An Ethological Model for Implementation in Mobile Robots

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

FAX: 031-650-6899
Tel: 031-650-3080

Abstract

This paper describes a neuroethological model of learning and motivation which accounts for many of the behavioral phenomena observed in animals. Unusual predictions from the model have been tested and shown to be demonstrable in laboratory Siamese fighting fish. In addition, the model is sufficiently mathematically well-defined to be 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 a simplified version of the model which was programmed into the robot, a thought experiment designed to show the main features of the model, and the preliminary robot experiments carried out.

Putting ethological models of animal behavior onto robots is interesting for both robotics and ethological research: the study of robot autonomy can be enhanced through an understanding of complex and realistic models of animal autonomy; and ethological research should benefit from a supply of guaranteed 'naive' agents on which rigorous testing of such models is tractable.

Keywords: learning, mobile robot, behavioral model

Running title: Ethological Robot Learning

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. One example is the Adaptive Resonance Theory (ART) set of tightly mathematically specified models of human cognition put forward by Grossberg and collected in Carpenter and Grossberg (1991). The ART1 model has recently been successfully used as a robot controller by Baloch and Waxman (1991). Halperin's Neuro-Connector model, described in this paper, is another mathematically well-specified model of animal behavior which has potential for use as a robot controller. The Neuro-Connector model is proposed as a means of choosing and refining which of a multiplicity of recognizable sensory states should trigger which actions, not as a means of learning perception or of developing behavioral components. It is assumed that sufficient sensor pre-processing exists to produce a form of 'feature detector', and that code enabling the robot to perform the desired behaviors is working. This model is only really suitable for use as a robot controller for those tasks requiring a sensor-rich robot to make complex sensory discriminations of a type difficult for a human designer to encode, *i.e.*, for robots needing to work in complex 'animat'-type environments (Wilson, 1990). Robots performing simpler tasks should be controlled by more computationally efficient means.

Halperin's Neuro-Connector model of learning and motivation is based on an unusual interpretation of Hebb's (1949) insight that synapses should strengthen when the connected neurons fire together , namely that it is only the *ends* of firing which are significant. This is combined with Sinclair's (1981) 'Rest Principle' 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 learning behaviors such as classical conditioning, operant conditioning, and priming of responses. These features are discussed in more detail in Halperin's (1990) PhD thesis. When the model was completed further analysis showed that it made some unexpected predictions about behavior, including specifying the conditions under which backward conditioning should occur. 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.

In this paper we describe a simplified version of the Neuro-Connector model and the way in which it can be initialized and used on a robot. 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 fairly complex behavioral phenomena observable in animals. We are currently investigating the capabilities of a less simplified version of the Neuro-Connector model.

1 The Neuro-Connector Net

The Neuro-Connector 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 many commonly used neural network simulations (*e.g.* the Perceptron described by 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 can be considered to use 'feature detectors'.

It should be noted that the model described in Halperin's thesis contains extra features and enhancements and which allow explanation of other biological phenomena not discussed here, such as classical (stimulus-stimulus) conditioning and hormonal and motivational influences.

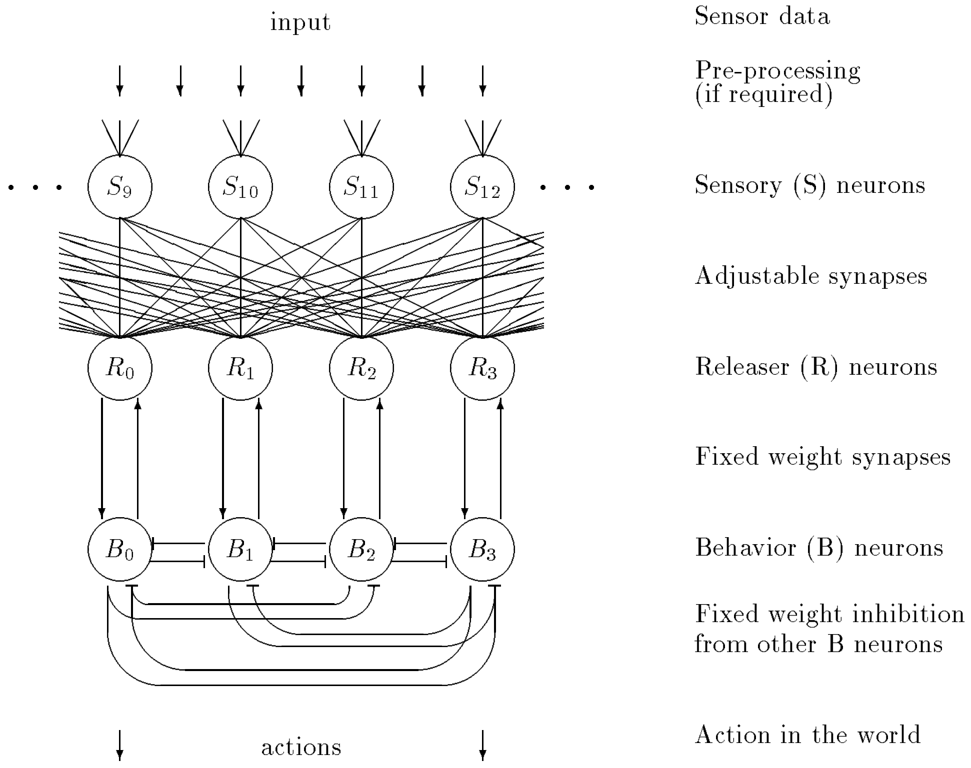

Figure 1: Part of a Neuro-Connector Net

1.1 Topology and Operation of the 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 expected pattern of firing is that at any time various S neurons will be on, reflecting the current sensory state of the agent (both internal and external sensors). Various Rs will be on due to their getting sufficient input from the Ss. One R will be on due to its B being on, whether it has much S input or not. One B will be on. The Ss and Rs currently active will keep changing, whereas B firing will tend to be more persistent due to the feedback loop with the R and the inhibition from the current B to the others. The current B will stop firing (and the behavior it produces finish) when either another B neuron comes on and inhibits the current B, or the current B adapts or times out.

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, thus allowing the net to learn.

When a releaser neuron accumulates enough input to fire, its behavior neuron is stimulated. The B neuron will only fire if the stimulation it receives is sufficient to overcome inhibition from the B neuron currently active. B neurons inhibit each other such 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

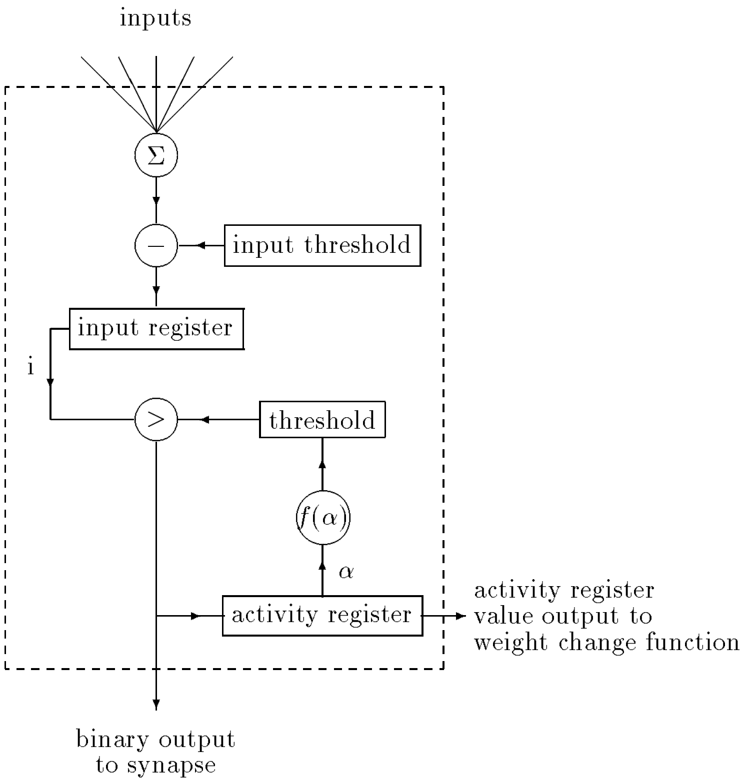

Figure 2: Inside a Neuron

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' (1986b) terminology or a 'characteristic' in the terminology of McFarland and Sibley (1974) . 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 ...

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.

1.2 Components of a Neuro-Connector Net

The components of the Neuro-Connector model comprise 'neurons' and 'synapses', each containing memory and some idea of the passage of real time. More mathematical descriptions of these components are given in Hallam, Hallam and Hayes (1994).

1.2.1 Neurons

The three types of neuron mentioned in the previous section all have the same internal structure and operation, illustrated in Figure 2.

Neuronal input is thresholded and passed to the input register. This register contains a single value which can be considered to be a memory of recent input, averaged over time but with a strong weighting towards the most recent input values. This decaying average filter (otherwise known as a relative activity smoothing unit, a leaky integrator process, or a first-order lag process) operates by exponentially decaying the current input register value towards the new input value.

The new input register value is compared with a threshold (which may vary dynamically, see below) and the neural output is then calculated. Current versions of the model all use binary neurons, so the output is 1 if the input register value exceeds threshold and 0 otherwise. B neurons have a further comparison stage forcing B outputs to zero until only one B is active at any time.

The activity register contains a value which can be considered to be a memory of the recent activity of the neuron, and a decaying average filter which exponentially decays the current activity register value towards the current output value as described above. The new activity register value is passed to the weight change functions detailed in Section 1.3 and also to the threshold change function. Since B neurons are not involved in weight changes, they do not have an activity register.

The output of each neuron is determined not only by its input but also by the behavior of its threshold. If the threshold does not change then the neuron gives a constant output for constant input, after a small delay for the input register to get above threshold. This is the case for the R neurons and some Ss. If the threshold does change then sufficient constant input will cause the neuron to be on until the threshold exceeds the input register value, at which point the neuron will turn off and will stay off until the threshold decays again. Varying the threshold change function or parameters can result in neurons with different responses to standard inputs and thus enables modeling of different biological neuron types.

1.2.2 Synapses

Three types of synapse exist connecting the different neuron types. S–R synapses have plastic or adjustable excitatory weights and are responsible for the learning in the model. These are discussed in greater detail below. R–B and B–R synapses have fixed excitatory weights and are responsible for the feedback loop which causes behaviors to be persistent. B–B synapses have fixed inhibitory weights and are responsible for ensuring that only one behavior is active at any time. The assumption that the set of behaviors is mutually exclusive is important in order that actuators do not receive concurrent contradictory commands. Animal behavior can also be classified into a set of mutually exclusive 'activities', as argued in McFarland and Sibley (1974).

S–R synapses contain — in addition to a record of their current weight and of which is their pre- and post- synaptic neuron — a time parameter t_{exp}. This time parameter is the time expected to elapse between the S neuron finishing firing due to the stimulus disappearing and the R neuron finishing firing due to the successful completion of the behavior released by that R. The synapse also keeps track of the actual difference in firing, t_{obs}, between its S and R neurons by recording the current activity register value in its R neuron and the previous activity register value (t_{exp} seconds ago) of its S neuron. The expected time t_{exp} is set such that when a behavior is successful there should be an 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, and the correlation is not as good. Correlation is also bad if either neuron fails to fire. Weight adjustment occurs dependent upon the degree and direction of this correlation.

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. All weights can be fixed after a setting-up period if it is desirable for the 'learning phase' to finish, but normally the agent learns continuously.

1.3 Weight Adjustment

Only S–R synapse weights are plastic, adjusting according to temporal correlation in firing. Since temporal correlation is measured by an exponential decay process, only the *ends* of firing are significant. The temporal correlation does not have to be perfect as there is a window within which strengthening of some degree will occur, although obviously perfect correlation causes the greatest strengthening. This strengthening window is centered about a time t_{exp} after the end of S neuron firing.

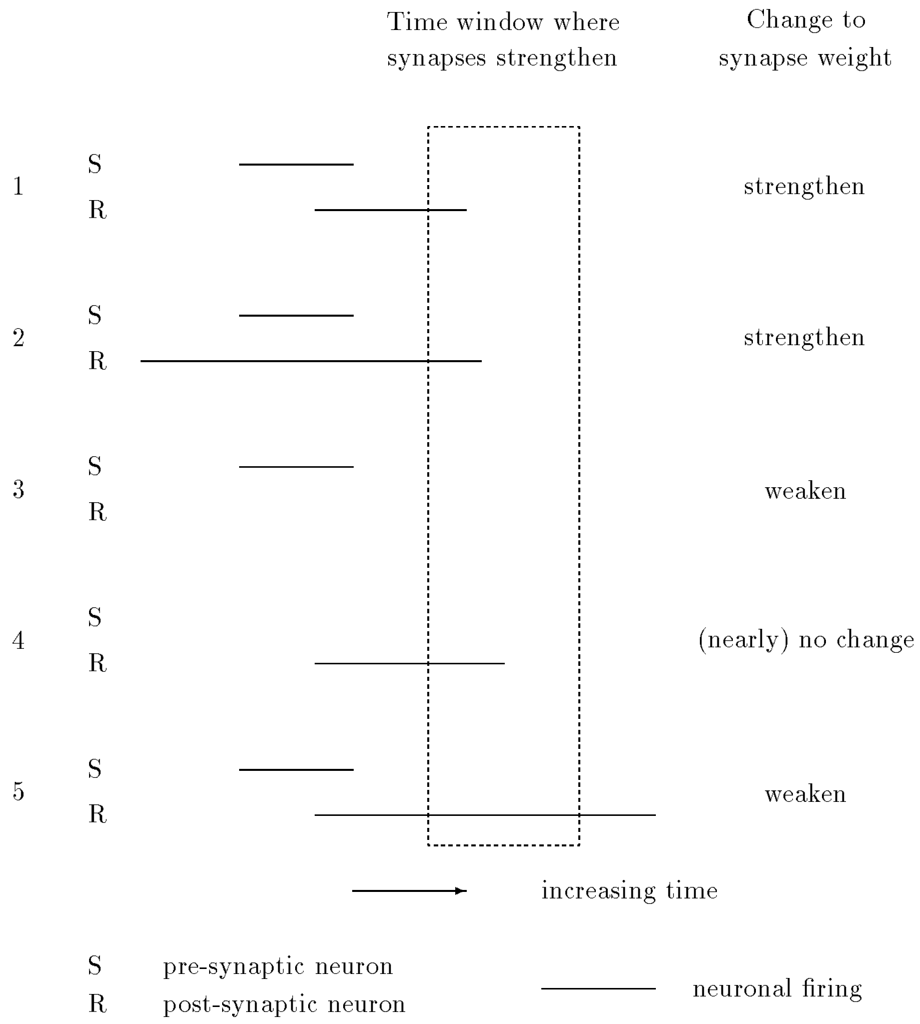

Figure 3: Qualitative Weight Adjustments (taken from [Hal90b])

The required behavior, illustrated in Figure 3, is as follows:

1. If the ends of firing in S and R are well correlated then the weight increases.
2. This is true whatever the initial correlation in firing.
3. If the S goes on without the R the weight change should be negative, since it this would indicate that the current situation is unrelated to this R's behavior.
4. If the R goes on without the S the weight should not change. This is so that the same behavior, *e.g.* 'flee', can be triggered by different stimuli.
5. If both neurons go on but are not well correlated then the weight decreases.

These rules are combined into a continuous control surface which gives a value for the coactivation of the S and R neurons, as illustrated in Figure 4. This surface was chosen because it implements the criteria given above; there are probably other surfaces which would also work.

The calculation of the weight change is detailed in Hallam *et al.* (1994). Figure 5 illustrates this calculation for a simulated synapse where the S neuron was on for a shorter time than the R but both finished approximately t_{exp} apart. In outline it proceeds as follows:

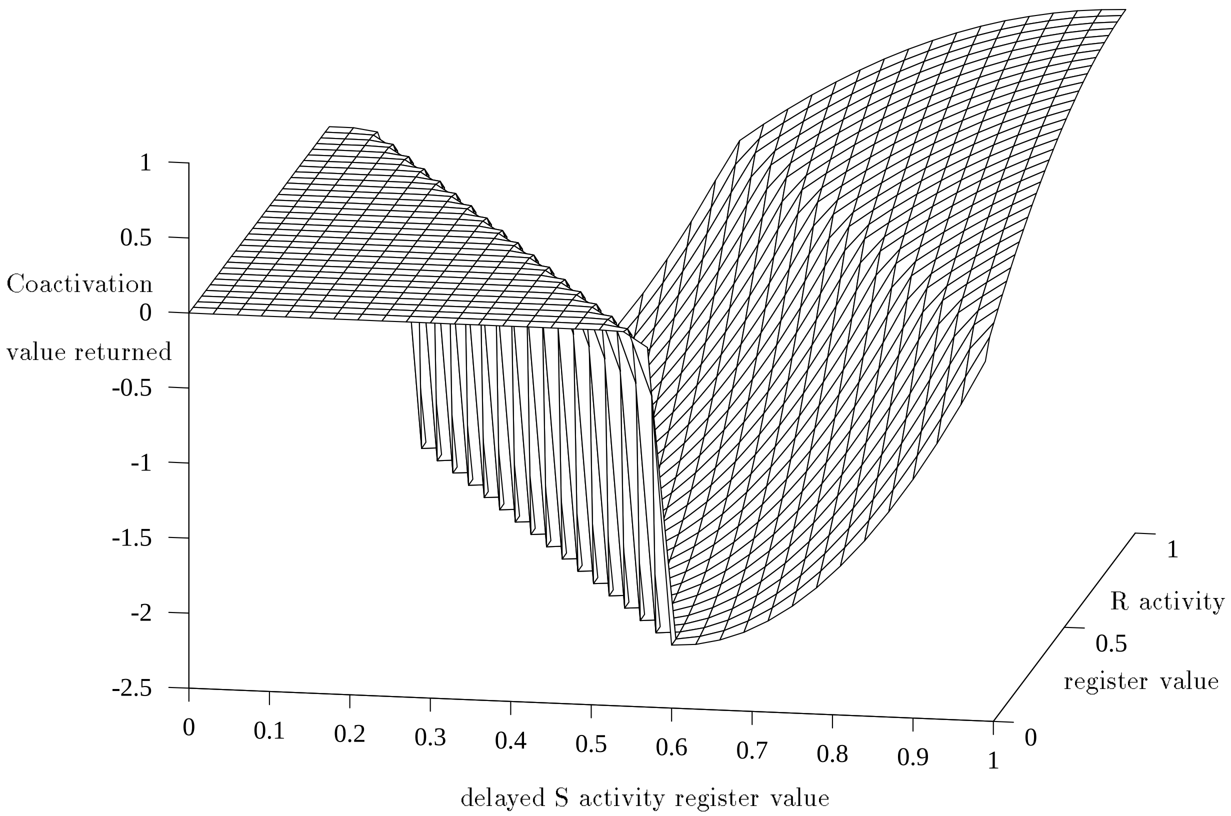

Figure 4: The Coactivation Returned for Combinations of S and R Activations

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 (Figure 5a) and a coactivation value (Figure 5b) is returned from the surface in Figure 4. A command register contains a decaying average filter whose value can be considered to contain a memory of this coactivation, *i.e.* of the correlation in activation between the S and the R neuron. This command register value (Figure 5c) is multiplied by a consolidation function value (Figure 5c) which is zero when either neural activity register has a value over about 0.1 and is one most of the rest of the time. The consolidation function implements Sinclair's (1981) 'Rest Principle' by ensuring that no weight change takes place unless both neurons have been off for a little while.

Weights are changed (Figure 5d) by a decaying average process with target values of 0 or 1 and a gain which is the product of the command register value and the consolidation value. The direction of the weight change depends on the sign of the command register value when the consolidation function starts returning non-zero values. The size of the weight change depends on the value in the command register at that same point and on the rate of decay of the command register value.

2 A Thought Experiment: Using the Neuro-Connector Net to Detect and Respond to People

The following discussion describes using a Neuro-Connector net for a task requiring sophisticated sensory discrimination: detecting and responding to people. This thought experiment is included in order to explain more easily how the net is initialized, how the releasing stimuli are refined, what designer knowledge is required, and how the behaviors inter-relate.

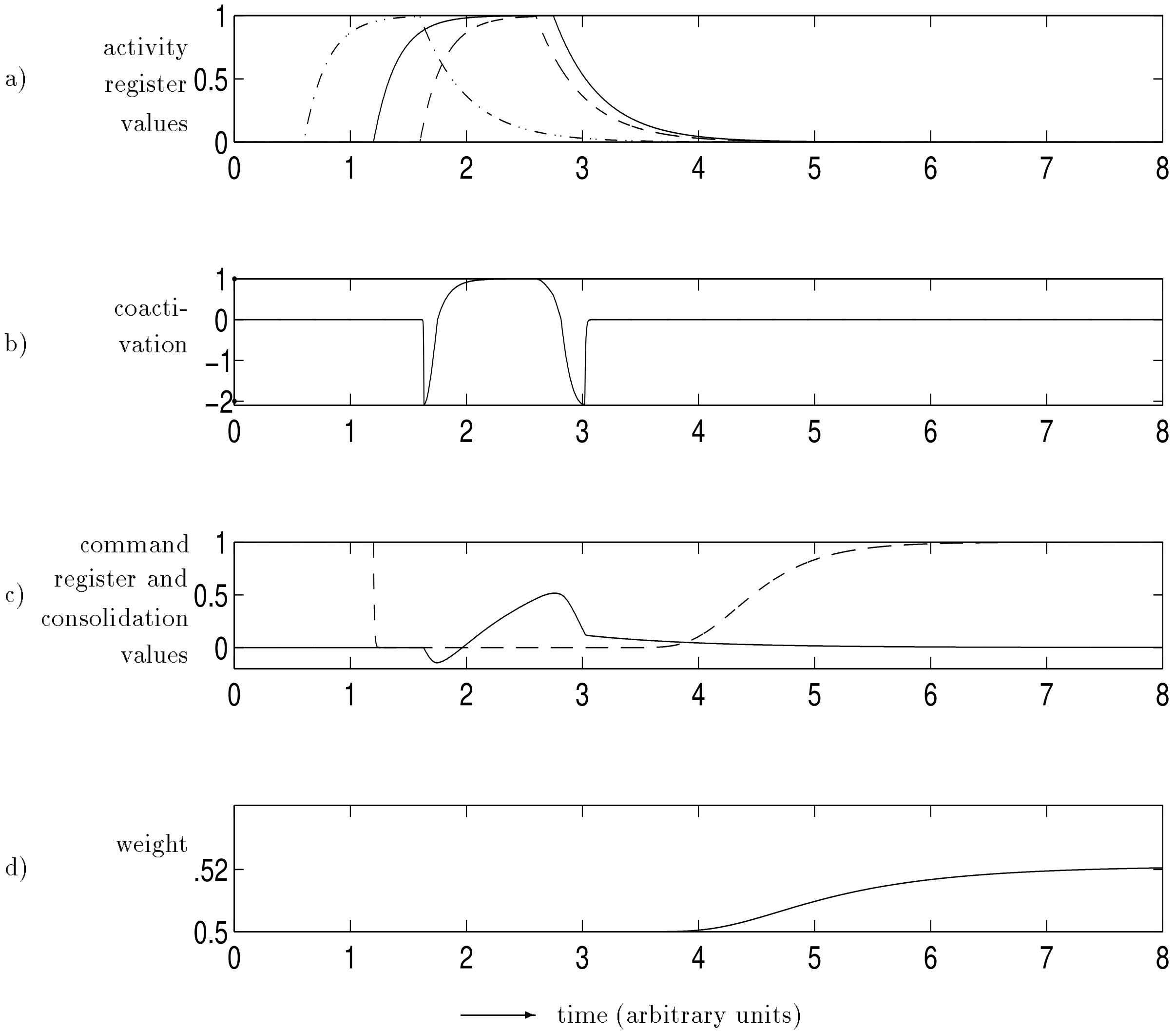

Figure 5: Changes in the Registers

a) Activation Register Values versus Time.
– · – S; —— R; – – – delayed S; $t_{exp} = 1$.

b) Coactivation Value versus Time.

c) Weight Change Rate Constant Factors versus Time.
—— command register value; — — consolidation value returned.

d) Synapse Weight versus Time.

2.1 Background, Task and Environment

Assume that we are working in a sizeable robotics laboratory with large south-facing windows. The environment includes fixed furniture, movable furniture, computer equipment, other mobile robots and robot arms as well as people. In preliminary experiments we equipped our robot with heat sensors and movement sensors and expected people to show up as moving heat sources. Results showed that we missed people who were working at terminals because they were too still, unless we increased movement sensitivity to the extent that we falsely detected patches of sunlight as people. We then chose to try the Neuro-Connector approach, and fitted our robot with other sensors: microphones and sound processing, light-dependent resistors, a bank of touch sensors and anything else we had handy. Since this is a fictional scenario, we can assume that we made sensory neurons for every reasonable sensor state without running into processing problems!

On top of a basic navigational competence, we gave the robot three person-specific behaviors as illustrated in Figure 6: 'stop beside person', 'talk to person', and 'robot response'. The first shows the discriminatory competence that we want: the ability to detect people. The second is a test behavior which is designed to disambiguate actual people from other situations giving similar sensory readings. For our robot a suitable test might be for the robot to ask for its touch sensor array to be stroked. The third behavior is responsible for signaling to the robot whether the person test was successful, it might also cause the robot to say "Thank you!". We will assume that all (or nearly all) people using the lab will respond correctly when the robot addresses them. We will also assume that the robot is normally moving, actively seeking people to approach.

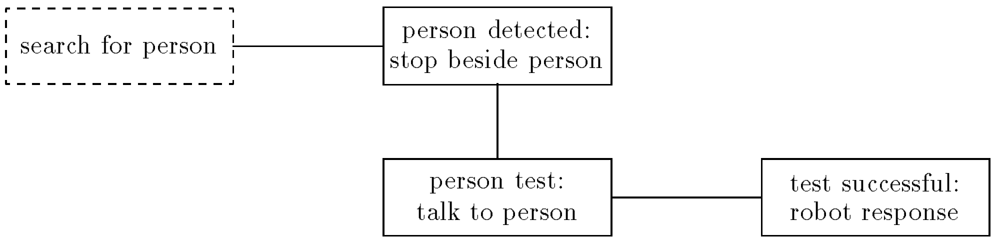

Figure 6: Behaviors for a Fictional Task
Dashed boxes represent behaviors which have many components.

2.2 Initialization

Various parts of a Neuro-Connector net require initialization specific to the robot task: behavioral priorities, weights, expected times, and neuronal adaptation times. This section considers the initialization required for the task discussed above.

2.2.1 Behavioral Priorities

Many subtasks are implemented as sequences of behaviors like that illustrated in Figure 6, where the ordering of the behaviors for that subtask is known but behaviors from other sequences might interleave in places. Within such a sequence, behaviors are prioritized such that later behaviors always inhibit earlier ones. In our example, the 'robot response' behavior inhibits both the 'stop beside person' behavior and the 'talk' behavior. 'Talk' inhibits 'stop beside person'. The premature performance of later behaviors is precluded by ensuring that the S neurons for these later behaviors include conditions which only become true when the necessary earlier behaviors have finished successfully. Therefore S neurons which we predict might have suitable releasing connections with 'talk' contain not only conditions which might be used to detect people but also the requirement that the robot has deliberately stopped. Priorities between subtask sequences are less clear and depend upon the relative importance of the subtasks. Essential reflex behaviors will obviously have highest priority.

2.2.2 Weights

The net can be initialized such that there are high-weight connections between each behavior-releasing R neuron and sensory neurons that respond to situations in which we expect the behavior to be appropriate, and low-weight connections everywhere else. We do not have to know the best S neurons to choose; one of the reasons for investigating this model is that it can refine its 'releasing' stimuli. We do have to choose our releasing S neurons such that the behavior will normally operate correctly.

For this example, we will again consider that most moving heat sources are people, and will set up the following three high-weight connections:

- from the sensory neuron for 'moving heat source and moving self' (for ease of reference, let's call this S_1) to the 'stop beside person' behavior;
- from the S for 'heat source close nearby and stationary self' (S_2) to the 'talk' behavior;
- and from the S for stroking (S_3) to the 'robot response' behavior.

2.2.3 Expected Times and Adaptation Times

When the robot detects a moving heat source it will move next to it and then stop. 'Moving self' becomes false so S_1 goes off as the robot stops. We can therefore set t_{exp} for any synapse connecting the 'stop beside person' behavior to any S neuron containing a requirement for 'moving self' to be zero. Other S neurons will not be connected to this behavior, as we have assumed that the robot is moving when it is searching for people.

The robot will ask the presumed person to stroke it, and give instructions for the uninitiated. These instructions will continue until the robot is stroked, although we will set the behavior to time out eventually. Since we cannot assume that the person will move away as soon as the test is over, the conditions causing S_2 to fire will continue to be true after the talking is finished. This means that either we need a negative t_{exp} or S_2 must adapt out. Since we don't know how long the person will stay admiring our robot and therefore cannot ensure that even a negative t_{exp} will remain constant, we'll get S_2 to adapt out. In fact, since we want the robot to discover a more accurate and reliable person-detector than S_2, we will make all '<possible person detector> and stationary self' sensory neurons adapt out with the same timing. Then we can set t_{exp} for any synapse connecting the 'talk' behavior to an S neuron containing the 'stationary self' condition to be the time taken for the robot to make its initial request minus S_2's adaptation time. We will have set the adaptation time ourselves; the time taken for the robot to make its request can be measured by a stopwatch; therefore we can calculate t_{exp}.

The 'robot response' behavior must be set to adapt out such that its R neuron finishes firing t_{exp} after S_3. The size of t_{exp} and the length of the adaptation time don't matter as long as this correlation exists. This causes the 'robot response' behavior to be self-reinforcing as described below.

Assuming that both B and S neurons adapt out (rather than being interrupted or the sensory state changing), then B adaptation time, S adaptation time, and t_{obs} are related in the following way:

$$t_{obs} = B(dur) - S(dur) + B(delay) \tag{1}$$

where $B(dur)$ is the B neuron adaptation time, $S(dur)$ is the S neuron adaptation time, and $B(delay)$ is the time taken for B to fire after S starts firing. $B(delay)$ can be measured or else worked out from neuronal gains; $B(dur)$ and $S(dur)$ are set by ourselves.

End-of-sequence behaviors such as the 'robot response' behavior of Figure 6 need to operate correctly in order to confirm the whole sequence of behaviors. This is achieved by setting $B(dur)$ sufficiently short that the behavior is not likely to be interrupted, and setting $S(dur)$ and t_{exp} such that $t_{exp} = t_{obs}$ fulfills the equation above for any synapse which might become a releasing connection. In this way end-of-sequence behaviors become self-reinforcing, in that their releasing connections will always strengthen.

Other behaviors should not be self-reinforcing but rely on the next behavior interrupting them at about the right time if their connections are to strengthen. This reliance is achieved by increasing $B(dur)$ such that the t_{obs} which occurs when the behavior is not interrupted is longer than t_{exp} in connected synapses.

2.3 In Operation

Since all the people in our theoretical lab behave correctly, then when the robot has talked to a real person it will get stroked almost straight away. The 'talk' behavior therefore gets switched off by the 'robot response' behavior and the 'talk' R neuron stops firing due to having no input (remember, we got the S neurons to adapt out). Any S neuron which stopped firing at about the correct time t_{exp} compared with this R will have its synapse with this R strengthened. S neurons which respond to something other than 'person close by' will only rarely finish at this correct time, so their synapses with this R will generally be weakened.

If our robot had incorrectly identified a patch of sunlight or a fellow robot as a person, then it would not get stroked and the 'talk' behavior would continue longer than expected. The timing relationship between R-'talk' and the sensory neurons which had fired in response to the sunlight or robot would therefore be rather different from that expected and these synapses would weaken (rule 5 of Figure 3). Any S neuron which responds to both robots and people will have its synapses with R-'talk' weakened when a robot is approached and strengthened when a person is approached. In this way S neurons responding to general stimuli remain irrelevant for triggering the 'talk' behavior. Note that, if we had no sensory neurons capable of specifically detecting people, this behavior would disappear due to all S neurons responding to other things having their connections with this behavior weakened each time the other things were tested. This would be a major design flaw. S neurons which only fire when a person is close by will have their synapses with R-'talk' strengthened. Even if these S neurons were originally not considered to be likely candidates for 'person detector' their firing will become sufficient to release that behavior.

After many encounters, we can examine the net and record which sensor combinations are considered reliable and specific person indicators by seeing which S neurons have developed high-weight connections with the R-'stop beside person' and R-'talk'. Note that we should not assume that these S neurons are equivalent or that any of them is a general 'person detector'; there may be specific S for different types of people or even for different individuals, depending on the discriminatory ability of our sensors. It is also possible that our robot would recognize some people and not others. This would certainly be the case if some people never responded to the robot's request for stroking, since they would then be categorized as non-people and the robot would learn not to approach them.

If it was known that the releasing stimuli would not change, then low-weight synapses could be removed once the robot had developed the correct behavior. This would decrease the processing time required considerably, but would also remove the robot's ability to learn. If the releasing stimuli are expected to change in known ways (*e.g.*, the only change will be in light levels) then less radical pruning could be performed.

2.4 Working with Insufficient Knowledge

Although sufficient knowledge of the task is normally available that we can craft behaviors which should be adequate for achieving that task, often the other knowledge that we need is not so readily available. We are assuming here that we are talking about tasks which have too much variation to be achieved by simply sequencing the behaviors, and need too fine a discriminatory ability for a human designer to be able to specify the conditions under which each behavior should occur. We are still assuming that the robot has enormous numbers of sensory neurons responding to different sensory states, as this will compensate for any lack of information about the environment or about what sensor readings mean.

In the above discussion we used our knowledge of distinguishing features of people (that they move and give off heat) to provide our robot with default responses that worked well enough that the robot was normally doing the right thing. The net then redefined which were the releasing stimuli until only specific ones were used. What if we have no such knowledge, and can't easily find reasonable releasing stimuli? Without knowledge of releasing stimuli, how can we set t_{exp} or adaptation times?

As long as we can watch the robot operating in its environment and can force behaviors directly (perhaps using a radio modem), then we can get the net to initialize itself through the normal operation of its weight change algorithm.

The task and behaviors should be analyzed and behavioral sequences identified where behaviors are expected to follow one another and where subsequent behaviors are dependent upon the successful operation of previous behaviors. Behavioral priorities should be allocated according to position within the sequence, and adaptation times should be set to be long. End-of-sequence behaviors should be timed and set to be self-reinforcing as described above. Critical behaviors should be given fixed-weight synapses and high priority in order to provide essential reflexes. Since in many cases the most suitable t_{exp} is not known, multiple synapses with different t_{exp} can be used between pairs of neurons.

All synapses should be initialized with low weights and the net run as normal except that either R or B neurons are given extra input links which can be stimulated by a human trainer. The robot should be run through its task with the correct behaviors being triggered at the correct times by the trainer. The trainer should ensure that even rare events happen if they require the robot to learn to respond. R and B neurons will be firing at the right times forced by the trainer; the S neurons will be firing in response to the state of the environment. S neurons which have the correct time relationship to any R will have their synapses with that R strengthened and will come to be able to trigger that behavior. After four test runs (according to the robot results reported in Section 5.6) the robot should be able to operate by itself.

2.5 Responses to Change

Most changes in the environment are not expected to affect the behavior of our robot since it has so many sensors and sensory neurons. There will be many S neurons with releasing connections to each behavior, and most behaviors will be able to be released by several different sensory modalities. Releasing connections for a behavior which needs to recognize people might connect with S neurons representing 'I can see a face', 'Here is a moving heat source', and 'I can hear voices'. Therefore although a change such as the lights being switched off may cause visual sensors to cease being useful, the information from the infra-reds can still be used to trigger the behavior. When the lights are switched back on then the light-dependent stimuli will become significant again, since R firing without S does not weaken connections (rule 4 of Figure 3). The same argument applies if a sensor fails — other releasing stimuli will be used instead and the robot's behavior may degrade slightly but should not fail completely.

Slower changes in the environment *e.g.* sunset may cause other S neurons to develop releasing connections as an effect of normal learning.

3 Characteristics of the Model in Operation

The rate of learning is 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 some of the MIT robots (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.

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 (1985), where the dominant behavior is interrupted periodically by a short burst of a secondary behavior. 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 sufficiently early. 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, and consequently the agent does not get stuck retrying a behavior that is for some reason impossible.

Where the B neuron adaptation time is short and the R neuron always 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.

4 Some Biological Learning Phenomena

This section outlines a few biological learning phenomena which the model can and cannot explain. It assumes a familiarity with basic learning terminology.

Since the only criterion for adjusting synapse weights in this model 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 in the artificial experimental conditions of the laboratory. **Secondary conditioning**, where a stimulus is trained to a conditioned stimulus instead of an unconditioned one, occurs as readily as primary S–R conditioning. **Blocking**, however, cannot be shown — where two stimuli are both shown in the right time relationship with a response, then the model will learn releasing connections with both of them whether or not one of the stimuli was already a releasing stimulus.

S–S conditioned responses (more commonly known as classical or Pavlovian conditioning) can also be accounted for by this model if we include some of the extensions to the model described in Halperin's (1990) thesis. Obviously we need synapses between S neurons; these are considered to follow the same learning rules as detailed for S–R synapses. We also need to consider the time course of neural responses. In order to explain S–S conditioning we have to assume that many sensory neurons adapt out early and stay off for a long time, *i.e.* that they respond only to novel stimuli, to changes in state, to the beginning or onset of an event. S neurons responding to stimuli which appear together therefore turn off together, thus developing high-weight connections (t_{exp} is assumed to be zero for S–S synapses). This idea is developed in more detail in Chapter 7 of Halperin's (1990) PhD thesis.

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). This form of long-delay backward conditioning is expected to be transient, since using the newly developed releasing stimulus causes neuronal firing which is badly correlated and so weakens the synapse again. The requirement for showing the neutral stimulus with correct timing and the transience of the conditioning produced perhaps explains why the existence of backward conditioning is debated (Spetch *et al.*, 1981).

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).

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.

5 Robot Experiments

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. Preliminary tests ensured that synapse weights changed as specified and that postponed conditioning occured. A mobile experiment investigated the refinement of releasing stimuli and backward chaining of failure. All were successful.

5.1 The Robot

'Ben Hope' is an autonomous transputer-based mobile robot (see Figure 7) 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.

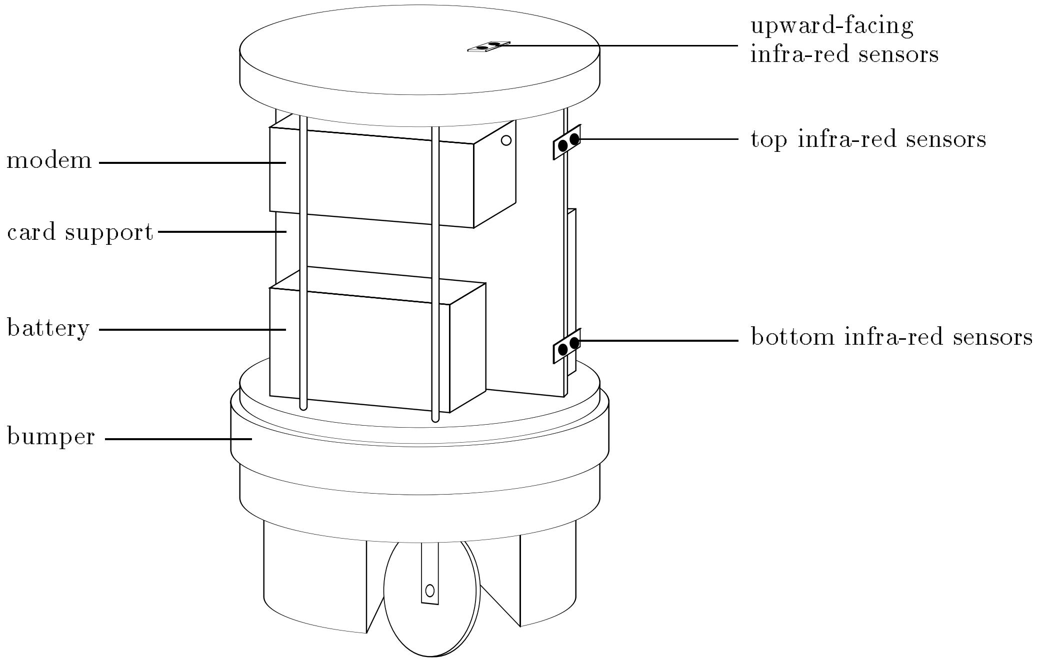

Figure 7: The Mobile Robot Ben Hope

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 7. 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.

5.2 The Model Used

Twenty-one sensory neurons, four releaser neurons and four behavior neurons were set up and the S and R neurons fully connected, although only six synapses were recorded during these experiments. 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.

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.

Name	Prio-rity	Actions Involved	Sensor Conditions
ADVANCE	0	Approach object	Far top and/or far bottom
WIGGLE	1	Back 8cm, wiggle back and forth	Near bottom and near top else near bottom and far top, with or without bumper
AVOID	2	Avoid object — back 15cm, turn 90°	Bumper only
AFTERPAT	3	Back, turn 180°, pause	Upwards-pointing left and right infra-red sensors

Table 1: The Behaviors Used and their Releasing Stimuli

5.3 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.

5.4 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 (0.8) and one to a low weight (0.01). Both S_0 and S_1 were connected to the releaser for the behavior AFTERPAT, which causes the robot to back up and turn through 180°. This behavior 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}). All neurons were stimulated simultaneously, the S neurons for a constant 0.5 seconds, sufficiently long that the activity register was quite high. This variation in firing relationship is 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. The net was reset between each test to provide the same starting conditions for each adaptation time.

When t_{obs} is small, this corresponds to the situation in which 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.

Adaptation times in the range of 0.7 to 3.9 seconds were tested, each twice. After sufficient

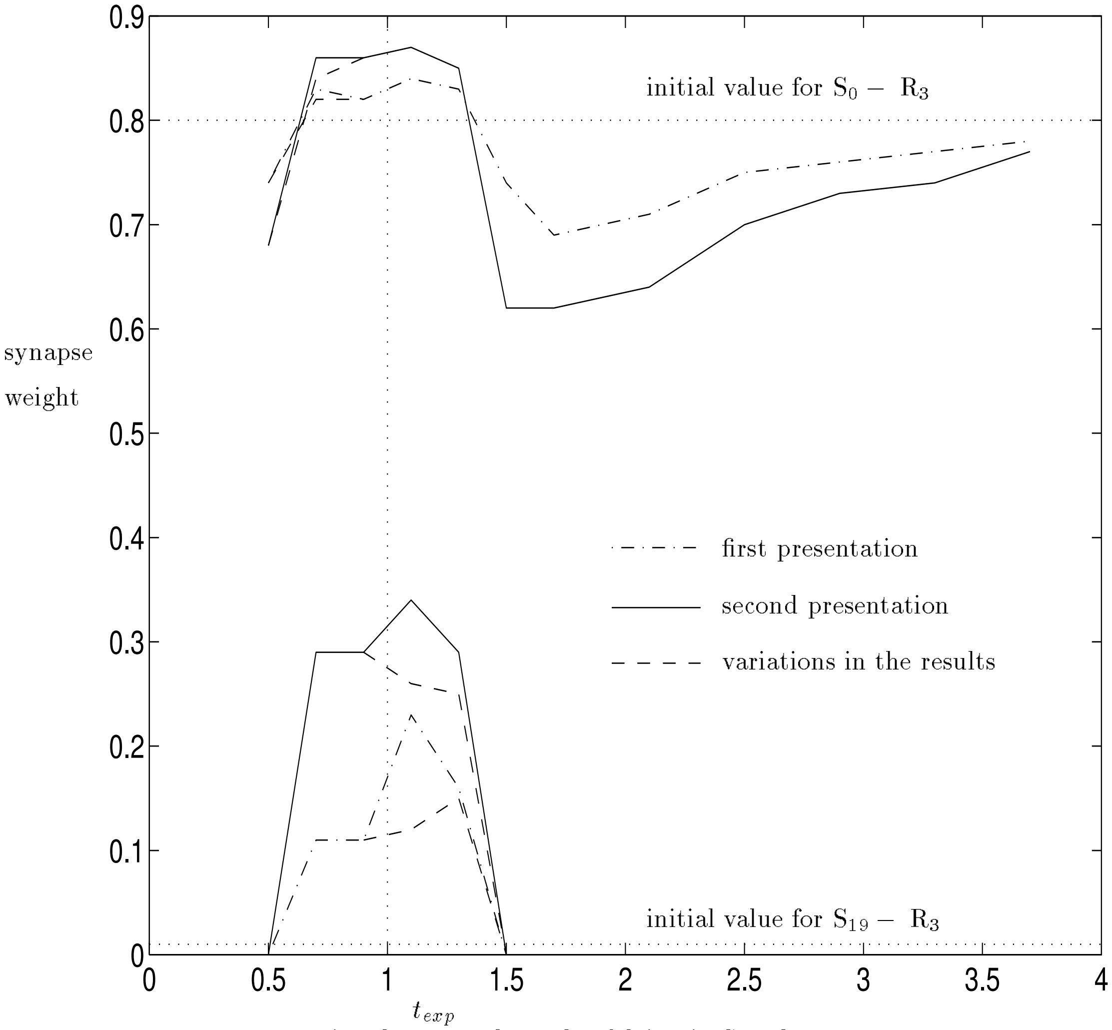

Figure 8: Synapse Weights Vary with Timing

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.

The results are given in Figure 8, 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.

5.5 Postponed Conditioning

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 4). This experiment was designed to show that postponed conditioning does indeed occur as predicted by the model.

The adaptation time in the R-WIGGLE neuron was at its default of 2 seconds, which meant that the wiggle behavior continued sufficiently long 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.

5.6 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 by using a simulated wall which the experimenter could place near to the robot. 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, so flaws in experimental design do not waste carefully unstimulated animals.

5.7 Discussion of Experimental Results

The results shown in Figure 8 demonstrate 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 quickly than behaviors which only after-discharge briefly. In a robot, the unlearning characteristics of each behavior can therefore be adjusted by changing just one parameter — the adaptation time.

The dramatic increase in the low (S_1 – R-AFTERPAT) synapse weight shows that previously insignificant stimuli can become important quite quickly. In the postponed conditioning experiment of Section 5.5 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 operant 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. 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 described in Section 5.5 showed that postponed conditioning can be demonstrated in the robot. This non-aversive, transient, form of backward 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 biological mechanisms.

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.

6 General Discussion

The Neuro-Connector model is well suited for use as a robot controller for tasks in which the stimuli which should release particular behaviors are poorly understood or are liable to change, and particularly where sensor information is plentiful. It is not as well suited for tasks which are better understood or where only sparse sensor information is available, where more traditional robot control techniques will suffice.

This model employs an unsupervised connectionist or neural net which uses reinforcement from the environment to control learning. The model learns quickly and continuously. This means that a robot controlled by this model should continue to operate despite changes in circumstances, even where these changes were unforeseen.

Plentiful sensory information is expected to be available and pre-processed. The task is expected to be modular and attainable with the behaviors available. These behaviors are expected to exist already, to be prioritized, and to work correctly under ideal conditions. They may use sensor information directly.

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. Where timings are uncertain, multiple synapses can be used between each S–R pair. Lack of knowledge of the most specific releasing stimuli is not a problem if a 'good enough' releasing stimulus can be found such that the robot will perform correctly the first few times. 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. In particular, a Neuro-Connector net can be trained by hand *via* direct stimulation of the B neurons.

This is a great strength of the model in situations in which robots are sensor-rich and/or are expected to operate in environments which are poorly understood.

In the implementation used here, behavioral sequencing was achieved by using dissimilar B – B inhibitory weights, such that later behaviors could always interrupt and inhibit earlier 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 single chains of behaviors, deciding priorities between different chains or systems is not so easy. For many robot applications hand-prioritizing of behaviors is quite sufficient (and is often done, *e.g.* (Brooks, 1986a)), but it will not work where relationships between behaviors are complex. Adjustable priorities based on sensor readings will be necessary for the simulation of animal behavior, since the relative importance of *e.g.* obtaining water or food is not intrinsic but depends upon the internal state of the animal. An investigation of other principled methods of behavior switching is currently in progress.

The incorporation of real time, not just iteration-number time, is one of the strengths of this model when considered for real robot applications. However, this also causes some problems. For example, a robot with a newly-charged battery moves more quickly and turns more easily than a robot whose power level is low. Thus the same behavior will take different amounts of time depending on the battery charge level. Animals also may chose to do things slowly or quickly. It is not clear how this model will cope with these differences in timing. It is also not clear that all behaviors can be made to have a constant time relationship with their releasing stimuli. It appears that adjusting the width of the time window within which strengthening occurs involves changing the coactivation surface illustrated in Figure 4. This will not be simple.

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 neuro-ethological models are so complex, and second, because these complex ethological systems interact with equally complex environments. With robots, ethologists will be able to test their complex models and 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.

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.

There are still some problems with the model, notably the concept of prioritizing behaviors and difficulties associated with using precise time relationships. In particular, the behavioral sequencing problem must be explored; the sensitivity of learning to parameter settings investigated; and alternatives to hand-setting of t_{exp} 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 biological neuron types in various combinations should also be investigated. These investigations could most usefully be carried out first in computer simulation before transfer onto robots. Some investigation of the effect of varying parameter settings for single synapses has already taken place and is reported in Hallam *et al.* (1994). Some thoughts on improving the computational efficiency of the mathematical aspects of the model are also explored in that paper. A second robot similar to Ben Hope is being built, thus increasing the amount of experimentation time available.

7 Conclusions

The Neuro-Connector model is exciting to roboticists wishing to build robots which exhibit adaptive behavior 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 robot learns for itself the sensory conditions under which each behavior should occur. 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, despite these being unforeseen by the programmer.

Much further work is necessary, both in terms of discovering the advantages and limitations of the model for controlling robots and in investigating the types of behavior which can be achieved. It would also be useful for robot experiments if the model could be made more computationally efficient. Some of this work is already in progress.

Acknowledgements

Bridget Hallam acknowledges receipt of SERC grant number 9130847X. Janet Halperin acknowledges receipt of SERC grant number GR/H/23344. Ben Hope was built under SERC grant number GR/F/5852.3. 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: Bob Fisher, Gillian Hayes, Edward Jones, Ashley Lotto, Lisa Saksida, and Jeremy Wyatt.

References

Baloch, A. A. and Waxman, A. M. (1991). Visual learning, adaptive expectations , and behavioral conditioning of the mobile robot MAVIN. *Neural Networks*, 4(3):271 – 302.

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

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

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

Carpenter, G. A. and Grossberg, S., editors (1991). *Pattern Recognition by Self-Organizing Neural Networks*. MIT Press.

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

Hallam, B. E., Hallam, J. C. T., and Hayes, G. M. (1994). A dynamic net for robot control. In van der Smagt, P., editor, *Progress in Neural Networks Special edition on Robot Control*. In press. Also available as Research paper number 694 from the Department of Artificial Intelligence, University of Edinburgh.

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

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

Halperin, J. R. P., Dunham, D. W., and Ye, S. (1992). Social isolation of *betta splendens* increases primed social display but decreases display readiness. *Behavioral Processes*, 28:13–32.

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

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, D. J. (1985). *Animal Behaviour.* Longman Scientific Technical.

McFarland, D. J. and Sibley, R. M. (1974). The behavioural common path. *Philosophical Transactions of the Royal Society of London, Series B*, 270:265–293.

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

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

Spetch, M. L., Wilkie, D. M., and Pinel, J. P. J. (1981). Backward conditioning: a re-evaluation of the empirical evidence. *Psychological Bulletin*, 89:163–175.

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

Wilson, S. W. (1990). The animat path to AI. In Meyer, J.-A. and Wilson, S., editors, *From Animals to Animats*, pages 15–21. MIT Press.