

The **rate of learning** can be adjusted by changing the gains in the neuronal registers, but is expected to be fast due to the exponential nature of the adjustment functions. In the experiments described in this paper, three or four presentations of a stimulus were sufficient to cause major changes in the behavior. This fast learning rate contrasts with the large number of presentations needed for reinforcement learning *e.g.* [Sutton, 1984].

Behaviors are assumed by this model to be chained together by the setting of initial synapse weights such that the change in sensor conditions caused by the correct operation of one behavior is precisely the stimulus needed to start up the desired subsequent behavior. Thus the agent 'uses the world as its model' in the same sense as the MIT robot Genghis [Brooks, 1990]. This has the advantage that behaviors which are successful both contribute to and reinforce a whole sequence of behaviors, whereas behaviors which are not successful do not disrupt the system. Unsuccessful behaviors are retried a few times (until the B and R neurons responsible for this behavior adapt out), then some other behavior is chosen instead.

There is an increased tendency towards behaviors for which no stimulus is present due to the threshold lowering. This results in the phenomenon of **increased motivation caused by isolation** as reported in [Halperin et al., 1992].

The behaviors selected show **persistence** in the absence of new stimuli, due to the B–R feedback loop. This persistence is limited by adaptation in the neurons causing them to switch off eventually regardless of the sensor situation and allowing less obviously suitable behaviors to surface temporarily. This model thus should show the **time-sharing** described by McFarland [McFarland, 1985]. This prediction is untested at present.

Where this persistence is unexpected by the system (ethologists call this prolongation of behavior *after-discharge*) the connection between S and R is weakened as R doesn't finish firing soon enough. Eventually the connection will not be strong enough for the B neuron to fire, so the previous B neuron is not inhibited, unless another behavior is ready to go at that point in the sequence. Thus the previous B–R pair will after-discharge, weakening that R neuron's S–R connections. This **backward chaining of failures** means that eventually the agent doesn't even start a chain of behaviors known to be rarely successful. This means that the agent does not get stuck retrying a behavior that is for some reason impossible.

Where the B neuron adaptation time is short enough that the R neuron stops firing within

the time window where synaptic strengthening occurs, then this behavior is **self-reinforcing**. Any stimulus which causes such neurons to fire acts as a **reward**, reinforcing the previous chain of behaviors. Normally adaptation times are set to be too long for this positive effect to occur and we rely on the onset of another behavior to interrupt the B–R feedback loop and so stop the current behavior in the time window which allows synapse strengthening to occur.

The formation of newly significant S–R synapses means that stimuli reliably associated with the response tend to cause it, whether these stimuli were set up to be releasers or not. The weakening of S–R synapses where the S fires without the R means that stimuli which are not unique to this behavior will tend to become progressively less significant. This **refinement of stimulus association**, combined with the backward chaining of failure mentioned above, will cause the agent to learn better identification of the correct stimuli.

Since in this model the only criterion for adjusting synapse weights is the time correlation between events, it follows that any stimulus which reliably and accurately predicts a given behavior will cause the behavior to occur. Therefore S–R **conditioned responses** (as can be demonstrated in many species) are easily demonstrated by this model. S–S conditioned responses are also accounted for, but in a less obvious way: they depend upon the prevalence of onset neurons, as described in [Halperin, 1990].

The insight that correlation between ends of firing is more significant than any correlation between the beginnings is one of the distinctive and unusual features of this model. It led to the controversial prediction of **postponed conditioning**. Postponed conditioning is the development of a tendency to display a particular behavior in response to a previously ineffective stimulus which is repeatedly presented just before the after-discharge of this behavior ends. In fish, it was demonstrated by showing fish a mirror to produce a social response called "display". The mirror was removed and the display continued to be performed by the fish. This continued performance of the 'display' behavior after the releasing stimulus was gone would generally continue for a few minutes, the precise time depending on the degree of arousal of the fish when the releasing stimulus disappeared. Just as the fish began to relax a small neutral object such as a pen cover held upright was shown to the fish. After this had been done 5 to 12 times the fish was shown the pen cover alone, and it was statistically likely to display to it. Various control and other confirmatory experiments were carried out [Halperin, 1990] [Halperin and Dunham, 1993].

Another major feature of this model is that **interruptions** interfere with the learning process, since both the pre- and post- synaptic neuron have to be quiet for a time before consolidation of the weight change occurs. [Sinclair, 1981] argues the case for interruptions preventing synapse weight increases in animals.

2 Experimental Section

A program implementing the simplified version of Halperin's net described in this paper was written and used on an autonomous mobile robot to test whether the model was in itself capable of producing the effects claimed for it, or whether there were essential underlying biological processes that had been omitted.

2.1 The Robot

'Ben Hope' is an autonomous transputer-based mobile robot, see Figure 5, developed at the Department of Artificial Intelligence in Edinburgh in the course of another project [Forster, 1991]. It is a cylindrical robot with a diameter of 30cm and a height of 50cm. It has sufficient battery power to run the base for about 30 hours before recharging, and has a separate battery for the head which will support about 2.5 hours of typical activity. The head comprises a card support plate with transputer motherboard, four transputer modules, a Flight 68K processor card and a power converter card; also the head battery and a full duplex 9600baud radio modem. Various sensors have been used with Ben Hope, including a CCD camera, a ring of twelve Polaroid

sonars, an eight-segment bumper, and active infra-red sensors.

The **sensors** used here were six infra-red emitter-receiver pairs and the bumper. The infra-red sensors were thresholded and used as binary sensors, arranged in pairs on top and on the front of the robot as in Figure 5. Each pair consisted of one infra-red unit detecting white objects at about 60cm and one detecting white objects at about 20cm.

For the experiments described here the transputer's facility for parallelism was not used as all the code was contained in one Parallel C 'task'. Increasing the degree of parallel implementation would probably be an advantage for more complex tasks as the model is inherently parallel — however the simplified version of the model described here is easily computationally tractable without the need for parallel implementation.

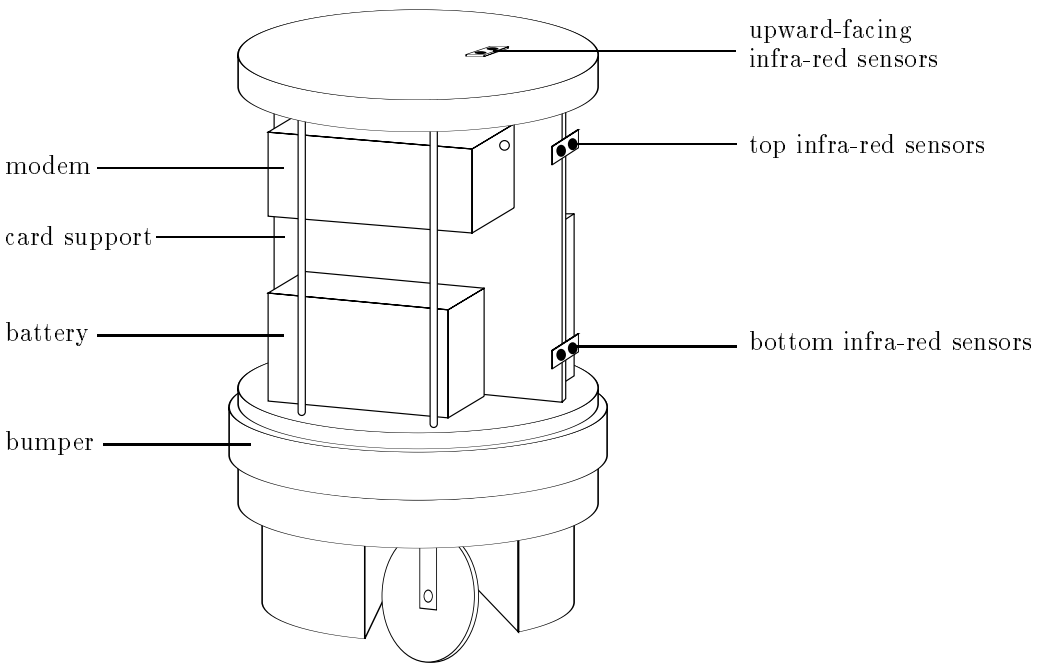

Figure 5: The Mobile Robot Ben Hope

2.2 The Model Used

In the implementation described here R neuron thresholds did not adjust; S and B neuron threshold functions were step functions, flipping from minimum to maximum after an 'adaptation time' described in seconds and flipping back after a 'refractory time' similarly described. Behaviors were prioritized by hand by putting dissimilar weights on the B–B inhibitory links, allowing behavior switching to occur.

This was necessarily a sequential implementation in which the program cycled, doing each operation in order. Between updates of each function it was assumed that sensor readings and the target values for the exponential decays did not change. While obviously a false assumption, it does give reasonably accurate results for most functions, especially since integrations were done from the equations and not approximated. However, this method may not be so accurate for the more complicated weight change functions.

Twenty-one sensory neurons, four releaser neurons and four behavior neurons were used. The sensor propositions that fed into the sensory neurons, the conditions initially set up to release the behaviors, and all the settable neuron and synapse parameters are described in the Appendix. Each R neuron was connected to each S with a low weight of 0.01. Ten releasing

connections as in Table 1 were initialized with weight 0.8. All S–R synapses were plastic and all others fixed. The timestep used (program cycle time) was set to be 0.1 seconds; t_{exp} was set to be 1 second for all synapses.

Name	Priority	Actions Involved	Sensor Conditions	Sensory Neuron Number(s)
ADVANCE	0	Approach object	Far top and/or far bottom	1, 8, 11 and 17
WIGGLE	1	Back 8cm, wiggle back and forth	Near bottom and near top else near bottom and far top, with or without bumper	5, 6, 13 and 14
AVOID	2	Avoid object — back 15cm, turn 90°	Bumper only	3
AFTERPAT	3	Back, turn 180°, pause	Upwards-pointing left and right infra-red sensors	0

Table 1: The Behaviors Used and their Releasing Stimuli

The **behaviors** implemented were as listed in Table 1. Note that the wiggle behavior is set to respond to four different perceptual conditions corresponding to walls with and without boxes in front. The reward stimulus was a pat on the 'head', as sensed by both the upward-pointing infra-red sensors. Note also that none of these behaviors were active unless specifically triggered by their R neurons.

2.3 Early Tests

The robot was tethered while initial experiments were carried out to check that the model behaved as expected. Infra-red-reflective objects were presented by the experimenter to stimulate the sort of sensor readings expected from lab encounters.

S neurons were stimulated for 0.5 seconds by the experimenter moving an infra-red-reflective object in front of the appropriate sensors. The robot used a radio link to send information about the operation of the net to a monitor screen.

2.4 The Effect of Consequence Timing on Synapse Strength

This first experiment was designed to ensure that both strengthening and weakening of synapses occurred according to the arrival time of significant consequences.

For this experiment two S–R synapses were investigated, one initialized to a high weight and one to a low weight. In this case, S_0 – R-AFTERPAT was initialized to 0.8 and S_{19} – R-AFTERPAT to 0.01. AFTERPAT causes the robot to back up and turn through 180° and was "released" by the upward-facing infra-reds being stimulated (S_0 being on).

The experiment consisted of a series of test runs using different delays between the ends of firing in the S and the R neurons (different t_{obs}). This variation was achieved by changing the adaptation time in the B neuron (B_MAX_ON_TIME) and thus the length of time that the R neuron is held on by the B–R feedback loop. The step threshold simplification proved useful here in enabling the time between the ends of firing in the S and R neurons to be precisely known. Note that negative t_{obs} cannot be investigated by this method, as the S neuron will

hold the R on even if the B has timed out. After each test run the final weight of each synapse was collected. These final weights are plotted against t_{obs} to form the graphs shown. Thus each point on a graph of a single synapse comes from a different test run.

When t_{obs} is small, this corresponds to the situation when the S neuron is on 'too long' so weakening is generally expected. When $t_{obs} = t_{exp}$, the time correlation is perfect and strengthening is expected to be maximal. When t_{obs} is large, R is on 'too long' and some weakening is expected which fades out as t_{obs} gets even larger.

All neurons were stimulated simultaneously, the S neurons for a constant 0.5 seconds, long enough that the activity register was quite high. The R neuron was switched off a known time after the S neurons, by fixing the adaptation time in B-AFTERPAT. The adaptation time was fixed for each test run, and the net was reset between each test to provide the same starting conditions for each adaptation time.

Adaptation times in the range 0.7 to 3.9 seconds were tested, each twice. After sufficient time had elapsed that the neurons had turned off completely and the weight change had been consolidated, the stimulus was presented again so that all the neurons were restimulated. Both the final and the intermediate synapse weight were recorded for each adaptation time used.

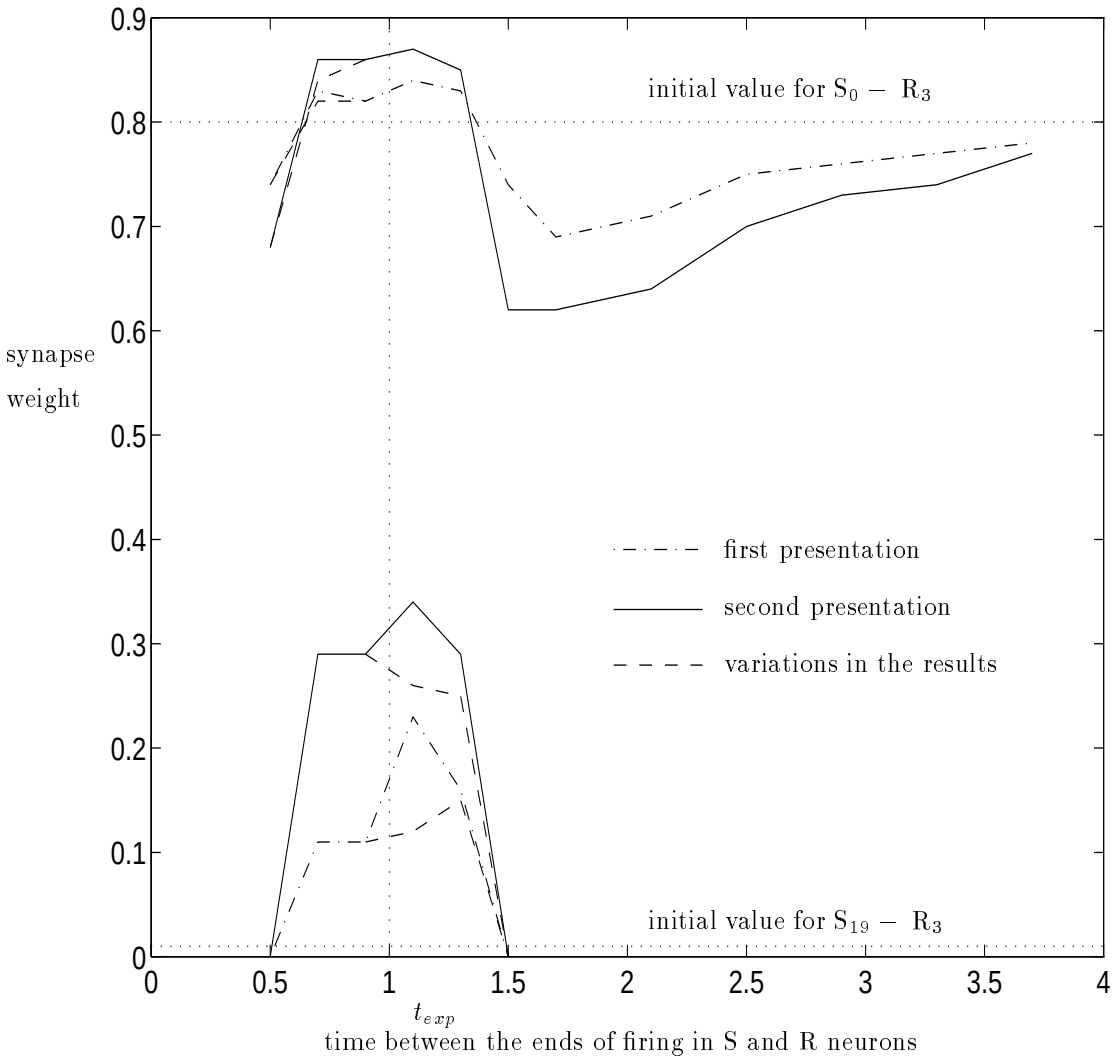

Figure 6: Synapse Weights Vary with Timing

The results are given in Figure 6, and show that synapse weights do strengthen and weaken depending upon the difference between the times that the S and R neurons finish firing. The

overall shape of the graph is exactly that predicted by the model. When the actual difference in ends of firing between the S and R neurons is close to t_{exp}, then the weight increases, demonstrating learning. Just outside the strengthening window the weight decreases significantly, demonstrating unlearning. As the time mismatch becomes greater, the weight changes very little. This means that, if the R neuron comes on but the S neuron hasn't been active recently, the weight doesn't change. This is to enable behaviors to have more than one releasing stimulus without the stimuli interfering with each other.

The non-smooth nature of the peak has been shown to be an artifact of using a sampled system.

2.5 Postponed Conditioning

This experiment was designed to show that postponed conditioning does indeed occur as predicted by the model. Postponed conditioning is a type of associative learning where a previously neutral stimulus shown towards the end of a behavior comes to cause that behavior (see Section 1.5).

The adaptation time in the R-WIGGLE neuron was at its default of 2 seconds, which meant that the wiggle behavior continued long enough to cause weakening of the weight in the one-second synapses attached. However, the bottom near infra-red sensor was stimulated such that its S neuron stopped firing t_{exp} before the wiggle ended.

Within four presentations the bottom near infra-red sensor was sufficient to stimulate wiggle behavior alone.

2.6 A Mobile Experiment

The task for this experiment was for the robot to distinguish walls with boxes against them from walls without.

The net was set up as before with neuronal parameters as in the Appendix and t_{exp} set to be one second for all synapses. All tethers, power cables, *etc.* were removed from the robot. The robot was made to advance towards a wall where it would wiggle, due to the stimulation of R-WIGGLE by the S neurons connected to near infra-red sensors. A wall with a box in front stimulated the bottom near infra-red sensor while the top near infra-red was not triggered; without the box both near infra-reds were triggered at the same time. When the robot wiggled in front of a wall with a box, this behavior was 'rewarded' by 'a pat on the head', *i.e.*, by triggering both upward-pointing infra-reds. This works as a reward because the AFTERPAT behavior is self-reinforcing and inhibits the WIGGLE behavior, thus switching it off at the right time. When the robot wiggled in front of a wall without a box, this behavior was not rewarded.

After 4 unrewarded wiggles for walls, wall presentation ceased to elicit wiggling. Wiggling for boxes against walls continued for the duration of the experiment if the pat was correctly timed.

During preliminary attempts at this experiment, ADVANCE was stimulated by bringing an infra-red-reflective object within sight of the far infra-red sensors, and moving this object such that the robot was led to the wall. However this caused the ADVANCE behavior to time out and weakened the response to the extent that the robot ceased to advance towards 'walls'. In later experiments this problem was avoided; one of the advantages of using robots as experimental subjects is that they can forget all they have learned just by the experimenter restarting the program.

2.7 Discussion of Experimental Results

The results shown in Figure 6 show that, if the difference between the ends of firing in S and R is correct, then the synapse weight increases as expected. Where R turns off after the correct

time due to the adaptation time in the B neuron then this behavior will self-reinforce and act as a reward or end-of-sequence behavior. Most R neurons are not switched off in this way, but by another B starting up and inhibiting their B neuron. Such behaviors will continue too long ('after-discharge') if no other behavior is released, causing weakening of the synapses with that R neuron. Weakening is greatest when t_{obs} is only just outside the strengthening window. Therefore the model predicts that behaviors which can be made to after-discharge for a long time will fade out less fast than behaviors which only ever after-discharge briefly. In a robot, the unlearning characteristics of each behavior can therefore be adjusted by changing just one parameter — the adaptation time.

Note that t_{obs} is not calculated as simply as (B on time − S on time), even when the neurons all start with all registers at zero. The reason for this is that it takes some time for the releasing stimulus to cause the B neuron to come on. Assuming that all registers start at zero, the time taken for the R neuron to switch on once the S neuron is on is given by:

$$(S-R)\ on_delay = \frac{1}{\text{input register up gain}} \times ln\left(\frac{\text{input register target}}{\text{input register target} - \text{R threshold}}\right) \quad (2)$$

For the starting values used here,

$$\begin{aligned}(S-R)\ on_delay &= \tfrac{1}{9} \times ln\left(\tfrac{0.8 + 0.01 \times \text{number of low weight synapses on}}{0.8 + 0.01 \times \text{number of low weight synapses on} - 0.5}\right), \text{ and}\\ (R-B)\ on_delay &= \tfrac{1}{9} \times ln\left(\tfrac{0.8}{0.8-0.5}\right)\\ &= 0.1090\end{aligned}$$

The timestep used in this implementation was 0.1 seconds, so two program cycles would be required for the value in the B neuron input register to pass its threshold value once the R neuron had come on. However, the addition of low-weight synapses to the equation when calculating the (S – R) on_delay was sufficient to bring the R neuron on within one timestep of the S neuron firing. In this experiment the R neuron would have come on one cycle after the S neuron and the B neuron would have come on two cycles later. This means that the value of t_{obs} is

$$\text{B on time} - \text{S on time} + 0.3$$

with these gains and timestep size.

The dramatic increase in the low (S_{19} – R-AFTERPAT) synapse weight shows that previously insignificant stimuli can become important quite quickly. In the postponed conditioning experiment of Section 2.5 the robot learned to respond to the bottom near infra-red sensor within four presentations, which is an enormously fast learning rate compared with other robot learning methods. This production of a 'new' S-R connection forms the basis of one type of classical conditioning. Coupled with unlearning, it causes the set of stimuli predicting a behavior to be refined until only the most specific and reliable are used as releasers. This has the consequence that, if the designer has missed some useful and reliable predictor of some behavior, the system will correct itself. Weakening was not easy to show here, as the synapse weight started so low and the robot was set to display weights to only two decimal places. The slight variation in the results at any given time relationship was probably due to variations in the precision of the stimulus presentation timings.

The width of the time window where strengthening occurs was 0.8 seconds for the neuronal parameters used (given in the Appendix). It would be useful to be able to adjust this for behaviors which cannot be made to have such a constant time relationship with their releasing stimuli.

The experiment of Section 2.5 showed that postponed conditioning can be demonstrated in the robot. Postponed conditioning was only discovered as a result of the development of this model, and remains a rather distinctive feature of it. The demonstration of this feature in the robot therefore indicates that the implementation is faithful to the model, and that the observed phenomenon of postponed conditioning in animals is not likely to be due to unspecified stimuli.

The mobile experiment started off being rather frustrating, as the robot would not follow the experimenter to the wall. However we soon realized that this, too, was a consequence of

the fast learning demonstrated by this model — indeed, this backward chaining of failure is one of the features of the model. Having sorted this out, the robot learned to distinguish between walls with and without boxes in only four presentations. Again, the speed of learning was very impressive. In fact, learning at this speed may cause a robot to learn responses to sensor noise despite the buffering effect of the input register; or to learn correlations which are really coincidences. The learning rate may therefore need to be reduced in further experiments.

3 General Discussion

In the implementation used here, behavioral sequencing was achieved by using dissimilar B – B inhibitory weights, such that 'more important' behaviors could always interrupt and inhibit 'less important' behaviors but not *vice versa*. In this way a chain of behaviors was produced, with each correct new behavior interrupting the previous one. Such chains will be stable if the final, highest priority behavior is self-reinforcing, otherwise they will decay away. While this hand-setting of behavioral priorities is acceptable for many robot applications (and is often done, *e.g.* [Brooks, 1986a]), it will not work where relationships between behaviors are complex. The use of neuronal pools as specified in [Halperin, 1990] may solve this problem, but is unlikely to be computationally tractable, at least until parallel hardware implementations are available. For robot work an extension to analogue neurons may be necessary, or an investigation of other principled neuroethological models of behavior switching.

Training a robot which uses a Neuro-Connector net can be done in several ways ranging from traditional one-stage programming through to a more teacher-pupil style. Obviously it is possible to set t_{exp} for the various synapses and to set 'releasing' high-weight S–R connections into the program at the start, and even to get these right most of the time. However, where the programmer has less information about tasks or internal representations, or otherwise finds it difficult to imagine what the robot sensor readings will be at various times, then other training methods are possible. Where t_{exp} is not known, then multiple synapses with different t_{exp} can be set up between each pair of neurons, and the robot run through its tasks by hand. Those synapses with 'correct' t_{exp} will strengthen while the others will weaken. Where computing power is limiting, low-weight members of a synapse set can be pruned afterwards. Alternatively, an internal clock or even a human with a stopwatch can provide information about when stimuli change and when behaviors start and finish, enough to set t_{exp}. Once t_{exp} are set correctly, it is easy to train high-weight S–R synapses by running the robot through its tasks a few times, triggering the required behaviors by hand. After this it will have learned an approximate representation of the circumstances under which each task should be performed.

This model is exciting to roboticists because of the high rate of learning coupled with stimulus refinement. This makes human-like teaching methods practical, where a robot is trained by merely being 'told' what to do when. The system is therefore user-friendly and trainable by non-roboticists. Continuous on-line learning allows the approximate representations of reinforcing situations, whether formed during training or programmed in, to refine themselves adaptively at run-time. This allows the robot to adjust to gradual changes in the environment or in its own sensor readings.

For ethologists, the possibility of testing ethological models on robotic 'simulated animals' is exciting. Robots provide an invaluable extension of mathematical modeling and computer simulation. This extension is critical for two main reasons; first, because ethological and neuroethological models are so complex, and second, because these complex ethological systems interact with equally complex environments. With robotics, ethologists will be able to test their complex models and see the behavior they predict with the standard observational techniques of ethology, with less fear of over-interpretation or mis-interpretation than with computer simulation results. Another advantage of using robots is that the environment does not have to be simulated and therefore nearly all available effort can go into creating a faithful implementation of the model and designing suitable experiments.

Much further work is necessary. As previously mentioned, the behavioral sequencing problem must be explored. The sensitivity of learning to parameter settings should be investigated,

and alternatives to hand-setting of t_{exp} must be studied. For ethological studies, the ability of the model to produce the types of behavior claimed should be confirmed and its independence of unspecified, underlying biological processes should be demonstrated. The effect on behavior of modeling different neuron types, *e.g.* onset, offset and tracking neurons, in various combinations should also be investigated. These investigations could most usefully be carried out first in computer simulation before transfer onto robots.

A more general implementation suitable for investigating many of these points has already been written and is under test on computer. A second robot similar to Ben Hope, which can be devoted to this project, is being built increasing the amount of experimentation time available.

It should be noted that this series of experiments required many untutored subjects. This would have been expensive to achieve with real animals due to difficulties in preventing their learning relevant information beforehand. With a robot, it can be guaranteed that the subject has no prior training or knowledge just by switching the robot off between tests. Using a robotic system allows ethological models to be tested with greater rigor than is usual, and over a wider range of testing conditions and parameters. However, comprehensive animal testing is still required if any sort of ethologically-relevant conclusions are to be drawn, since robots and animals perceive and act on their environments in quite different ways. Ideally, the two methods should complement each other.

4 Conclusions

The experiments described above show that even the simplified version of the model implemented here is sufficient for causing:

- both strengthening and weakening of synapses,
- fast learning,
- stimulus refinement,
- backward chaining of failure, and
- postponed conditioning.

given that a suitable set of sensor propositions and behaviors are already available.

The rapid learning demonstrated by the system makes this model exciting for those interested in controlling robots. The ability to refine the stimuli triggering behaviors is also useful, in that it allows the robot to respond to a changing environment and obviates the need for accurate knowledge of a robot's internal representations of events. A working system can be tailored to specific tasks by non-roboticists, due to the types of training method possible.

The ability to test the model on a robotic 'simulated animal' is exciting for ethologists, as extensive testing can then be done without using up a large supply of suitable animals. Verification of the model is therefore easier and less expensive, so can be done with greater rigor. Animal tests are still necessary, as a robot interacts with its environment differently from an animal, but these can be reserved for confirming the more unusual results and predictions from the simulation.

Much further work is necessary, and has been started. An extension to analogue neurons will probably be necessary before complex robotic tasks can be controlled using this net.

Acknowledgements

Bridget Hallam acknowledges receipt of SERC grant number 9130847X. Janet Halperin acknowledges receipt of SERC grant number GR/H/23344. All authors thank the Department of Artificial Intelligence, University of Edinburgh for computing, workshop and office facilities.

Thanks go to the people who reviewed this paper and commented on its intelligibility: Gillian Hayes, Ashley Lotto and Martin Westhead.

References

[Brooks, 1986a] Brooks, R. A. (1986a). Achieving artificial intelligence through building robots. A.I. Memo 899, MIT.

[Brooks, 1986b] Brooks, R. A. (1986b). A robust layered control system for a mobile robot. *IEEE journal of Robotics and Automation*, 2:14–23.

[Brooks, 1990] Brooks, R. A. (1990). Elephants don't play chess. *Robotics and Autonomous Systems*, 6:3–15.

[Forster, 1991] Forster, P. (1991). Hardware description of the mobile robot "Ben Hope". RUR working paper 5.1, Dept of Artificial Intelligence, Edinburgh.

[Halperin et al., 1992] Halperin, J., Dunham, D., and Ye, S. (1992). Social isolation of *betta splendens* increases primed social display but decreases display readiness. *Behavioural Processes*. In press.

[Halperin, 1990] Halperin, J. R. (1990). *A Connectionist Neural Network Model of Aggression*. PhD thesis, Department of Ethology, Toronto University, Canada.

[Halperin and Dunham, 1993] Halperin, J. R. and Dunham, D. W. (1993). Postponed conditioning: Testing a hypothesis about synaptic strengthening. *Adaptive Behaviour*, 1(1):39–64.

[Hebb, 1949] Hebb, D. (1949). *The Organization of Behavior*. Wiley & Sons, New York.

[McCulloch and Pitts, 1943] McCulloch, W. S. and Pitts, W. H. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133.

[McFarland, 1985] McFarland, D. (1985). *Animal Behaviour*. Longman Scientific Technical.

[Rosenblatt, 1962] Rosenblatt, F. (1962). *Principles of Neurodynamics*. Spartan Press.

[Sinclair, 1981] Sinclair, J. D. (1981). *The Rest Principle: A Neurophysiological Theory of Behaviour*. Erlbaum, Hillsdale, NJ.

[Sutton, 1984] Sutton, R. S. (1984). *Temporal Credit Assignment in Reinforcement Learning*. University Microfilms International.

[Tyrrell, 1993] Tyrrell, T. (1993). *Computational Mechanisms for Action Selection*. PhD thesis, University of Edinburgh.

Appendix : Parameters Used

Sensory conditions

Code	Number	Location of detected objects
UL-UR	0	Above on left & right
UL	1	Above on left side
UR	2	Above on right side
C	3	Contact, nothing visible
C-FB-FT	4	Far top and far bottom, also contact
C-NB-FT	5	Near bottom, far top, also contact
C-NB-NT	6	Near bottom and near top, also contact
C-FB-NT	7	Far bottom and near top, also contact
C-FT	8	Far top, also contact
C-NT	9	Near top, also contact
C-NB	10	Near bottom, also contact
C-FB	11	Far bottom, also contact
FB-NT	12	Far bottom and near top
NB-FT	13	Near bottom and far top
NB-NT	14	Near bottom and top
FB-FT	15	Far bottom and top
FB	16	Far bottom
FT	17	Far top
NT	18	Near top
NB	19	Near bottom
MS	20	Motor strain

Behaviours

Name	Priority	Effect
NULL	-1	Do nothing — no behaviour
ADVANCE	0	Approach object
WIGGLE	1	Back 10cm, wiggle back and forth
AVOID	2	Avoid object — back 15cm, turn 90°
AFTERPAT	3	Back, turn 180°, pause

Sensory neuron parameters		
S-DEF-THRESHOLD	0.5	Default threshold
S-DEF-I-UP-GAIN	9.0	Up gain for input register
S-DEF-I-DN-GAIN	9.0	Down gain for input register
S-DEF-A-UP-GAIN	9.0	Up gain for activation register
S-DEF-A-DN-GAIN	4.0	Down gain for activation register
S-REFR-TH	1.5	Refractory threshold default
S-MAX-ON-TIME	0.5	Adaptation time default
S-RECOVERY	0.5	Recovery time default

Releaser neuron parameters		
R-DEF-THRESHOLD	0.5	Default threshold
R-DEF-I-UP-GAIN	9.0	Up gain for input register
R-DEF-I-DN-GAIN	9.0	Down gain for input register
R-DEF-A-UP-GAIN	9.0	Up gain for activation register
R-DEF-A-DN-GAIN	4.0	Down gain for activation register

Behaviour neuron parameters		
B-DEF-THRESHOLD	0.5	Default threshold
B-DEF-I-UP-GAIN	9.0	Up gain for input register
B-DEF-I-DN-GAIN	9.0	Down gain for input register
B-DEF-A-UP-GAIN	9.0	Up gain for activation register
B-DEF-A-DN-GAIN	4.0	Down gain for activation register
B-REFR-THRESHOLD	1.5	High threshold for refractory period
B-REFR-TIME	4.0	Refractory period (seconds)
B-MAX-ON-TIME	2.0	Maximum on time for B neuron (seconds)
B-REFR-TH	1.5	Refractory threshold default
B-MAX-ON-TIME	2.0	Adaptation time default
B-RECOVERY	4.0	Recovery time default

Synaptic parameters		
DEF-SR-K1	1.5	Weight increase integration constant
DEF-SR-K2	3.0	Weight decrease integration constant
DEF-RB-POS-WT	0.8	Default positive weight for R-B connections
DEF-RB-NEG-WT	-0.01	Default negative weight for B-B connections
DEF-RB-LGNEG-WT	-1.5	Default strong inhibitory weight for RB, hence BB
RB-FBCK-WEIGHT	0.5	R-B feedback weight
SR-DEF-WEIGHT	0.01	Default weight for S-R connections