Division of Informatics, University of Edinburgh

Institute of Perception, Action and Behaviour

Animal learning models as robot controllers

by

Bridget Hallam

Informatics Research Report EDI-INF-RR-0030

Division of Informatics **July 2000**

http://www.informatics.ed.ac.uk/

Animal learning models as robot controllers

Bridget Hallam

Informatics Research Report EDI-INF-RR-0030

DIVISION *of* INFORMATICS
Institute of Perception, Action and Behaviour

July 2000

pages 131-140 of the SIRS2000 proceedings

Abstract :

Robots can do a range of wonderful things, but they can also appear really stupid. I would like my autonomous, sensor-rich, robot to be able to: complete its task whenever possible, despite distractions and disabilities; learn the best, most reliable cues for success of the various task components; have sensible default actions whenever the situation is unknown; cope with an unpredictably changing environment; and pay attention whenever I want to contact it. Dreamland? At the moment. Yet animals can do these things, and they are not inherently more capable than robots. So why not use an animal model as a robot controller? This paper describes work on the implementation and testing of a model of animal learning in a robotic context. The model is outlined and its interesting features described. An example under-specification problem is given. Experimentation summarised here included a trial naive implementation on an autonomous mobile robot and extensive classical conditioning simulations on computer. More details are given of a simulation experiment to produce behavioural chains and unlearn an unsuccessful chain. Current work involving new robot implementations is outlined. The appropriateness of using an implementation of this model as a robot controller is discussed.

Keywords : training robots, robot learning, simulated biology, simulated classical conditioning, implementing biological models

Animal Learning Models as Robot Controllers

Bridget Hallam[*]

IPAB, Division of Informatics,
University of Edinburgh, 5 Forrest Hill,
Edinburgh EH8 7DQ UK

Abstract. Robots can do a range of wonderful things, but they can also appear really stupid. I would like my autonomous, sensor-rich, robot to be able to: complete its task whenever possible, despite distractions and disabilities; learn the best, most reliable, cues for success of the various task components; have sensible default actions whenever the situation is unknown; cope with an unpredictably changing environment; and pay attention whenever I want to contact it. Dreamland? At the moment. Yet animals can do these things, and they are not inherently more capable than robots. So why not use an animal model as a robot controller?

This paper describes work on the implementation and testing of a model of animal learning in a robotic context. The model is outlined and its interesting features described. An example under-specification problem is given. Experimentation summarised here included a trial naive implementation on an autonomous mobile robot and extensive classical conditioning simulations on computer. More details are given of a simulation experiment to produce behavioural chains and unlearn an unsuccessful chain. Current work involving new robot implementations is outlined. The appropriateness of using an implementation of this model as a robot controller is discussed.

1 Why Should Animal Learning Simulations Interest Roboticists?

Arkin (p31–32, 1998) gives three reasons for justifying the study of animal behaviour under a robotics heading:

> "First, animal behavior defines intelligence. ... Second, animal behavior provides an existence proof that intelligence is achievable. It is not a mystical concept, it is a concrete reality, although a poorly understood phenomenon. Third, the study of animal behavior can provide models that a roboticist can operationalize within a robotic system. These models may be implemented with high fidelity to their animal counterparts or may serve only as an inspiration for the robotics researcher."

There are also situations where the information needed for traditional programming is not available, for example where interactions with unpredictable people are involved, or where the environment is constantly changing, or on distant planets.

Under these sorts of circumstances we want a robot to be able to decide for itself what to do. Not only that, we want it to be able to assess the success of what it did and use that assessment to influence its choice of what to do next time it is in similar circumstances. In other words, we want it to be able to evaluate the consequences of its actions. We would also like it to be able to accept advice, perhaps from humans at the scene, but not necessarily use such advice — after all, the humans may be burglars.

Also, robots which are capable of animal-type learning can be trained in a manner analogous to the methods used for animals (and children). This has the advantage that non-programmers could customise their robot's behaviour relatively easily and intuitively, thus widening the market for robot use and increasing the acceptance of domestic robots. This idea could also be used to make fun toys!

It is unlikely that we could build such a versatile, intelligent machine from first principles. Instead, it seems sensible to learn from studies of existent, versatile, intelligence. As Sutton and Barto [1981, p135] pointed out:

> "Animal learning theory constitutes a large body of carefully explored and tested theories about fundamental processes of learning. Given this, it is surprising how little contact and interaction there has been between animal learning theory and adaptive systems theory, particularly insofar as the latter attempts to mimic neural networks or biological adaptive systems in general."

Fortunately, there is much more interaction between biologists and roboticists nowadays than there was in 1981, although there is still scope for more communication and understanding between the fields.

[*] Email: bridget@dai.ed.ac.uk

2 The Chosen Model

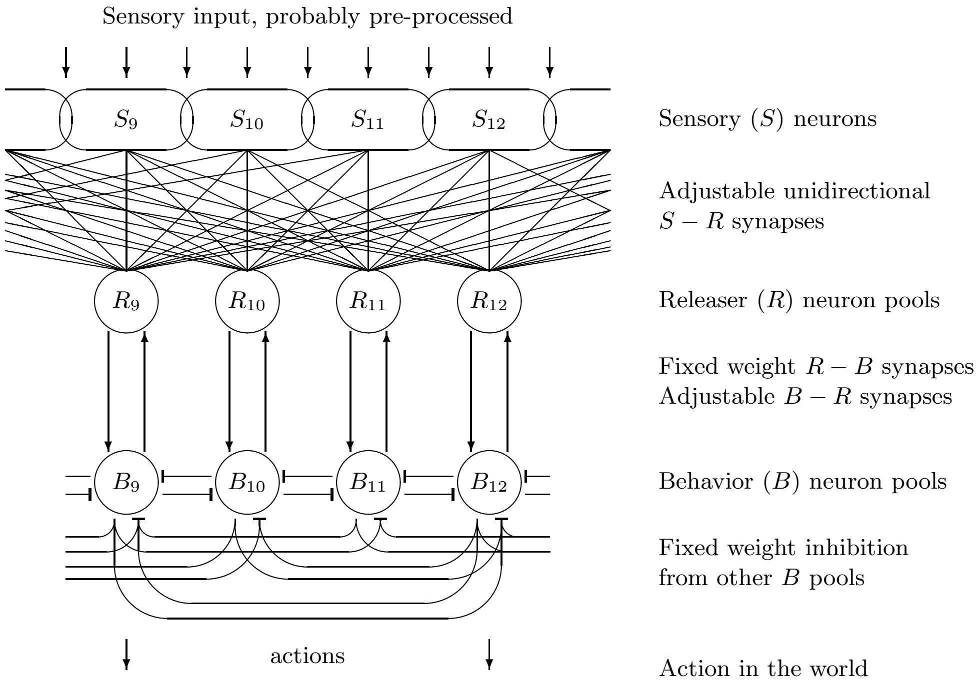


Fig. 1.: Part of a Neuro-Connector Net.

The biological model chosen for testing was Halperin's Neuro-Connector model of learning and motivation [Halperin, 1990]. This model was chosen because of its insight that each successful behaviour has an expected duration, that unsuccessful behaviours tend to run on too long, and that co-incidences are rare. Measuring the duration of robot behaviours is easy, much easier than reasoning about cause-and-effect.

Halperin's Neuro-Connector model of the effect of social isolation on the aggressive display of Siamese fighting fish is a real biologist's theory as to the mechanism of action selection under various circumstances. Halperin [1995, p493] describes her model as:

> "a working hypothesis for the functional mechanism underlying much of vertebrate learning."

If correct, this implies that the model should be able to produce various species-independent learning phenomena. Even in mobile robots. This model had already predicted a new form of learning — postponed conditioning — which was subsequently demonstrated in fish [Halperin and Dunham, 1992].

The model claims that comparing the actual duration of a behaviour with the expected duration is all that is necessary for the building and maintenance of behaviour chains — no reasoning about cause-and-effect is needed. The model also has well-defined learning rules given in mathematical form as well as textual description. It relies on starting with innate (pre-programmed) knowledge, but also learns continuously, updating what it 'knows' in the light of experience. Although new behaviours cannot be learned, the agent discovers the correct (*i.e.*, most likely to be successful) sensory state under which each behaviour should be performed.

This model is presented in detail in Halperin [1990] and Hallam et al. [1997]. Here, the main features are summarised.

The model comprises pools of neurons arranged in three layers: S, R, and B, as illustrated in figure 1. A feedback loop between Rs and Bs allows behavioural persistence. Competition between behaviours prevents conflicting motor signals. Learning takes place in unidirectional $S \rightarrow R$ synapses.

$S \rightarrow R$ synapse weights are adjusted according to their *offset* (finishing firing) times. The actual or observed difference in offset times t_{obs} is compared with an expected time t_{exp}, which is a synaptic parameter. $S \rightarrow R$ synapse weights increase only if R finishes firing within a small time window around t_{exp} after S finishes. The qualitatively different cases are illustrated in figure 2.

Weights increase if firing is correlated as in rule 1. Weights decrease if firing is not correlated as in rule 2, or if only S fires (rule 3). Weights hardly change (rule 4) if only R fires, so that multiple stimuli can cause the same

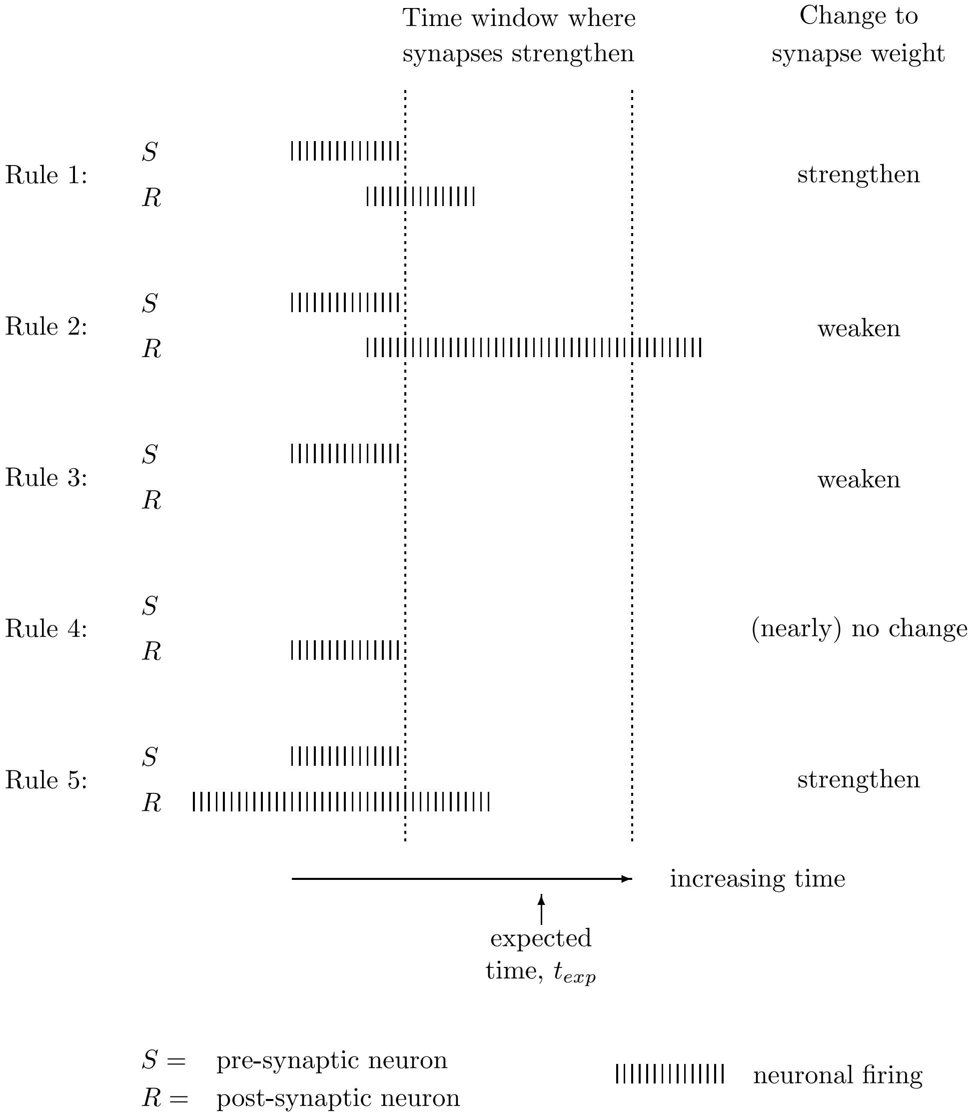


Fig. 2.: Qualitative Weight Adjustments.

response. The most controversial weight change rule is rule 5, which says that large differences in neural onset times can still allow synaptic strengthening if offset times are correlated.

Neurons have a maximum firing duration (*i.e.*, the time from onset until it times out), known as its 'adaptation time'. There is then a period known as the 'refractory time' before which the neuron will not fire again whatever its input.

Although Halperin's model is a biological model, with many good features supporting its biological plausibility, two major problems exist: neurons are binary, and there is no inhibition except between B neurons.

2.1 Basic Theoretical Characteristics of this Model

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

The rate of learning is expected to be fast due to the use of exponential weight-change functions. This fast learning contrasts with the large number of presentations needed generally for reinforcement learning and compares favourably with rates achieved using Perceptron-based nets. Fast learning is normally desirable for robots and is realistic in terms of animal abilities. There is also resistance to forgetting, because no weight change happens unless a relevant stimulus is seen, and even then weights don't change much if the behaviour does not appear.

In general behaviours are expected to produce the releasing stimulus for another behaviour as part of their successful operation. In this way a series of related behaviours forms a chain with each behaviour producing the correct stimulus for the next behaviour in the chain. Since each link in the chain depends upon successful action in the environment surrounding the agent, any failure is noticeable. 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 actions which are successful both contribute to and reinforce (by interrupting the previous behaviour at the 'expected' time) a whole sequence of actions.

Where no new behaviour starts, the current behaviour continues for a while even if its releasing stimulus is removed, due to the $B \rightarrow R$ feedback loop. The behaviours selected thus show persistence in the absence of new stimuli. Behaviours normally continue until one of two things happens: a new stimulus state starts a new behaviour which interrupts the current behaviour, else the current behaviour times out.

For a few self-reinforcing neurons B adaptation time is short enough that R neurons stop firing within the time window where synaptic strengthening occurs. Any stimulus which causes such neurons to fire acts as a reward, reinforcing the previous chain of behaviours.

Normally adaptation times are set to be too long for this automatic positive effect to occur. If no new behaviour appears then the current (unsuccessful) behaviour continues longer than expected by the system. Ethologists call this prolongation of behaviour *after-discharge*. The Rs held on by the behaviour finish firing too long after the Ss that stimulated them and these synapses weaken. Eventually the connections will not be strong enough for the R pool to fire and the consistently unsuccessful behaviour will not be produced under the current stimulus conditions. Then the previous behaviour in the chain will not be interrupted correctly so will after-discharge, weakening a second set of $S \rightarrow R$ synapses. This backward chaining of failure means that eventually the agent doesn't even start a chain of actions known to be rarely successful. Thus the agent does not get stuck retrying an action that is for some reason impossible. Both forward and backward chaining are much in evidence in maze-learning experiments such as those used to investigate reinforcement learning.

When a long behaviour spans several neural adaptation and refractory times, less obviously appropriate behaviours may surface temporarily in the gaps. This is an observable biological phenomenon [McFarland, 1985]. However these minor behaviours are operating under the control of the dominant behaviour and not according to their own inherent timings. They tend to start late, after the appropriate sensory state has been apparent for a time, and finish early when the dominant behaviour recovers. It is unclear whether the probability of occurrence of these behaviours should decrease (because their timings are wrong) or not change (because they don't overrun).

Another 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].

3 The Implementation Process

The process of implementation immediately showed insufficiencies and inconsistencies in the model specifications. The learning rule was very well specified, the architecture was not. Different possible implementations of plausible architectures could lead to qualitively different behaviour. One major difference between the model and the implementation is that the implementation uses a single neuron in place of the pools specified by the model.

The limitations of the specifications in Halperin [1990] and Hallam et al. [1997], and some of the consequences of the various possible alternative in-fills, are described in detail in Hallam [2000a]. An example problem is outlined below.

Experimentation with the mathematics given by Halperin [1990, appendix I] shows that the width of the strengthening window (see figure 2) is fixed at around 2.4 time units. The window can be widened by changing neural gains, but then the weakening caused by behavioural overrun disappears, and this is a central feature of the model.

The textual description of the learning rule states that weights increase only if the actual time difference in R and S offset t_{obs} is approximately equal to the expected time difference t_{exp}; but also states that the strengthening window is 'opened by firing in S [Halperin, 1990, p125]. If a strengthening window with width 2.4 is opened by S onset then S and R burst length have to be extremely short, too short to allow maximal weight changes. If the window is opened by S offset then both rules are possible concurrently, within the mathematics given by Halperin, when and only when t_{exp} is 1.7 model time units. While there is some evidence that there is an optimal timing difference for maximal learning for individual biological systems (*e.g.* about 0.5 seconds for skeletal muscle systems [Woody, 1982], about 8 seconds for suppression of rat licking [Boice and Denny, 1965], a few minutes for suppression of lever-pressing [Kamin, 1968]), this limitation on t_{exp} is inconvenient for the design of simulated or robot systems.

These problems, coupled with differences due to the differences between hardware and 'wetware' and the simplifications inherent in trying to create a program which could be processed in real time on an mobile robot, mean that the implementation is definitely not 'the model'. However, it was sufficiently close to give correct synaptic weight changes with various stimulus patterns, which is considered to be the main focus of the model description.

4 Experimentation

A naive implementation of this model was run on a dustbin-style mobile robot with an on-board PC. The robot learned to distinguish plain walls from boxes against walls [Hallam et al., 1993], in only four presentations. This implementation showed that the system could work, but also showed up many ambiguities and underspecifications in the model. A more principled implementation and more detailed investigation was called for.

4.1 Simulation studies

A much more accurate implementation was used in computer simulation of the model characteristics mentioned in section 2.1. Synapse strengthening and weakening took place as specified for only a narrow range of parameter values.

The stated model characteristics were successfully reproduced. Synapse weights increase and decrease according to the rules of figure 2. Fast learning and fast unlearning are possible, and the model is resistant to forgetting. Behaviours are persistent and can successfully be prioritised, chained, and made self-reinforcing. Behavioural failures can cause weakening of a whole behavioural chain so that chains which are rarely successful can unravel until they are not started (see experiment below). New releasing stimuli can easily be learned.

The main feature of program behaviour discovered that was not discussed by Halperin is that imprecise releasers tend to develop and not weaken out, leading to too many releasers for the behaviours. This has the consequence that high-priority behaviours tend to be performed too often and low-priority behaviours may be forgotten.

Behavioural Chaining This example was chosen to be presented in greater detail because it is the more complex than most of those mentioned above and the most directly relevant to robot behaviour.

Chaining behaviours implies that the Bs can be called in order and that each B called can interrupt the previous one. In this experiment the ordering is achieved by making each behaviour switch on an S which releases a behaviour with a slightly higher priority. Imagine that it is some aspect of the successful completion of the prior behaviour that creates the sensory situation that triggers the subsequent behaviour. Strictly speaking a successful chain does not require differential prioritisation of behaviours since each B can switch on an S just before its own adaptation time removes it from the behaviour competition. But the B adaptation time needs to be set longer than this so that we are able to detect unsuccessful behaviours by their overrunning. Prioritising therefore becomes important when we need to be able to decide whether a behaviour is successful or not.

Chaining Experiment A Neuro-Connector net comprising four S, four R and four B neurons was set up as in figure 3, and high-weight connections (0.6) made between S and R pairs with the same index number. t_{exp} was set at 8 time units for all synapses. Three behaviours were designed that each waited for 6 time units, switched off their own releasing stimulus, waited another 8 time units, and switched on the releaser for the next behaviour. B adaptation time was 17 and their refractory time 50. The fourth behaviour was self-reinforcing, switching off its own stimulus at time 8.5, roughly t_{exp} before its own adaptation time caused it to time out.

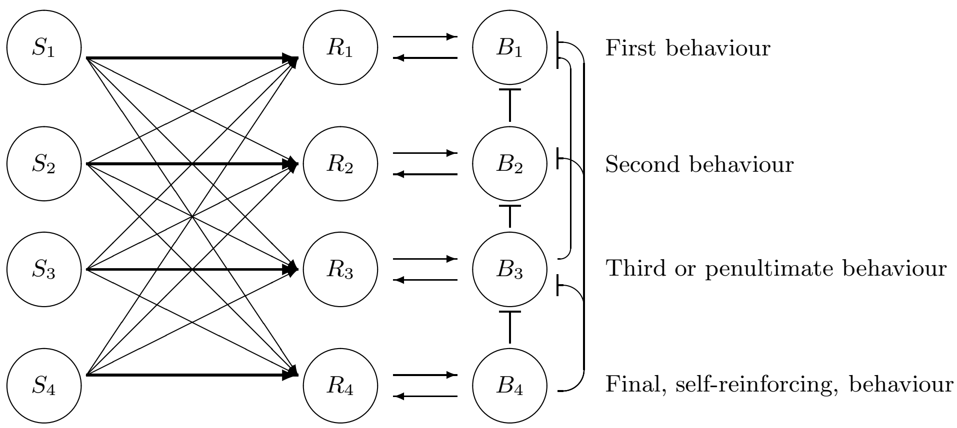


Fig. 3.: Net Used for the Chaining Experiments.

Following presentation of stimulus 1, all behaviours were produced in order, as desired. After 25 such reinforced presentations all high-weight connections had increased weight from 0.6 to 0.99. All low-weight connections stayed low. A problem arose if behaviours did not switch off their own releasing stimuli, since it was easy for timings to be such that the S timed out just before a subsequent behaviour finished, causing strengthening from that S to a later behaviour and removing the earlier behaviour from the chain! Fortunately, it is reasonable to suppose that performance of a behaviour can remove its own releasing stimulus as well as producing the correct releaser for the next behaviour.

This experiment shows that the net can chain behaviours in simulation. When used on a robot each behaviour will not start an S neuron as here but will do some activity expected to produce a change in stimulus state. It is this change in stimulus state which will be detected by the Neuro-Connector net in the form of new Ss firing. Thus successful behaviour in the environment will be necessary for the production of behavioural chains.

Backward Chaining of Failure Halperin [1991], in describing the theoretical behaviour of her model, specifies that a robot performing a sequence of actions which is consistently unsuccessful will learn not to start the sequence. This was tested in simulation as described below.

The last experiment was repeated but with the penultimate behaviour B_3 'failing' by not switching on S_4, the releasing S for the final behaviour. Since the final behaviour did not occur, B_3 overran. This weakened its releasing $S_3 \rightarrow R_3$ synapse until eventually this behaviour was not called and the previous behaviour overran. You may find it helpful to imagine the agent continuing to try the behaviour without success, and so without changing the sensory state in the required manner. Each behaviour in turn overran, keeping its R on too long and weakening the releasing $S \rightarrow R$ connection, dropping out in turn as illustrated in figure 4.

The reason for each behaviour taking longer to drop out than its successors in the chain is that early behaviours continue being reinforced by the correct finishing of later behaviours until the behaviour just after them in the chain fails. Therefore synapse weight still increases for early behaviours even while the final behaviours are failing (figure 4b). All releasing synapse weights started at 0.6; the last surviving high-weight synapse had reached 0.993 before its weight started falling. After 44 presentations none of the behaviours started. This shows that the agent can learn not to start a whole sequence of behaviours which is known not to be successful. This supports the claim of Halperin [1991] that a robot fly with the task of labelling plastic could learn not to waste time on plastics that were not labellable, at least so long as the two types of plastic were distinguishable by other means.

Since there were no other releasing stimuli for any of these behaviours in this experiment, the failing behaviours were lost. In a more complex system other releasing stimuli should be present. As a minimum, there should be one releasing stimulus for each of the slightly different circumstances in which these behaviours should operate. Any Ss which have not fired in these particular 'bad' circumstances will not have lost their releasing ability. There will be overlap in the sensory neuron representation of these stimuli, but hopefully not so much that 'correct' and 'incorrect' releasing stimuli cannot be distinguished.

This experiment proved very difficult to initialise correctly because the weakening that happens when R is on 'too long' fades out so fast with increasing overrun. Initially B adaptation time was set at 20 time units and weakening was too slow to be practical. Even with B adaptation time set at 18 time units instead of the 17 finally used it took 32 presentations for the penultimate behaviour in the chain (the first to disappear) to cease to appear, instead of the 9 needed here.

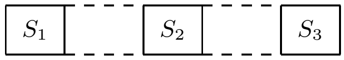
This difficulty implies that synaptic weakening caused by R overrun is likely to be slow, since the timing required to produce maximal weakening has to be so precise. Otherwise, there is scope for adjusting the mathematics of the model both to widen the strengthening window and to make the weakening caused by R overrun less fragile.

Replicating Animal Classical Conditioning Classical conditioning is an animal learning phenomenon whereby animals learn to predict consequences. Learning to predict consequences would be a useful ability for robots. Classical conditioning is easily demonstrated in many variations, has been extensively studied in animals and even to a fair extent in simulation (*e.g.*, see Schmajuk [1997]), and gives results which are surprisingly constant across animal species.

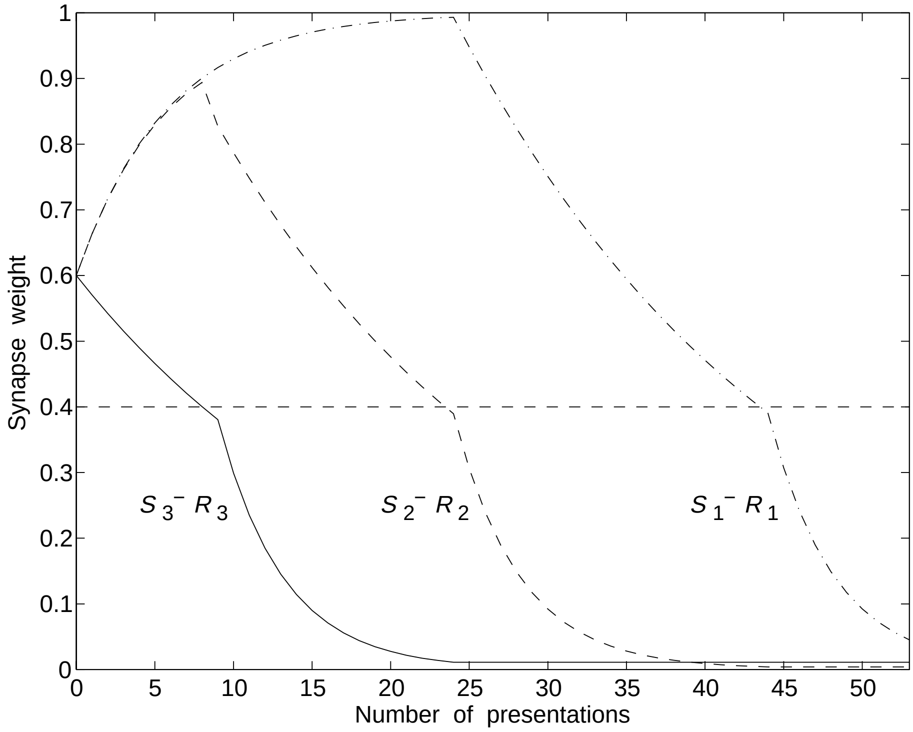
The generality of classical conditioning means that all general models of animal learning (such as Halperin's model is claimed to be) can be expected to be able to produce these effects. It also makes it plausible that a computer or mobile robot should be able to replicate the phenomena.

Basic classical conditioning was easily obtained, maintained, and extinguished. The results of 14 simulated conditioning experiments using my implementation of this model are described in detail in Hallam [2000a], two are also available in less detail in Hallam [2000b]. In summary, eight related classical conditioning phenomena were successfully reproduced (*i.e.*, gave results similar to animal results), four gave results which were similar in some senses but not in others, and two gave results inconsistent with animal studies.

The success of the Neuro-Connector model at replicating these phenomena is similar to the succes of many dedicated models of conditioning, which is impressive given that the Neuro-Connector model was not designed



(a) Stimulus Presentation Sequence



(b) Weight Changes when Behaviour 3 Fails

all bar B_4	B_1 and B_2	B_1 only

(c) Observable Behaviours

Fig. 4.: The Backward Chaining Effect of Unsuccessful Behaviours.

for this purpose. However, obtaining a reasonable replication of animal results required specific assumptions or parameter relationships not mentioned in the model. Many, but not all, of these extra assumptions were reasonable. Occasionally the assumptions necessary for accurate replication of one phenomenon contradicted those necessary for replication of another.

The appearance of biological learning phenomena is not surprising given the biological origin of the model.

The related phenomenon of instrumental conditioning is the basis of most animal training, and it would be nice to be able to *train* our complex robots rather than having to teach them through programming.

New Robot Experiments Most of the previous work using this model has been simulated. Currently we are working on transferring this work to real robots. We have a Neuro-Connector net controlling a Khepera in a Michel simulator [Michel, 1996], almost ready to run on the physical robot. We also have a Lego robot with a 68000 processor on-board which is being controlled by a Halperin net, although neither robot does anything interesting yet. Unfortunately these are not autonomous robots; sensory data is processed on-board but the values are then transmitted to the Halperin net program which processes them and sends the number of the chosen behaviour back. Results from both these experiments are due in September.

5 As a Robot Controller?

The Neuro-Connector model produces some interesting results in simulation, but there is considerable scope for model development. For robot control, the major focus must be on adapting the mathematics of the model to allow t_{exp} to be set with more freedom, while keeping all the major features of the model. Some work on adapting the mathematics of the model is in progress.

The Neuro-Connector model was not developed as a robot controller and is not optimised for this task. It is complex and computationally expensive and is therefore not the best system to use for most of the robot tasks which currently exist and for which other control solutions are available. However, one reason for investigating this model is its potential for controlling a new generation of sensor-rich robots which are intended to perform tasks requiring sophisticated sensory discrimination. The Neuro-Connector model does not decrease the work required to operate a robot, but is intended to make the reliable use of a sensor-rich robot possible. This section attempts to be realistic about the difficulties and advantages of the Neuro-Connector model in this context.

5.1 Robot Requirements

- Behaviours are expected to already exist and to be sufficient for the task(s) given. This is felt to be a reasonable requirement, since most robot tasks are easily modularised into suitable component parts and the production of code to cause reliable operation of small pieces of task is not normally too difficult. Servoed behaviours are allowed by the model and are expected to be useful; not all relevant sensor information has to be fed into S neurons.
- Any effect of power level on the speed of movements needs to be minimised since behaviour duration is so critical.
- Sensor information is expected to be plentiful. Each sensory state which the robot is required to disambiguate must cause firing in at least one specific S neuron which responds to no other states. Currently much imagination must be used to decide which particular combinations of sensor readings are likely to be useful and an S neuron set up for each. This is not considered too difficult a task since extra S neurons cause no trouble. However, it may become a problem if processing capability is severely limited so an optimal solution is required. In future it may be possible to use a self-organising net for this task, at present preliminary trials are needed. These trials, probably using an artificially slow robot to allow sufficient processing time, can show which Ss are most useful. Hand-pruning of excess synapses reduces the problem.
- It must be possible for the programmer to choose normally-right releasing S neurons for each behaviour, so that high-weight synapses can be set appropriately. It is necessary that the robot can behave correctly sufficiently often for the most accurate releasers to be learned.
- It is not necessary that the best, specific, S neurons are known in advance, since they will be learned while the system is in operation.
- Adaptation times for self-reinforcing behaviours need to be correct, so that the behaviour always finishes in time to strengthen its releasing connection. Other behavioural adaptation times need to be sufficiently long that the net can recognise when a behaviour is being performed incorrectly, but the only upper limit on these times is the patience of the observers. Sensory neuron adaptation times sometimes need to be short so that they finish at a constant time before the behaviours they release finish.
- Behavioural priorities need to be set correctly. These are normally obvious within a behavioural sequence, in that later behaviours should always operate as soon as the situation warrants, so should always have priority over behaviours earlier in the sequence. Priorities between sequences are harder to set.
- The values for the parameter t_{exp} and the observed t_{obs} have to match to within a small constant time, 0.8 model time units in these experiments. This may only be possible by preliminary trials involving the timing of behaviour duration and significant sensor readings. Otherwise, it may be possible to use multiple synapses with different t_{exp} between neural pairs.

Future Work This model is inherently computationally expensive, both in terms of the weight change functions and in terms of the number of synapses and neurons which are expected. Work-in-progress includes optimisation of the implementation and some simplification of the mathematics to improve performance. Simplifications making the weight update functions more computationally efficient could be made. Synapses and neurons can be ruthlessly pruned after a learning phase if the main problem for the designer was understanding robot sensors. As long as the environment is not expected to change too much while the robot is in operation, many non-releasing connections can be removed once the robot has met all situations significant to its task. If the environment is expected to change then any pruning must be done with great care as it reduces the ability of the system to learn new connections. Currently, this pruning must be done by hand.

Extra input connections to behaviour neurons can be initialised and used not only for demonstration learning but also for instructing the robot while it is in operation. Human-generated input can have sufficient magnitude to force the robot to perform the behaviour instructed, else can be set at an intermediate level such that the input can be considered as a suggestion.

The current constraints on t_{exp} are decidedly awkward. It would be nice to be able to set t_{exp} to any value, and with an accuracy which is a percentage of behaviour duration. This involves being able to change the size of the strengthening window, if the model feature that the strengthening window is opened by S firing is to be retained. This feature is useful since it means that low-priority behaviours are not 'punished' by the non-performance of the behaviour whenever a higher-priority behaviour is also appropriate. Preliminary mathematical investigations indicate that widening the time window in which strengthening occurs is not a trivial adjustment, and may be impossible without losing 'behavioural overrun' weakening.

Between sequences it would be useful to be able to use variable priorities which depend partly on sensor readings, so that behaviours such as 'recharge' can have a sliding scale of importance dependent upon battery power level. Currently a series of S neurons is needed which respond to 'power level less than x and recharge point less than y away' to enable opportunistic use of recharging points; otherwise the robot will ignore recharging points until its power level is 'low' and then it will stop everything else to find one; or it will recharge whenever it can sense a recharging point, even if it has only just left the same point.

The results so far indicate that un-learning of non-specific releasers (such as those 'good enough' releasers initialised with high-weight synapses) may not be as easy as first supposed. This needs further investigation.

6 Conclusion

The Neuro-Connector model allows behaviours to follow one another in sensible ways. The robot will tend to perform a sequence of behaviours pertaining to a single goal, but will also be able to make use of opportunities which arise and to be interrupted whenever something urgent requires action (including attentional actions, if these have been programmed). Chaining behaviours and responding to interrupts are obviously basic requirements for all robot controllers. The ability to make use of opportunities allows for greater efficiency of operation, especially when the robot has multiple goals.

The model does not require all the information given to it to be correct, although it does require the information to be 'nearly right', enough that behaviours are triggered correctly. The system can therefore be given as much information as is available without fear of failure if some of this information is subsequently discovered to be wrong. This means that it is not necessary to waste information. Any system incorporating sensor noise can accept information which has only a certain probability of being correct, but few check and adjust this information. By contrast, the Neuro-Connector model automatically changes incorrect information about releasing stimuli into accurate information, if the robot is capable of sensing the relevant events. Synaptic expected times do have to be correct, however, and adaptation times have be related to them.

This ability of the net to adjust and refine the releasing stimuli for the behaviours means that the designer does not need accurate imagination about robotic sensory states. As the number of sensory states increases it becomes more difficult for people to understand what each state signifies. It is hard to imagine exactly what state in a sensor-rich robot corresponds to a significant event in the environment and how this state differs from all others, especially once there are several states for each significant event, as is postulated for our system.

This ability to continuously refine sensory data means that changes in the environment are not always fatal either, since the robot will learn to respond to the new sensory states. It should also forget the old ones if these have become wrong., but this may not happen as planned. Many changes in the environment will not affect all the releasing connections for a behaviour anyway, even if these perceived changes are really the result of sensor failure.

It may be easy to run this model in a parallel implementation, since the model itself is inherently parallel. The speed-up afforded would depend greatly on the degree of overlap between S neurons expected to be significant for different subtasks, and thus the number of R neurons attached to each S.

If a behaviour disappears due to its having no releasing connections, then that behaviour cannot reappear unless forcing inputs are available. This means that there is potential for disaster if either the robot cannot reliably disambiguate significant sensory states or if the initial releasing connections were set up badly. In the first case, each potentially releasing S neuron also fires in relation to other behaviours and has its synapses with each behaviour weakened when another behaviour runs. In the second case, poor initialisation of releasing connections means that some behaviours are not performed correctly a sufficient number of times for accurate releasing connections to form.

The usefulness of this model as a robot controller is, unfortunately, more limited than was first supposed. The potential is still extremely interesting, but more model development is required before the implementation will work as a robot controller. Whether it is better to try to improve the Neuro-Connector model or to invent a new model, perhaps using some of the ideas from the biological model, is uncertain. The insight about measuring behaviour duration and comparing this with an estimate of how long the behaviour is expected to take if successful is still extremely valid and is well worth adopting.

Biological models are difficult to implement with any accuracy. However, the attempt is very educational for a roboticist and also produces useful information about the completeness and sufficiency of the model's specification. Experiments using the implementation can also provide insightful feedback to model authors, *i.e.*, can make significant scientific contibutions to biology.

We are working on converting the binary neural output to analogue, so that the measure of a behaviour's priority can include some measure of the urgency of the required behaviour. This is nowhere near as straightforward as it might appear due to the knock-on effects of changing the output on the weight update functions. This conversion to analogue is also important for biological plausibility.

Increasing our knowledge of implementable animal learning should also help us design robots which can learn and react sensibly, as animals can. However, if (when?) we are successful we may have some of the same problems that we have training our children!

References

R. C. Arkin. *Behavior-based Robotics.* MIT Press, Cambridge, MA, 1998.

R. Boice and M. R. Denny. The conditioned licking response in rats as a function of the CS–UCS interval. *Psychonomic Science*, 3:93–94, 1965.

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

B. E. Hallam. *Simulating Animal Conditioning: Investigating Halperin's Neuro-Connector Model.* PhD dissertation, University of Edinburgh, 2000a.

B. E. Hallam. Simulating classical conditioning using a neuro-connector net. In *From Animals to Animats 6.* MIT Press, 2000b.

B. E. Hallam, J. C. T. Hallam, and G. M. Hayes. A dynamic net for robot control. In O. Omidvar and P. van der Smagt, editors, *Neural Systems for Robotics*, pages 227–269. Academic Press, San Diego, CA, 1997.

B. E. Hallam, J. R. P. Halperin, and J. C. T. Hallam. An ethological model of learning and motivation for implementation in mobile robots. Research paper 629, University of Edinburgh, 1993.

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

J. R. P. Halperin. Machine motivation. In J-A. Meyer and S. Wilson, editors, *From Animals to Animats*, pages 213–221. MIT Press, 1991.

J. R. P. Halperin. Cognition and emotion in animals and machines. In H. Roitblat and J-A. Meyer, editors, *Comparative Approaches to Cognitive Science*, chapter 19, pages 465–499. A Bradford Book. MIT Press, Cambridge, MA, 1995.

J. R. P. Halperin and D. W. Dunham. Postponed conditioning: Testing a hypothesis about synaptic strengthening. *Adaptive Behavior*, 1(1):39–64, 1992.

L. J. Kamin. "attention-like" processes in classical conditioning. In M. R. Jones, editor, *Miami Symposium on the Prediction of Behavior 1967: Aversive Stimulation*, pages 9–31. University of Miami Press, Miami, 1968.

D. J. McFarland. *Animal Behaviour.* Longman Scientific Technical, Harlow, 1985.

O. Michel. Khepera simulator version 2.0: user manual. Technical report, University of Nice, 1996.

N. A. Schmajuk. *Animal learning and cognition: A neural network approach.* Problems in the behavioural sciences. Cambridge University Press, Cambridge, UK, 1997.

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

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

C. D. Woody. *Memory, Learning and Higher Function.* Springer-Verlag, New York, 1982.

This article was processed using the TEX macro package with SIRS2000 style