

Design, analysis and comparison of robot learners

Jeremy Wyatt,
School of Computer Science,
University of Birmingham, U.K.
J.L.Wyatt@cs.bham.ac.uk

John Hoar
Department of Artificial Intelligence,
University of Edinburgh, U.K.
johnho@dai.ed.ac.uk

Gillian Hayes
Department of Artificial Intelligence,
University of Edinburgh, U.K.
gmh@dai.ed.ac.uk

January 21, 1998

Abstract

This paper outlines some ideas as to how robot learning experiments might best be designed. There are three principal findings: (i) in order to evaluate robot learners we must employ multiple evaluation methods together; (ii) in order to measure in any absolute way the performance of a learning algorithm we must characterise the complexity of the underlying decision task formed by the interaction of the agent, task and environment; (iii) that in fact this goal is too difficult to attain in practice, and progress in robot learning must rely on comparative work. Four methods for agent analysis are presented. These are used to analyse a robot that learns to push boxes from reinforcement. Using these techniques we have been able to show that $Q(\lambda)$ learning outperforms one step Q-learning on a typical robot task. The differences are statistically significant. We emphasize the importance of experimental design in order to integrate the various forms of evaluation.

1 Introduction

It is difficult to define a coherent experimental method for robot learning. This is because an observed phenomenon may be caused in a number of ways. The robot's behaviour may be the product of the robot's learning algorithm; its initial knowledge; some property of its sensors; the environment; or of an interaction between some subset of these. This makes it very difficult to interpret results. How then can we design robot learning experiments so as to more easily generate meaningful results in the face of such complexity?

The answer arises out of the fact that the cornerstone of any experimental method is the form of evaluation used. Until recently little work has been done on the problems of evaluation in robotic learning [7, 5, 1]. Indeed there are questions as to whether it is possible to meaningfully assess the performance of learning robots [10]. This paper argues that the correct approach is to employ multiple forms of evaluation. This is the only way to disambiguate the source of an error or behaviour when there are many separate causes of any given behaviour. In addition it is the only way to provide detailed explanations of why a learner failed or succeeded.

Fortunately, because robot learning is closely related to a number of fields, it is possible to draw on their experience and evaluation methods. We have adapted methods for evaluation and experimental design from psychology, ethology, statistics and engineering. Four separate methods for analysis are presented here. It is not yet clear which particular subset of techniques will prove most appropriate for the evaluation of robot learning. One finding of this paper is that they are, however, complementary: the weakness of one is the strength of another.

The structure of the paper is as follows. In Section 2 a typology of evaluation methods is suggested. In Sections 4 — 5 we present four forms of evaluation. For each method we describe its application to a specific robot learner. The robot, the learning task, and the learning algorithms employed are described in Section 3. The four methods are: evaluation of sensors and sensory processing; analysis of internal estimates of performance; quantitative external analysis of performance; and qualitative analysis of the robot's behaviour. In Section 7 additional remarks are made concerning the integration of these different evaluation techniques into a coherent experimental framework.

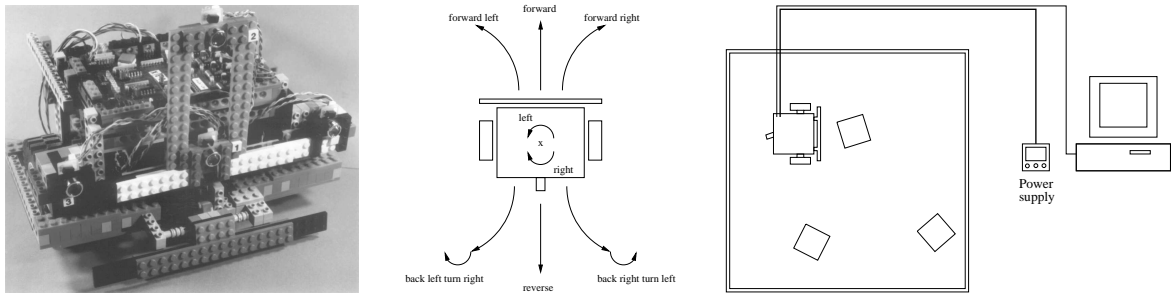


Figure 1: Asterix the robot (left); the set of possible actions (centre); and the experimental setup (right). The actions turn right and turn left are on the spot turns. Turn hard right and turn hard left are not shown; but have the same motor commands as turn right and turn left respectively: merely being of double the duration.

2 A typology of evaluation

There are several criteria for categorising forms of evaluation. The distinctions we draw are between *internal* and *external* evaluation methods; between *quantitative* and *qualitative* measures of performance; and between the *exploratory* and *hypothesis testing* stages of experimentation.

Internal analysis concerns the robot's internal state. This includes the perceived state of the world; the robot's intended action; its estimate of the utility of a particular behaviour; and internal estimates of its current performance. Each of these has an external correlate: the actual state of the world; the action really performed; the utility of a behaviour as apparent to a human observer; and that observer's estimate of the robot's actual performance. Comparison reveals any discrepancies between the two views. The external observer's view is typically assumed to be correct and thus the discrepancy is described in terms of some deficiency in the robot's perception, actuators, or internal performance measure.

The second division is between quantitative and qualitative forms of evaluation. In order to generate an ordering on different robot controllers it is necessary to have a quantitative measure of performance. Such measures are task specific. Internal measures also depend upon the current sensory state. These functions are commonly referred to as reinforcement, error or evaluation functions. The design of such functions is typically not a trivial task [2]. A badly designed function may mislead us concerning the robot's true performance level. A well designed feedback function coupled with inaccurate perception will often do the same. External measures, however, are independent of the robot and its sensory state. Thus they avoid problems created by inaccurate perception and are modality independent.

Quantitative measures of performance may tell us when one method is better than another, or how far short of the optimum a behaviour falls, but they tell us little about why a robot succeeded or failed. This shortcoming is the motivation behind explicit use of qualitative analysis. During the course of developing a robot learner, we typically make many informal observations and draw conclusions based on them. It is sensible to establish these, as far as possible, as a separate form of analysis, by formalising the process of recording qualitative observations. Qualitative analysis exhibits two important components lacking in the quantitative methods. First it requires a rich description of the robot's behaviour, incorporating knowledge not captured in numeric performance measures. Second it leads to the formation of hypotheses as to why the robot behaved as it did. These hypotheses guide the next round of experimentation.

The third distinction drawn is that between exploratory experiments — typically used to form hypotheses to test — and experiments in which we test hypotheses. Normally only the latter type are reported [3]. In fact it is just as useful to report the experiments which led to the formation of a hypothesis. In Section 4 experiments of both kinds are described.

3 The agent, task and environment

The robot used for these experiments, Asterix, was constructed from a Lego Technic kit (Figure 1 (left)). Asterix perceives the world by means of several kinds of sensors. Infra-red emitters/receivers are used to crudely assess the distance between it and nearby objects. A sprung bumper triggers microswitches when Asterix is in front contact with an object. A Hall effect sensor encodes wheel revolutions, and a compass indicates whether

the robot is turning. Each sensor has some subtle properties. Some of these are analysed in Section 4. The robot has eight atomic actions Figure 1 (centre): forward, reverse, turn left, turn hard left, turn right, turn hard right, forward left, and forward right. There are also two composite action sequences: back-left turn-right, and back-right turn-left.

The task was box pushing. The agent is placed within an arena ($2m^2$) with black¹ walls (height $20cm$). The task is a reimplementaion of an experimental setup used Mahadevan and Connell [11]. We argue that implementation in robotics has a purpose different from that in other sciences. Rather than replicating experiments exactly, roboticists investigate whether a technique generalises across slightly different environments and sensory modalities: in order to assess the generality and robustness of the methods employed. Thus the experimental conditions should not be identical. It can be argued that standard equipment weakens the inductive generalisation we can make, because the behaviour may arise as an artefact of the hardware chosen. We follow [6] in arguing that that standard hardware and exact replication are to be avoided if possible.

Box pushing can be divided into three subtasks: box finding, box pushing, and becoming unstuck. Mahadevan and Connell’s hypothesis was that a behavioural decomposition simplified the learning task. A reward function was defined separately for each sub-task. They used Q-learning, in combination with two generalisation techniques. We reimplemented their work [8], using the statistical clustering method described in [11]; as well as the parameter values suggested. The applicability functions and the arbitration network were retained as reported by Mahadevan and Connell. The reward functions initially employed were identical to theirs. We later found it necessary to alter these to obtain the desired behaviour. The final reward functions are detailed in Appendix A.

Mahadevan and Connell used a robot with sonar as its primary sense; an infra-red sensor tuned to 4° as a bump sensor; and a motor current sensor to detect if the robot was stuck. After processing these sensors generated 2^{18} distinguishable states. The sensing on Asterix is rather more crude. In particular it is harder to detect small objects with the infra-red sensors. The changes in sensor modality can be shown to have a significant effect on the ability of the robot to learn, and on the reliability of its reward function. We spent considerable time revising the sensory configuration on the robot, as detailed in Section 5.

Learning computations were carried out on an off-board computer, connected to the robot by a tether, incorporating a power feed and a data-line (see Figure 3 (right)). Although this limits autonomy, it enables the collection of large amounts of data. It also allows the simple comparison of the perceived and actual states of the world; which in turn encourages explanations of why the robot robot behaved as it did.

Asterix, and the ten boxes used, were placed randomly in the arena. The boxes were white; 8-10cm wide; and 10-14cm high. Some boxes were round, or of slightly irregular shape. Every 250 steps in each learning run the boxes were randomly redistributed. This contrasts with the approach taken by Mahadevan and Connell where boxes and robot were given a fixed position, the only random element being the orientation of the robot.

4 Sensor evaluation

Whatever method is used for deriving a robot controller it is essential that the sensory features input to the controller contain sufficient information to distinguish all states relevant to the task. If it can be guaranteed the Markov assumption holds then a convergence criterion for a large class of algorithms (including RL algorithms) has been satisfied. Virtually all robot tasks violate the Markov assumption. One way to try to satisfy it is to carry out a detailed analysis of the sensors and sensory processing on the robot. By such analysis we can detect situations in which *perceptual aliasing* [17] occurs, and introduce additional sensors, or alter the sensory processing, in order to eliminate the aliasing as far as possible. This section presents just such an analysis.

To design sensory processing functions we must have a clear understanding of what the sensory signals mean within the environment in which the agent is situated. Four experiments were carried out on Asterix to understand some of the properties of its infra-red sensors. Two of these experiments were exploratory in nature. The sensory processing was designed based on these. The remaining experiments evaluated their performance in the environment.

4.1 Exploratory experiments with the infra-red sensors

¹We originally ran some experiments with white walls, but found that the robot’s sensors IR were saturated too easily.

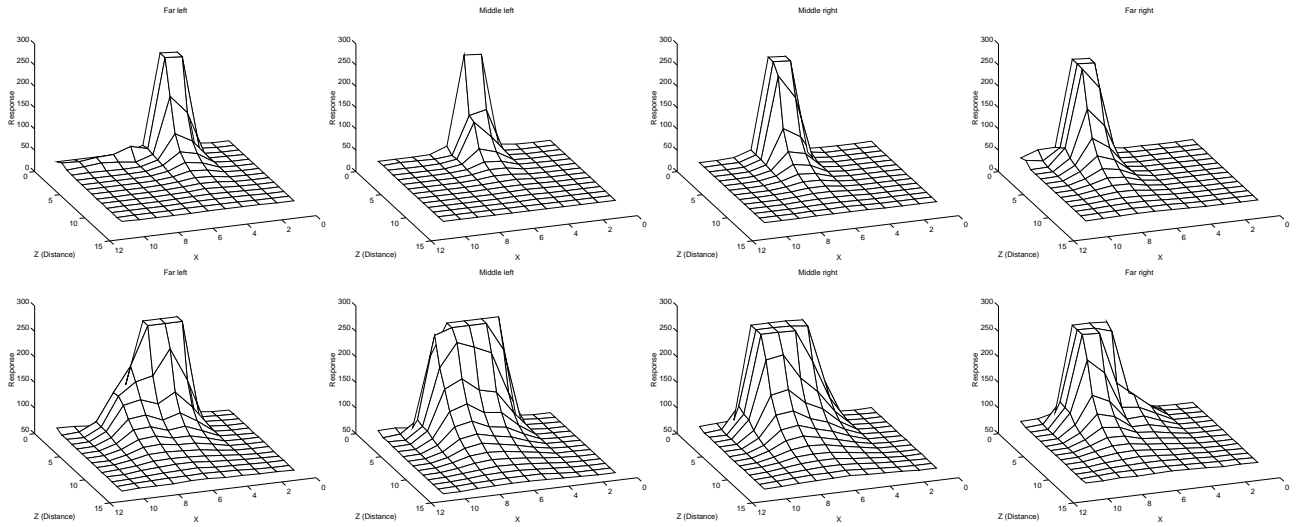


Figure 2: Experiment 1. The top row shows the response maps of each IR sensor in a passive array. The bottom row shows the response map of each IR sensor in an active array. For both rows the panels show from the left to right: the far left IR sensor, the middle left IR, the middle right IR, and the far right IR. Distances are measured in cms; sensor response is the raw figure produced by the sensor (0-255).

4.1.1 Experiment 1

The first experiment investigated the sensor response to boxes. Asterix has four infra-red sensors, which were initially arranged in a row. One of the boxes used for the box-pushing task was moved across a 10 by 12 grid of 5cm squares — in front of the robot — readings being taken from each IR at each position. In this experiment the face of the box was normal to the array of sensors. Each sensor’s response was recorded under two circumstances. First the response of each sensor was recorded with only that sensor emitting (passive array — Figure 2 top row). Next the response of each sensor was recorded with all sensors emitting simultaneously (active array — Figure 2 bottom row).

The readings of different sensors decay at different rates with respect to distance (compare the different panels in the top row), and have an asymmetric response with respect to lateral translation (top row). It can also be seen (bottom row) that when all the IRs are emitting the sensors have a higher response. The effective field of view for each sensor is expanded by the reflected emissions from neighbouring sensors. This increases range, while lowering already poor acuity.

4.1.2 Experiment 2

This investigated how the response changed with respect to changes in orientation of a box. A box was placed at a distance of 15cm in front of the robot, with a face parallel to the front of the robot. It was rotated anti-clockwise through 90° in 5° increments. All sensors were emitting. Figure 3 shows that the response as the box rotates is roughly sinusoidal in shape, the minima roughly corresponding to an orientation of 45° , and the maxima to when the box is facing the sensor. Since the sensors are in an array, these orientations are different for each sensor, and so the responses have different phase.

4.2 Design of general sensory processing

On the basis of these experiments we designed the sensory processing. The robot was eventually equipped with six IR sensors, four facing forward as shown in Figure 3 (right); and one on each side. The side sensors were included so that robot could distinguish which side a wall was on if it was on the edge of the arena. All IR values were quantised into three bins, representing object near, object far and no object present. The distances vary according to the size and colour and orientation of an object; but rough figures for a box with face normal to the sensor were 20cm to trigger near, and 50cm to trigger far.

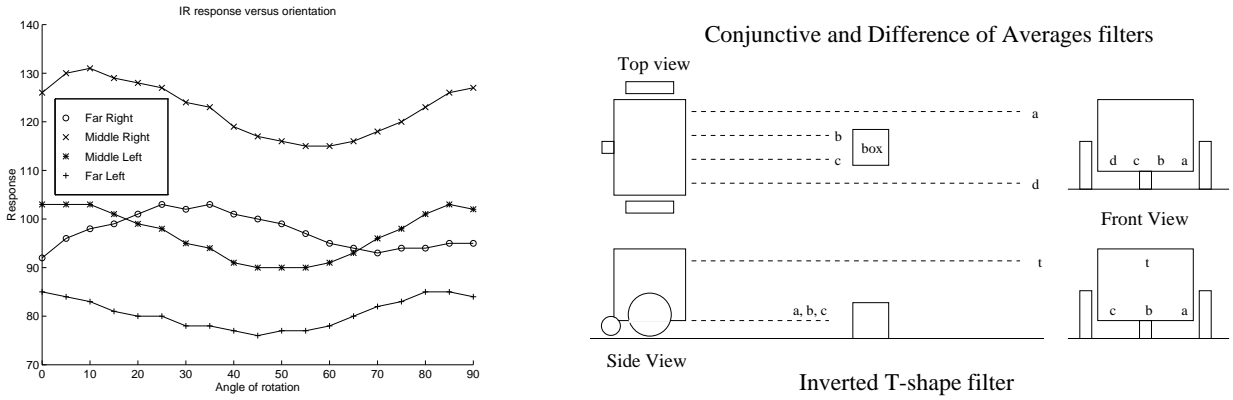


Figure 3: Experiment 2 (left): Response from IRs as orientation changes. IR filter designs chosen for testing (right). The letters a, b, c, d, t represent the positions of IR sensors in each design.

Correlation coefficient		
	Open Space	Near Wall
Conjunctive	0.6030	0.0673
Difference of Averages	0.6030	0.2673
T-shape	0.6804	0.1857

Table 1: Experiment 4: Correlation between positive filter response and the presence of a box, measured using the Pearson product-moment correlation coefficient

The signals from the Hall-effect sensor and compass were combined to create a single bit which indicated whether or not the robot was stuck. If both indicated no change then the robot was deemed to be stuck. The bumper was a simple binary sensor. The box-detection filters described in the following section generate another bit of information, indicating the presence or absence of a box. These sensors give the robot a total of 5832 sensory states. Together with the possible actions this makes the number of situation-action pairs 58320.

4.3 Design and testing of sensory filters

Using the insights gained from these exploratory experiments we designed three possible filters to detect a box more or less in front of the robot (see Figure 3 right). Two of these filters employ the original arrangement of four IRs in a row; the third filter uses an upside down T-shape. The first filter is the conjunctive filter. If the inner two IRs detect something close to but the outer two do not, then the object is narrow and likely to be a box. The difference of averages filter takes the difference between the averages of the two inner and two outer IRs. If this is greater than some threshold then a box is indicated. The T-shape filter thresholds the infra-red sensors individually, as for the conjunctive filter. If any of the lower infra-red sensors detect something, but the top one does not, then a box is signalled. We now describe the experiments conducted to compare the performance of these three filters.

4.3.1 Experiment 4

To compare the filters we initially conducted a statistical analysis based upon their observed identification rates. This allowed us to determine the absolute utility of each filter. This fourth experiment constitutes a test of any hypothesis that a particular filter is best². To test each filter the robot was placed in the arena. On thirty occasions it was placed far from the wall, and on thirty occasions close to the wall. For half the observations in each position a box was placed in front of the robot, in a random position and orientation. When the robot

²Although the tests are not hypothesis tests in the sense of providing significant differences they do provide an ordering on the filters based on a reasonable sample size.

was close to a wall, the robot’s position and orientation were also slightly randomised. It was recorded whether or not the filter’s output indicated the presence of a box.

Table 1 gives the Pearson product-moment correlation coefficient for each filter. This gives a measure of each filter’s rate of success. A value of -1 indicates that the filter is likely to be active when a box is absent, $+1$ indicates that the filter is likely to be active in the presence of a box. A value of 0 indicates no correlation. As can be seen, each of the filters has a moderate rate of success when the robot is far from the wall. Performance near the wall is much poorer.

There is a problem with a correlation based measure of a filter’s success. We must assume that the distribution of examples used to test each filter is the same as the distribution of the robot’s experiences when actually performing the task. This is typically not the case. It is practically impossible to pin down for learning

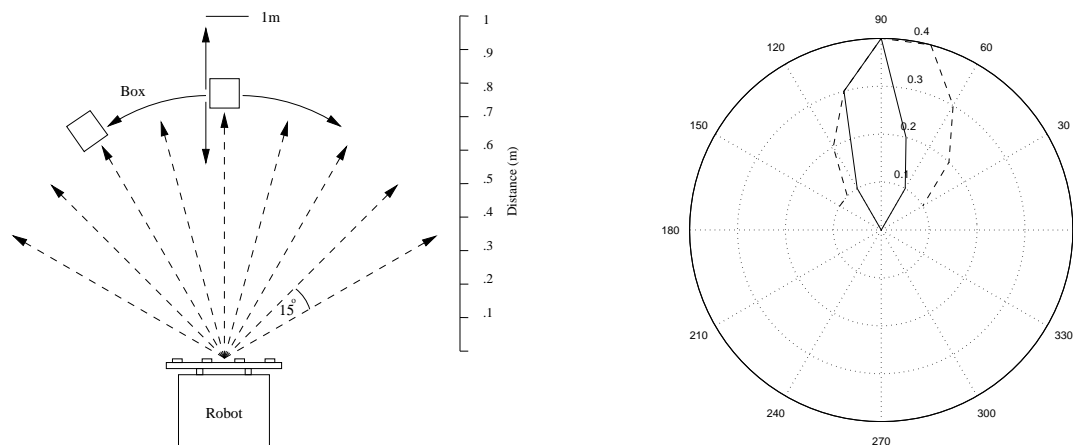


Figure 4: Experiment 5. Method (left): Varying the position and orientation of a box in open space. The robot and box are drawn roughly to scale. Results (right): the dashed line gives the outer limit of box recognition for the T-shape filter, the solid line for the conjunctive filter. The concentric rings have radii increasing in jumps of 10cm.

experiments exactly what such a distribution will be, as it will change through time in response to the changes in the robot’s behaviour. Secondly the class of algorithms used relies on the task being Markovian. A correlation based measure gives no information about the behaviour of a filter through time, and thus nothing about the sequence of experiences the robot might reasonably have.

One of the filters designed actually behaves in a manner which violates the Markov assumption. This can be seen from the more detailed analysis of the filters conducted in Experiments 5–7. In the first the behaviour of the box filters was examined when the robot was in open space. In the second, the robot was placed closer to the wall, and the position of a box altered. In the third no box was present and the behaviour of each box filter was examined in response to changes in the robot’s position and orientation when close to the wall.

4.3.2 Experiment 5

In the first experiment the robot was placed in open space in the centre of the arena³. The box was then placed in front of it along a line of projection perpendicular to the robot’s face. The box was placed at distances from 0cm to 100cm and the response of each filter recorded at 10cm intervals. This process was repeated along 8 further lines of projection. These were obtained by rotations about the centre front of the robot of 15° , 30° , 45° , and 60° ; four to the left and four to the right, as shown in Figure 4 (left). This experiment was conducted on the difference of averages filter, and the T-shape filter. The results are recorded in Figure 4 (right), where the plot gives the maximum distance at which a box could be identified by each filter. It can be seen that the T-shape filter has a wider field of recognition. Neither filter works beyond 40cm. This experiment thus gives a much more detailed account of the filter response in open space than Experiment 4.

³This is about 1.3m from any wall. It was verified that moving the robot around the centre made no difference to the base IR response.

4.3.3 Experiment 6

The purpose of this experiment was to detail the way in which filter response varied as the robot neared the wall. The robot was placed, face parallel to the wall at distances from 80cm to 20cm, in 10cm increments (see Figure 5). The box was placed between the robot and the wall, face parallel to the robot. The distance between the box and the robot was varied in steps of 10cm. At one extreme the near face of the box touched the robot bumper; at the other extreme the far face touched the arena wall. The results are detailed for the T-shape filter in Table 2, and for the difference of averages filter in Table 3. The most striking result is that T-shape filter produces a systematically erroneous response. The box disappears at a distance of 20cm when the robot is less than 80cm from the wall. This does not happen with the difference of averages filter. The phenomenon occurs because at a certain distance from the wall the reflected light from the wall and the box combine to stimulate the top IR sensor (sensor t in Figure 3 right). This will create a non-Markovian effect as the robot approaches the box. Thus it can be seen that even with very simple sensors and sensory processing, non-Markovian effects can arise that will be overlooked if sensory analysis is inadequate.

4.3.4 Experiment 7

In Experiments 5 and 6 the behaviour of each filter in the presence of a box was investigated. We saw that the T-shape filter systematically generates false negatives. But does either filter ever generate false positives? In this experiment the behaviour of each filter near to the wall but in the absence of a box was investigated. The robot was placed with face parallel to the wall at distances from 0-50cm from the wall and the response of each box filter was recorded at 10cm intervals. This process was repeated along seven different lines of projection. These were obtained by rotations about a point on the wall of 15° , 30° , 45° , 60° , 75° and 90° (see Figure 6). All rotations were to the robot's left. The responses are shown in Tables 4 and 5. The T-shape filter again performs poorly, generating false positives at a number of positions. The difference of averages filter generates none.

4.4 Summary

The analysis carried out in Experiments 5-7 explains the results obtained in Experiment 4. The T-shape filter has a larger field of recognition and so has a slightly higher correlation coefficient in open space. Close into the wall it generates both false negatives and positives, and the correlation coefficient is lower. However, what is important is not the correlation coefficient, but the fact that using the T-shape filter will generate non-Markovian effects as the agent moves around the environment. The other filter avoids this problem. Clearly in order to guarantee that the Markov assumption has been satisfied a detailed analysis is required, even for simple tasks. This is a significant drawback for methods such as reinforcement learning which make the assumption. We also conducted less detailed analyses of the other sensors. These are not reported here.

5 Quantitative comparison of robot learners

Having investigated the sensors, and developed appropriate filters based we ran learning experiments as described in Section 3. During each experiment the internal reward generated by the agent and the actual (externally measured) performance were both recorded. The former is the robot's estimate of its performance. The latter is an external observer's estimate of performance. The external measure is to be accepted as correct in the event of a discrepancy between the two. We encountered such discrepancies many times. These led to revisions to the robot's morphology (twice); the environment (twice); the set of actions the robot could perform; the robot's sensing (three additions); and fifteen revisions of the reward functions. Detailed qualitative analysis of the robot's interaction with the environment was required on each occasion in order to make the required change. Examples of this form of analysis are given in Section 6. This process was repeated until improvements in internally generated reward corresponded to improvements in externally measured performance. Only when the robot can be shown to have learned to improve its actual behaviour can we make meaningful comparisons between learning algorithms.

There are further issues in quantitative evaluation. First, what exactly is the relation between the internal and external measure of performance? When we design a reinforcement function from which a robot should learn we carry out two transformations on the minimal definition of optimal behaviour. First it is necessary to correlate the conditions that define success with the robot's sensory states. Second, we typically provide a more

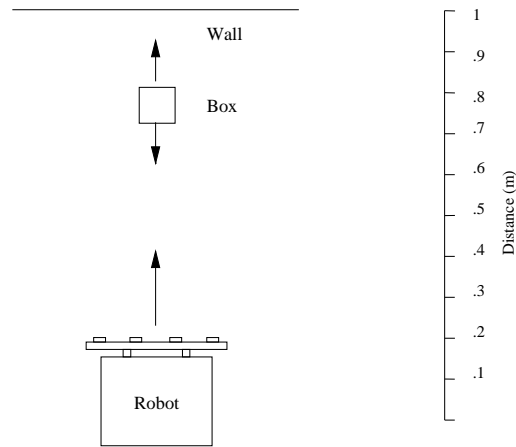


Figure 5: Experiment 6. Method: varying the distance of box and robot from the wall. This experiment was all conducted with the T-shape filter and the difference of averages filter.

Dist. betw. robot & wall	Dist. betw. box and robot (cm)							
	0	10	20	30	40	50	60	70
20 cm	+	+						
30 cm	+	+	-					
40 cm	+	+	-	+				
50 cm	+	+	-	+	+			
60 cm	+	+	-	+	+	+		
70 cm	+	+	-	+	+	+	-	
80 cm	+	+	+	+	+	+	-	-

Table 2: Experiment 6. Results: T-Shape filter. A + means that filter indicates a box, a - means that the filter indicates no box.

Dist. betw. robot & wall	Dist. betw. box and robot (cm)							
	0	10	20	30	40	50	60	70
20 cm	+	+						
30 cm	+	+	+					
40 cm	+	+	+	+				
50 cm	+	+	+	+	+			
60 cm	+	+	+	+	+	-		
70 cm	+	+	+	+	+	-	-	
80 cm	+	+	+	+	+	-	-	-

Table 3: Experiment 6. Results: Difference of averages filter. A + means that filter indicates a box, a - means that the filter indicates no box.

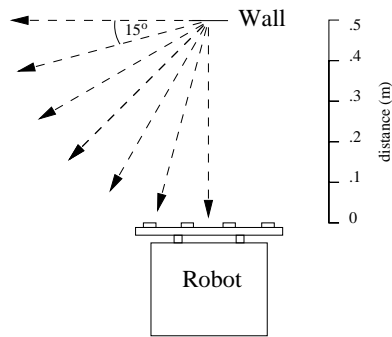


Figure 6: Experiment 7. Method: Varying the distance and orientation of the robot relative to the wall when no box was present. This experiment were all conducted with the T-shape filter and the difference of averages filter.

Angle from normal (degs)	Dist. betw. robot face and wall (cm)						
	0	10	20	30	40	50	N/A
0	-	-	+	+	-	-	
15	-	-	+	+	+	-	
30	-	+	-	-	-	-	
45	+	+	-	-	-	-	
60		+	-	-	-	-	
75			+	+	-	-	
90							-

Table 4: Experiment 7. Results: T-Shape filter. A + means that filter indicates a box, a - means that the filter indicates no box. Blank cells indicate no reading was taken. At 90° the robot was facing along the wall, distance thus being irrelevant.

Angle from normal (degs)	Dist. betw. robot face and wall (cm)						
	0	10	20	30	40	50	N/A
0	-	-	-	-	-	-	
15	-	-	-	-	-	-	
30	-	-	-	-	-	-	
45		-	-	-	-	-	
60			-	-	-	-	
75						-	
90							-

Table 5: Experiment 7. Results: Difference of averages filter. A + means that filter indicates a box, a - means that the filter indicates no box. Blank cells indicate no reading was taken.

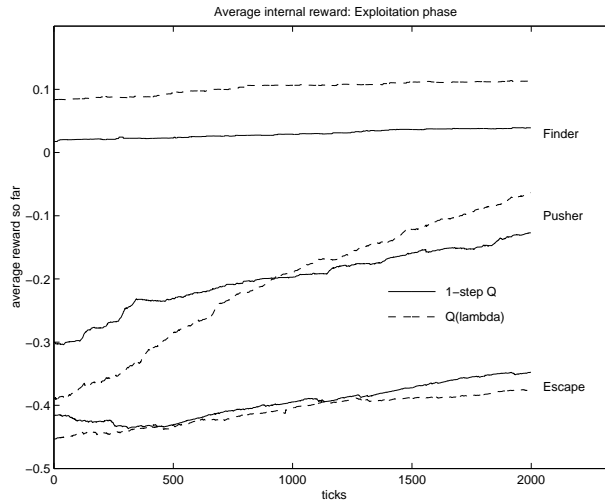


Figure 7: The mean average reward generated during the exploitation phase for all behaviours over 4 runs.

detailed performance measure than is strictly necessary in the hope that this will speed convergence. When carrying out external analysis, we can use a reward function devoid of these additions. In the box-pushing task, the performance is judged on the overall behaviour rather than for each sub-task — as is done by the decomposed learner. A simple measure is to time the total period of the run during which the robot was pushing a box. One last problem is that although both measures are quantitative only qualitative differences between them can be used. We can only say there is a discrepancy between the internal and external performance measures if one is increasing through time (indicating the robot is improving) while the other is falling through time (indicating the robot is getting worse), or is roughly constant. If performance improves by both criteria, even if at very different rates, then we cannot draw the conclusion that a discrepancy exists. Typically the discrepancy will occur when the *internal* performance metric indicates that the robot’s performance is *improving*, while the *external* performance metric tells us it is static or *decreasing*.

There is a second difficulty. There can be no impartial evaluation by an experimenter with an a priori expectation of outcome. It is thus necessary to introduce blind evaluation to robot learning. How can we separate the operational and observational roles in our experimental work? The route we took was to separate the observer and operator. The observer is ideally someone with minimal knowledge of the aims of the project, i.e. the observer is blind to the hypothesised outcome, and to the different treatments employed. To physically separate the observer and operator we employed a video based evaluation method. Each run recorded was viewed ‘blind’ by the observer, who marked the robot’s performance according to the external metric.

5.1 Experiment 8: Comparison of Q-learning and $Q(\lambda)$

In this experiment the hypothesis to be tested was that one-step Q-learning would be outperformed by a less widely used form of Q-learning called $Q(\lambda)$ [12]. $Q(\lambda)$ combines Q-learning with eligibility traces as employed in the well-known TD(λ) algorithm [15, 13]. Previously $Q(\lambda)$ has been shown to outperform one step Q-learning on simulated tasks [12, 19]. The version of the algorithm proposed by Peng and Williams, is exploration sensitive however. Several authors have noted that a simple change removes exploration sensitivity [14, 19, 16]. We refer to this altered version as *corrected* $Q(\lambda)$ ⁴.

The learning period was split in two: a period during which the robot explored the environment; and one during which the robot mostly exploited the environment (followed the greedy policy). An exploratory period was employed so that the robot gained a variety of experience, and did not become stuck in a sub-optimal policy. To measure the worth of the policies learned it is necessary to have a distinct period during which exploration is switched off and exploitation dominates. Measuring performance during the exploration phase merely introduces additional noise into the results, thus increasing the sample size required to generate significant differences. We

⁴Details may be found in [19, 8].

only report performance in the exploitation phase. Learning occurred during both phases. Precise details of the learning algorithms used are given in [8]. Replacing rather than accumulating eligibility traces were used [13]. The reward functions and experimental parameters used are listed in Appendix A. Each exploration phase was 3000 steps long, and each exploitation phase was 2000 steps. Four runs of each learner were made.

5.1.1 Evaluation by internal reward

Run	1-step Q	$Q(\lambda)$
	mins:secs	mins:secs
1	3:15	5:07
2	3:20	4:00
3	1:31	3:44
4	3:31	5:05

Table 6: Experiment 8. Results: Total time that the robot pushed boxes.

At each time step the internal reward generated was recorded for the active behaviour. The average reward generated over the exploitation period so far was calculated for each time step. Figure 7 shows this figure averaged over all four runs for each of the agent types. The internal reward for each behaviour increases for both learners throughout the learning period. The box pushing and escape behaviours improve with respect to the reward function defined, and the box finding behaviour shows a slower improvement. The maximum and minimum possible rewards are detailed in Appendix A. More importantly with respect to the hypothesis being tested the $Q(\lambda)$ learner decisively outperforms the one step Q -learner in two of the three behaviours. We conducted a Mann-Whitney test [4] on the average reward generated over the whole run for each behaviour. The difference between the learners is significant at the 5% level (the actual probability of the NH is 0.0152) for the pusher behaviour, and at the 10% level for the finder behaviour. The difference for the escape behaviour is not significant. But does this internal improvement correspond to an external improvement?

5.1.2 Evaluation by externally measured performance

To confirm that both agents had indeed learned to improve their performance with respect to the task we measured their actual performance over the exploitation phase. Each exploitation phase was videoed for each run, and the total time the agent spent pushing boxes recorded. The results are shown in Table 6. Each exploitation phase lasted for about 40 minutes. It can be seen that in every run the $Q(\lambda)$ learner outperformed the one step Q -learner. A Mann-Whitney test was conducted and the difference was again found to be significant at the 5% level (again $\Pr(\text{NH}) = 0.0152$).

6 Qualitative analysis

We previously noted that during the development of the learner, learning runs were made, which unlike the runs in Experiment 8, revealed a discrepancy between actual and supposed performance. In order to identify the source of the discrepancy we carried out a qualitative analysis of the robot’s behaviour. This section describes some typical example behaviours observed, and the hypothesised causes. The main finding of this section is that a great deal of work understanding the interaction between the robot and the environment is required in order to be able to successfully isolate the source of a behaviour and thus revise the sensory processing, morphology, action space, or reward function.

In an early run of the robot it was discovered that while the agent was improving its internal rewards with respect to the pusher and finder behaviours it was typically not performing as well as suggested. On careful observation of video tapes of the behaviour, we concluded that the robot learner was in fact converging to one of two overall behaviours. We refer to these as the “box avoider” and the “wall pusher” respectively (see Figure 8).

In the box avoiding behaviour the robot recognised boxes, generated reward by moving toward them, and then turned away. We hypothesised that this was because the reward function for the finder rewarded seeing boxes more heavily than bumping into them. Thus the agent generated more finder rewards by staring at boxes than by bumping into them; although when a box was bumped the box pusher module takes over and the box

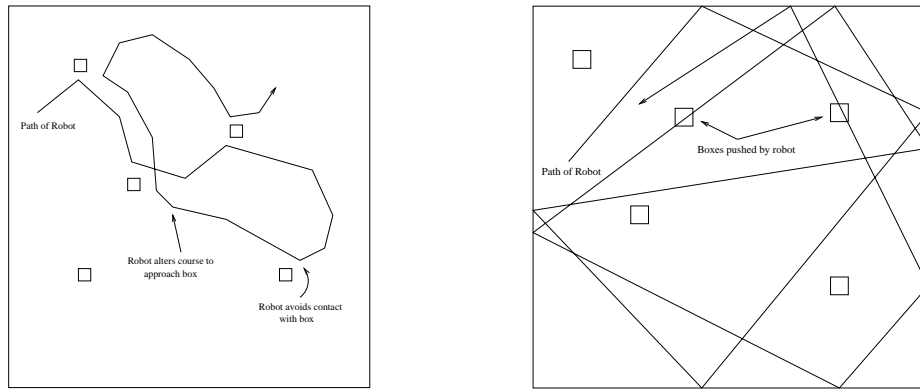


Figure 8: The box avoider (left) and the wall pusher (right)

was pushed. This is an example of a behaviour generated by an inappropriate reinforcement function. Designing appropriate reward functions is notoriously hard. Hence this is a phenomenon that we should expect to see frequently during the development of a robot learner.

In the wall pushing behaviour we found that the agent traversed the environment in straight lines, bumping into walls, pushing them, and then turning away. By comparing the robot's internal state to this behaviour, we found that the box pushing module became active and was rewarded when the robot bumped against a wall. This is because the Hall effect sensor — unlike the motor current sensor used in [11] — has a delay in reporting the cessation of motion. In addition it is unreliable. Because of this the agent is rewarded for pushing walls for a short period prior to the unwedging behaviour being activated. This is an example of subtle changes in the sensors making perception unreliable, and a reasonable reward function misleading.

7 Problems of Experimental Design

This paper has outlined four forms of evaluation. How these can be tied together into a coherent experimental framework? The general structure of our actual experiments is apparent from the sequence given in the paper. In this section some more detailed remarks are made about the manner in which the complete process of devising, designing and executing robot learning experiments might be conducted.

One possible sequence of events for robot experiments is as follows. First the general hypothesis to be tested should be outlined. In our case it was to compare the performance of $Q(\lambda)$ learning with simple one-step Q-learning. The next stage is the choice of task. This is inextricably linked to the robot type, and to the available sensors and actuators. Currently there is no canonical set of robot learning benchmark tasks. We believe that benchmark tasks have only a limited value in robot learning. This is because to reasonably compare the performance of one robot with another we must accurately characterise the underlying decision task. It is clear from Section 4 that this is a near impossible task. Even with detailed analysis we were only able to establish the most basic facts about the robot-environment interaction. To fairly compare the performance of two algorithms on two separate robots would require a much more detailed model. In fact it is much easier to discard the notion of benchmark tasks altogether, just as earlier we argued against the use of standard hardware. It is our contention that progress can only be made by careful comparative work.

Once the task has been selected then the internal and external performance measures can be more precisely defined. The internal performance measure may depend on particular sensors, and this may influence the choice of robot and environment. The next stage involves the construction of the robot and the environment. Following this the properties of the sensors must be analysed. There are two relationships that must be studied: the properties of the sensors with respect to a static environment; their behaviour when the robot interacts with the environment.

Following this preparatory work the actual comparison of a set of learning techniques can be carried out. Typically, in order to understand the properties of the robot and the environment a human-designed non-adaptive controller is also tested in order to check that the task can be performed using the robot. We did not carry out such an experiment and suggest that although using hand-designed controllers as experimental

controls is a laudable idea, they should not be developed prior to the conduct of the experiments with the learning controllers themselves. This is because in the process of implementing any controller a great deal is typically learned about the task. We can only accept the alternative hypothesis that controller A (adaptive) is more powerful than controller B (hand-coded) if no unseen advantage have been given to controller A. Therefore the hand-coded controller must be implemented second to ensure that any knowledge advantage lies with the controller hypothesised to be worse. The alternative hypothesis will typically be that the learning controller is better.

The order in which the various quantitative evaluation methods are conducted may not matter, given that the observer for external evaluation should be separated from the robot operator. Clearly at each stage sufficient data should be collected in order that statistically significant results can be obtained wherever possible. If discrepancies are discovered between the internal and external performance measures then the source of the discrepancy must be identified. We suggest that the cause may be one of three. Either the internal performance measure is inappropriate; the sensory information available to the robot is inadequate; or additional knowledge used to constrain the learning task is incorrect. This paper has given examples of the first two. Examples of the third would be if the decomposition of the controller into a hierarchy of simpler controllers was inappropriate and prevented learning; or if the learning controller's experience was biased using a teaching method, and the bias thus induced prevented the robot subsequently converging to the optimal behaviour.

Finally following the detection of any such discrepancy the experiments must be reconducted once the source of the error has been identified and eliminated. In order to eliminate the discrepancy between the internal and external performance measures hypotheses must be formed as to the source, and these hypotheses tested in turn. The method we used to form hypotheses concerning the source of the error was based on a qualitative analysis of the robot's behaviour. Thus the experimental process typically consists of many iterations in which we adjust the sensory processing and the internal performance measure until the learning techniques can be compared. A different technique would be required if the learning algorithm itself were designed to select the sensory features required to learn successfully [9, 18, 17].

8 Conclusion

The main findings in this paper are as follows. First, multiple evaluation methods are required in conducting robot learning experiments. Second, we have argued against the use of benchmark tasks. To be meaningful comparative work must be carried out on a single experimental platform. This is because absolute measures of performance used to compare results on different robots have limited meaning because of the enormous variation in robot sensory capabilities. Third, we have argued against standard hardware as a solution to this problem. Reimplementation on different hardware is the key to inductive generalisation in our field. Exact replication of robot experiments is of limited value.

Turning to the details of our study we have found that internal estimates of performance are inadequate if there is any likelihood that the sensors upon which the reward function depends are unreliable. Second, quantitative external analysis according to a similar performance metric can be used to detect discrepancies between the actual performance and that estimated by the robot. Third, qualitative analysis is absolutely necessary in order to provide a rich enough description of the robot's behaviour to generate explanations of why the robot behaved in the way it did. These hypotheses are used to modify the agent's sensors, sensory processing, morphology or reward function.

It is important for the particular class of algorithms studied that the Markov assumption hold. The only way to verify this is by carrying out a detailed analysis of the sensory properties of the robot. This takes up the majority of the development time. This must be regarded as a major drawback for robot learning algorithms which make the Markov assumption. Finally we have shown that one step Q-learning, currently the most widely used robot learning algorithm, is significantly outperformed on a real robot task by corrected $Q(\lambda)$ learning.

Acknowledgements

Earlier versions of this paper were improved by comments from the TIMR reviewers, and Axel Grossmann. Thanks to Sandy Colquhoun, Andrew Haston, Tom Alexander, Nuno Chagas and John Hallam for help with hardware, and to Douglas Howie for taking the photo of Asterix. This work was partially supported by EPSRC.

References

- [1] Tucker Balch. Social entropy: a new metric for learning multi-robot teams. In *Proceedings of 10th International Florida Artificial Intelligence Research Symposium*, pages 272–277. Florida AI Research Society, 1997.
- [2] A. G. Barto. Connectionist learning for control. In W. T. Miller, R. S. Sutton, and P. J. Werbos, editors, *Neural Networks For Control*, pages 5–58. MIT Press, 1990.
- [3] P.R. Cohen. *Empirical methods for Artificial Intelligence*. MIT Press, 1995.
- [4] G.M. Clarke & D. Cooke. *A basic course in statistics*. Edward Arnold, 3rd edition, 1992.
- [5] Yassine Faihe and Jean-Pierre Muller. Analysis and design of a robot's behaviour: Towards a methodology. In John Demiris Andreas Birk, editor, *Proceedings of Sixth European Workshop on Learning Robots*, 1997.
- [6] John Hallam and Gillian Hayes. Benchmarks for mobile robotics? In *Towards Intelligent Mobile Robots: scientific methods in mobile robotics*. Manchester University, School of Computer Science, 1997.
- [7] Henry Hexmoor. Robolearn 97: An international workshop on evaluating robot learning. Technical report TR 97-03, Department of Computer Science, State University of New York at Buffalo, April 1997.
- [8] J. Hoar. Reinforcement learning applied to a real robot task. Unpublished masters thesis, University of Edinburgh, Department of Artificial Intelligence, September 1996.
- [9] David Chapman & Leslie Pack Kaelbling. Input generalisation in delayed reinforcement learning: an algorithm and performance comparison. In *Proceedings of International Joint Conference on Artificial Intelligence*, 1991.
- [10] Leslie Pack Kaelbling. *Learning in Embedded Systems*. PhD thesis, Dept of Computer Science, Stanford, 1990.
- [11] S. Mahadevan and J.H. Connell. Automatic programming of behaviour-based robots using reinforcement learning. Research Report RC 16359 (72625), IBM Research Division, July 1990.
- [12] Jing Peng and Ronald J. Williams. Incremental multi-step q-learning. In W.W.Cohen and H.Hirsh, editors, *Machine Learning: Proceedings of the 11th International Conference*, pages 226–232, 1994.
- [13] Satinder Singh and Richard Sutton. Reinforcement learning with replacing eligibility traces. *Machine Learning*, 1996. Accepted for Publication.
- [14] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press/Bradford Books, 1998.
- [15] R.S. Sutton. *Temporal Credit Assignment in Reinforcement Learning*. PhD thesis, University of Massachusetts, School of Computer and Information Sciences, 1984.
- [16] C.J.C.H Watkins. *Learning from delayed rewards*. Thesis, University of Cambridge, King's College, Cambridge, England, May 1989.
- [17] S.D. Whitehead and Long-Ji Lin. Reinforcement learning in non-markov decision processes. Submitted to the Special Issue of the AI Journal on 'Computational Theories of Interaction and Agency'.
- [18] Steven D. Whitehead and Dana Ballard. Learning to perceive and act. Technical Report TR-331 (revised), University of Rochester, Department of Computer Science, June 1990.
- [19] J.L. Wyatt, G. Hayes, and J. Hallam. Investigating the behaviour of $q(\lambda)$. Technical Report 783, Department of Artificial Intelligence, Edinburgh University, January 1996. Presented at the IEE Colloquia on Self Learning Robots, Feb 12 1996, London.

9 Appendix A

9.1 Reward functions employed in Experiment 8

The reward function for the finder behaviour was as follows:

```
if BOXt and BUMPt+1 then
  rt = 5
else if BOXt and not(BOXt+1) then
  rt = -3
else if BOXt+1 then
  rt = 1
else
  rt = 0
```

where r_t is the reward at time t ; and BOX_t and BUMP_{t+1} indicate that the robot perceives a box at time t and that the robot's bumper is depressed at time $t + 1$ respectively. The reward function for the pusher behaviour is:

```
if BUMPt+1 and at = (forward or forward-left or forward-right) and not(STUCKt+1) then
  rt = 1
else if not(BUMPt) and BUMPt+1 then
  rt = 0.5
else if BUMPt and not(BUMPt+1) then
  rt = -4
else
  rt = 0
```

where a_t is the action at time t ; and STUCK_{t+1} indicates that the robot's compass and Hall-effect sensor both indicate it is not in motion. The reward function for the escape behaviour is:

```
if at = (reverse or forward) then
  rt = -1
else if STUCKt+1 then
  rt = -3
else
  rt = 0
```

This indicates that the maximum and minimum rewards for each behaviour are: finder=(5,-3); pusher=(1,-4); escape=(0,-3). In all experiments rewards were discounted using $\gamma = 0.9$ in the calculation of return.

9.2 Parameters for Experiment 8

As mentioned in the main text the learning component consisted of a Q-learner combined with a statistical clustering method originally developed by Mahadevan and Connell. The parameters used for the statistical clustering algorithm were the same as in Mahadevan and Connell's method: $\delta = 0.45$, $\rho = 2$, $\eta = 0.000001$, $k = 9$. Details of our implementation of the statistical clustering method can be found in [8]. The following parameters were common to both Q-learners:

It is significant that learning rates were declined separately for each behaviour. Each behaviour had the same initial learning rate, but as some behaviours are called more often than others and the learning rate decays through time, the learning rates at any one time will be different for each behaviour. The learning rate was decayed using the following equation:

$$\alpha_t = \alpha_0 \frac{1}{e^{M_b \alpha_s}} \quad (1)$$

Parameter	Meaning	Value
α_0	initial learning rate	0.2
α_δ	decay in learning rate	0.0003
α_{min}	minimum learning rate	0.01
p_0	initial Pr(random action)	0.9
p_δ	decay in p	0.0009
p_{min}	minimum value of p	0.1

where M_b is the number of time steps that behaviour had been active. The learning rate was never allowed to drop below α_{min} . During the exploration phase the probability of exploration commenced high and gradually declined, using an exponential law. The exploration method used was a semi-uniform distribution. The probability p of deviating from the greedy policy was initially .9 for each behaviour. The probability of taking a random action was declined according to:

$$p_t = p_0 \frac{1}{e^{M_\delta p_\delta}} \quad (2)$$

During the exploitation phase p was held constant at 0.1. The additional parameters for the $Q(\lambda)$ learner during the exploration phase were:

Parameter	Meaning	Value
λ_0	initial eligibility decay rate	1
λ_δ	decay in λ	0.0007
λ_{min}	minimum value of λ	0.3

During the exploitation phase λ was held constant at 0.3.