Adaptive Temporal Difference Learning of Spatial Memory in the Water Maze Task

Erik E. Stone, Marjorie Skubic, and James M. Keller

Abstract - **The Morris water maze task is a spatial memory task in which an association between cues from the environment and position must be learned in order to locate a hidden platform. This paper details the results of using a temporal difference (TD) learning approach to learn associations between perceptual states, which are discretized using a Self Organizing Map (SOM), and actions necessary for a robot to successfully locate the hidden platform in a "dry" version of the water maze task. Additionally, the adaptability of the temporal difference learning approach in non-stationary environments is explored.**

I. Introduction

Given the ability of animals to navigate and interact with the complex world around them, biological systems have been a source of much inspiration in the field of robotics. The inspiration for this work comes from a behavioral procedure originally designed by Richard Morris to study spatial learning in the rat called the Morris water maze [1]. In the typical Morris water maze experiment, a rat is placed into a circular pool of water from which the only escape is a raised platform. The raised platform is positioned just below the water's surface, and the water is made opaque to hide the platform from view of the rat. This ensures no local cues from the platform are used to guide the rat's behavior. Over multiple trials, normal rats learn to swim directly towards the hidden platform from any starting position around the edge of the pool, given the platform remains in a fixed location.

Much work has been done using the water maze to investigate spatial learning and memory tasks. Redish and Touretsky [2] used a simulated environment to evaluate a computational model of the hippocampus and how it allows rodents to solve the water maze task. In addition, Brown and Sharp [3] used a simulated water maze environment to investigate how spatial behavior could be guided by spatial information in the hippocampal formation.

Foster et al. [4] used temporal difference learning to model how hippocampal place cells might be used for spatial navigation by rats. First, they simulated a reward based navigational approach based solely on input from place cells. Second, they simulated a combined approach using input from place cells and information about the rats' self motion to acquire a goal independent coordinate system. Like Brown and Sharp [3], they used simulated place cells to provide a representation of the current position of the rat, as opposed to direct visual perceptual cues from the environment.

Krichmar et al. [5] constructed a dry version of the water maze task to assess the spatial memory of a brain-based device called Darwin X, whose behavior was guided by a simulated nervous system modeled on the anatomy and physiology of the vertebrate nervous system. A 16x14 foot rectangular room was used as the water area, with a hidden circular platform made of reflective paper. Darwin X was equipped with a color camera for vision, odometry for self-movement information, an IR sensor for platform detection, and IR sensors for obstacle avoidance.

Based on Krichmar's work, Busch et al. [6] used a simulated water maze environment to compare an attributed probabilistic graph search approach and a temporal difference learning approach based solely on visual cues from the environment. The simulated robot was equipped with three cameras to gather perceptual information from the environment and used a Self-Organizing Map (SOM) [7] to discretize the perceptual space.

The work described in this paper is based on extending the temporal difference learning navigational approach used by Busch et al. [6] from a simulated water maze environment into a physical "dry" water maze environment. Additionally, the adaptability of the temporal difference learning approach is investigated by moving the platform.

Section II of this paper describes the setup of the water maze task. Section III details how the robot gathers perceptual information from the environment, and how this information is discretized using an SOM. Section IV explains the details of the temporal difference learning system. Section V presents the results of two experiments that were carried out in the physical environment on a Pioneer P3-DX robot. Finally, section VI explores the adaptability of the temporal difference learning approach in non-stationary environments through simulation.

This work was supported in part by the U.S. National Science Foundation under Grant EIA-0325641 and DGE-0440524.

E. E. Stone, M. Skubic, and J. M. Keller are in the Electrical and Computer Engineering Department, University of Missouri, Columbia, MO 65211 USA (e-mail: ees6c6@mizzou.edu, skubicm@missouri.edu, kellerj@missouri.edu)

II. Water Maze Setup

A. Environment

A rectangular enclosure, modeled after the simulated environment from [6], is used for the "dry" water maze experiments. The enclosure is 5.26m by 6.06m, and contains a circular "hidden" platform that is 0.41m in diameter. Eighteen colored panels are arranged around the enclosure using approximately the same layout as in [6]. Fig. 1 shows pictures taken from within the enclosure.

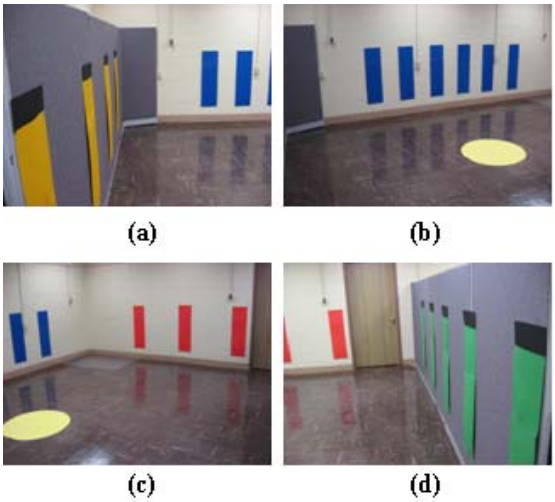

Fig. 1. Images taken of the enclosure. Images (a) through (d) were taken from left to right around the enclosure. The black paper mounted above the green and yellow panels helps in determining the top of the panels during segmentation.

Fig. 2. The Pioneer P3-DX robot used in the experiments. Two FireWire web cams are mounted on the front of the robot. The infrared sensor assembly can be seen beneath the front of the robot.

B. Robot

A pioneer P3-DX was used in the experiments. The robot is equipped with two FireWire web cams, seen in Fig. 2, with wide angle lenses. The cameras capture 640x480 color images. Each camera has an approximate horizontal field of view of 90 degrees before adjustment for lens distortion. Due to the adjustment, the cameras have an effective field of view of approximately 80 degrees. The robot's sonar sensors were used for basic obstacle avoidance.

Finally, the robot is equipped with an assembly of four infrared sensors, mounted beneath and towards the front of the robot, for detection of the hidden platform. All four of the sensors must be over the hidden platform for it to be detected.

III. Perception

A. Panel Detection

The panel detection process consists of first removing lens distortion from images captured from the web cams, filtering the images, and performing color segmentation. Only the top halves of the captured images are used for panel detection. This helps eliminate panel detection problems due to reflections from the shiny floor or the color of the hidden platform. Examples of the panel segmentation are shown in Fig. 3. Although the panel segmentation process is relatively robust, errors do occur do to limited camera resolution and inconsistent lighting conditions in the environment. Although these errors do not occur with great frequency, it is one source of error not present in simulation. The Open Source Computer Vision Library [10] is used for the image processing.

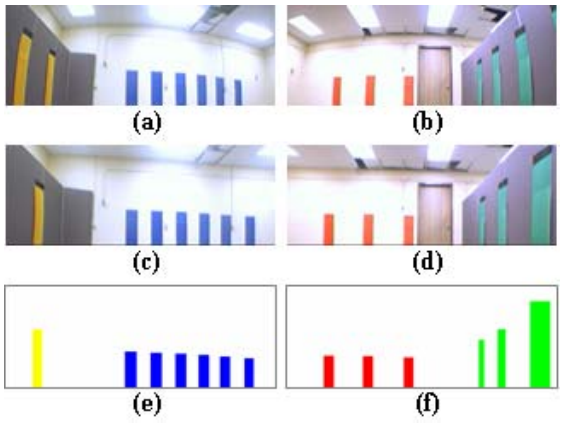

Fig. 3. An example of panel detection from the two FireWire web cams on the P3-DX robot. Images (a) and (b) are the top halves of raw images captured from the left and right cameras respectively. Images (c) and (d) are the corrected and filtered images from the left and right cameras respectively. Finally, images (e) and (f) show the resulting panel segmentation.

B. Discretization

In [6], a Self-Organizing Map (SOM) is used to discretize the perceptual space. The SOM allows the large number of possible perceptual states to be discretized into a useable number for the temporal difference learning system. The SOM(s) used in the experiments were trained from approximately 7,400 perceptual vectors collected by letting the robot randomly roam throughout the environment with an obstacle avoidance behavior. For the experiments described in Section V and VI, an 8x8 SOM was used.

Initially, the panel segmentation information collected from each camera is used to generate a feature vector. As each

camera has the possibility to encounter any of the 18 colored panels, each vector consists of 18 bins. If a panel of a certain color is detected, the height of that panel in pixels is stored in the first empty bin corresponding to the color of the panel, as the robot has no way to determine which of the panels it is observing. These feature vectors are then used to determine the current SOM node of the robot.

IV. Learning

In this spatial memory task, temporal difference (TD) learning [11] is used to learn an association between the discretized perceptual states (SOM nodes) and possible actions. TD learning is implemented using the Working Memory Toolkit (WMtk) [8] developed at Vanderbilt University, which is based on the biology of the prefrontal cortex and the midbrain dopamine system.

When the learning system is first initialized, the action preferences are set randomly. These preferences are then adjusted by the TD system based on rewards received during training episodes. A reward function is called at the end of each time step (after each move) and is defined as:

$$reward = \begin{cases} -5 & \text{if } c = 0 \\ -5 & \text{if obstacle detected} \\ 1 + \dfrac{(m-n)}{10} & \text{if goal detected} \\ -5 & \text{if } n \geq m \\ 0 & \text{otherwise} \end{cases}$$

where c is the current number of chunks in the learning system's store, m is the maximum number of moves allowed per episode, and n is the number of the current move in the current episode.

Thus, the robot is rewarded at the end of each training episode depending on whether it has or has not found the hidden platform. If the platform has been found, a positive reward inversely proportional to the number of moves required to find the platform is given. If the hidden platform has not been found, then a negative reward is given. In addition to the delayed rewards, the robot is given an immediate negative reward when the obstacle avoidance behavior is initiated, and when the learning system selects none of the five possible actions (c = 0). (When this occurs, an action is chosen at random.) These sparse measures of performance are the only feedback the robot receives.

Each training episode consists of a maximum of 51 moves. For each move the robot determines which SOM node it is currently in, and then selects one of five possible actions to take. The five possible actions are: hard left, left, forward, right, and hard right; each action is executed for one second. If the robot finds the goal before 51 moves have been executed, the run is ended. If the obstacle avoidance behavior is initiated, the robot rotates in place until its directional axis is 30 degrees beyond parallel with the wall (which is approximated using sonar readings), then starts a new move.

Finally, to evaluate the performance of the system, evaluation episodes are used. During these episodes exploration and learning are turned off such that the current action preferences of the system are always selected.

V. Experiments

Two experiments were developed for the physical environment to evaluate the performance of the TD system in learning the associations necessary to locate the hidden platform: a single corner experiment, and a four corner experiment.

A. Single Corner

In this experiment, the robot is allowed to train for a fixed number of episodes, at which point it is then evaluated. Each training episode consists of starting the robot at a single fixed starting location, which can be seen in Fig. 4. During each episode the robot is allowed up to 51 moves to locate the hidden platform. Once 51 moves have been executed, or the hidden platform has been found, the episode is ended. The robot is then repositioned to the single starting location for the next episode.

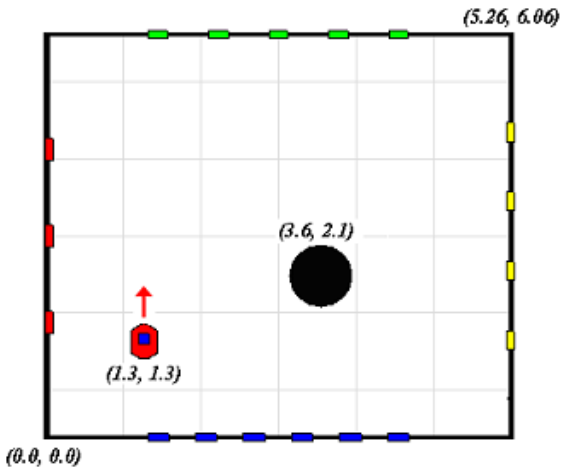

Fig. 4. Illustration of the single starting corner water maze task showing the starting location and heading of the P3-DX robot. Location coordinates are shown in meters.

The results of one single corner water maze experiment are shown in Fig. 5 and Table I, and are typical of those obtained during other tests. Fig. 5 (a) shows the number of moves per episode during a training sequence of 100 episodes. As can be seen, the robot fails to find the platform during most of the early training episodes. Fig. 5 (b) shows an example path of the robot during one of these early training episodes. Within approximately 20 training episodes, however, the robot appears to learn a path to the platform. Fig. 5 (c) shows the path of the robot during episode 22. As the training sequence continues the robot does fail to locate the platform during certain episodes. Fig. 5 (d) shows the path of the robot during episode 64, an episode in which it fails to find the platform. The path in Fig. 5 (d) is typical of many failed runs later in the

training process. The robot generally follows a successful path to the platform but just misses it. More information is needed to identify the exact reason. However, these failures could be due to the exploration of the TD learning system, or could be caused by some combination of parameters such as goal size, SOM size, sensory uncertainties, etc. Clearly, though, the frequency of episodes during which the platform is not found decreases as the training process progresses.

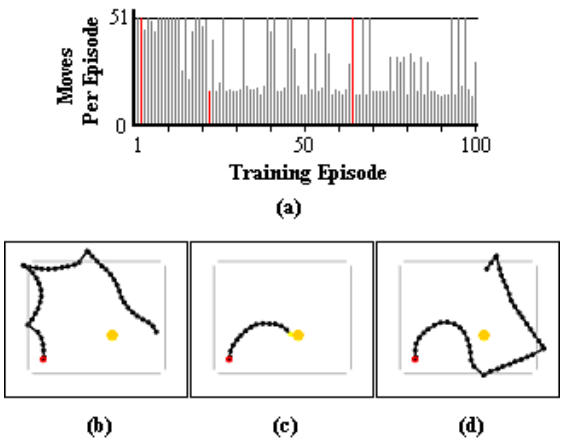

Fig. 5. (a) Plot of moves per episode during 100 training episodes for the typical single corner water maze task presented here. (b) Path of robot during episode 2. (c) Path of robot during episode 22. (d) Path of robot during episode 64. The displayed paths are highlighted in red in (a). The inner box in (b), (c), and (d) shows the distance at which obstacle avoidance is activated. The robot's path is logged using odometry during training and evaluation episodes. This information is not used by the learning system.

TABLE I
SINGLE CORNER EVALUATION RESULTS

	Before Training	*After 100 Training Episodes*
Evaluation Episodes	40	40
Average Moves per Episode	48.5	23.6
Episodes Platform not Found	32	8

To determine improvement, the robot was evaluated both before and after the training episodes. Table I shows the results of these evaluations. Noticeable improvement can be seen after the 100 training episodes as compared with the results obtained before training.

B. Four Corner

In this experiment, the robot is again allowed to train for a fixed number of episodes, at which point it is evaluated. Each training episode consists of starting the robot at one of four starting locations, which can be seen in Fig. 6. The four starting locations are visited in the following sequence: lower left, lower right, upper right, and upper left. During each episode the robot is allowed up to 51 moves to locate the hidden platform. Once 51 moves have been executed, or the hidden platform is found, the episode is ended. At the end of each episode the robot is repositioned to the next starting location in the sequence described above.

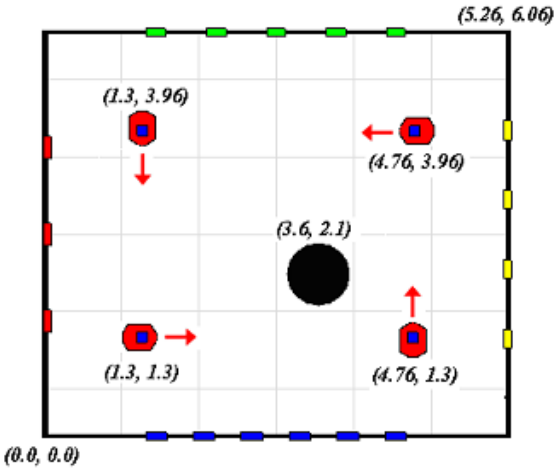

Fig. 6. Illustration of the four starting corner water maze task showing starting locations and headings of the P3-DX robot. During training and evaluation, the starting locations were visited in the following sequence: lower left, lower right, upper right, and upper left. This sequence was repeated until the desired number of episodes was completed. Location coordinates are shown in meters.

The results of one four corner experiment are shown in Fig. 7 and Table II, and are typical of those obtained during other tests. Fig. 7 (a) shows a moving average of the number of moves per episode during a training sequence of 100 episodes. The average is taken over the current episode and the previous three. A moving average is used to display the change over all four starting locations.

As in the single corner task, the robot fails to find the platform during many of the early training episodes. Fig. 7 (b) shows the paths of the robot from each starting location for episodes 2-5. After approximately 70 training episodes, the robot appears to learn a path to the platform from each of the starting locations. Fig. 7 (c) shows the paths of the robot from each starting location for episodes 69-72. Here again, however, as the training sequence continues the robot does fail to locate the platform during certain episodes. Fig. 7 (d) shows the paths of the robot from each starting location for episodes 91-94, in which the robot fails to locate the platform from the lower right starting location during episode 94.

As in the previous task, the robot was evaluated before and after the training episodes to determine improvement. For the four corner task, the evaluation episodes are carried out from all four starting locations in the same sequence as during training. The evaluation consists of 40 total episodes, thus 10 evaluation episodes are conducted from each corner. Table II shows the results. Here again, noticeable improvement can be seen after 100 training episodes as compared with the results before training.

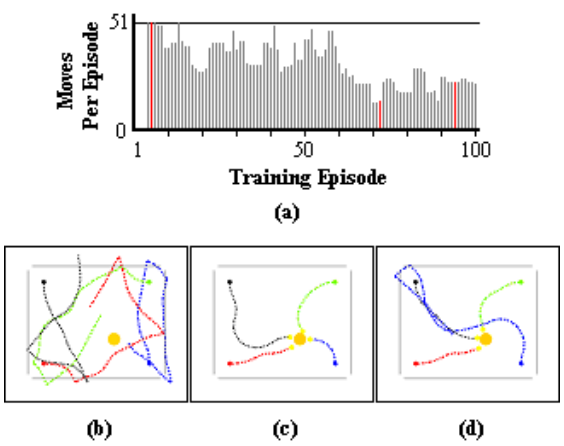

Fig. 7. (a) Plot of the sliding window average of moves per episode during 100 training episodes for the typical four corner water maze task presented here. (b) Paths of robot during episodes 2-5. (c) Paths of robot during episodes 69-72. (d) Paths of robot during episodes 91-94. The average of the displayed paths is highlighted in red in (a). The inner box in (b), (c), and (d) shows the distance at which obstacle avoidance is activated. The robot's path is logged using odometry during training and evaluation episodes. This information is not used by the learning system.

TABLE II
FOUR CORNER EVALUATION RESULTS

	Before Training	*After 100 Training Episodes*
Evaluation Episodes	40	40
Average Moves per Episode	46.0	20.1
Episodes Platform not Found	30	3

C. Physical vs. Simulation

Although environmental differences make a direct comparison of moves per episode in the simulated and physical environments difficult, results obtained with the physical robot are similar to results obtained in simulation, on the same experiments, in terms of the number of training episodes required for the robot to learn a path to the hidden platform (both for one or four corners).

VI. NON-STATIONARY ENVIRONMENTS

One of the main motivations for using the TD learning approach is the ability for online learning and adaptability of the system to non-stationary environments. However, the experiments reported thus far have all involved a stationary environment. Therefore, experiments using the four corner task were conducted in simulation, using the setup from [6], to test the adaptability of the TD approach.

A. Initial Testing

Fig. 8 (a) shows the results, averaged over 10 trials, of allowing the robot to train for 400 episodes with the platform at an initial starting location, moving the platform to a new location, and then allowing the robot to train for 400 additional episodes. As can been seen from the graph, the robot does learn to find the platform at the new location; however, its performance is noticeably worse after 400 training episodes as compared to the performance obtained with the original location. This degraded performance is explained by the fact that the paths the robot learns to the new location almost always build from the paths the robot learned to the original location. Thus, the robot learns paths to the new location that, generally, first pass through the original location. As Fig. 8 (a) illustrates, the learning which takes place during the initial 400 episodes seems to hinder the ability of the TD system to fully adapt, learn optimum paths, to the new location. This result is further illustrated in Fig. 8 (b), in which the platform location was changed twice during a training sequence.

Fig. 8 (c) shows the results of extending the experiment conducted for Fig. 8 (a) for 400 additional episodes. Specifically, the platform was moved back to where it originally started after episode 800. As can be seen from the graph, there is not a large spike in the number of moves per episode when the platform is moved back to where it originally started, as the paths the robot learned to the new platform location generally pass through the original platform location. Although this indicates that a significant amount of what was learned during the initial 400 episodes was retained, it also supports the idea that an inability to "forget" what was previously learned significantly hinders the ability of the TD system to truly adapt in non-stationary environments.

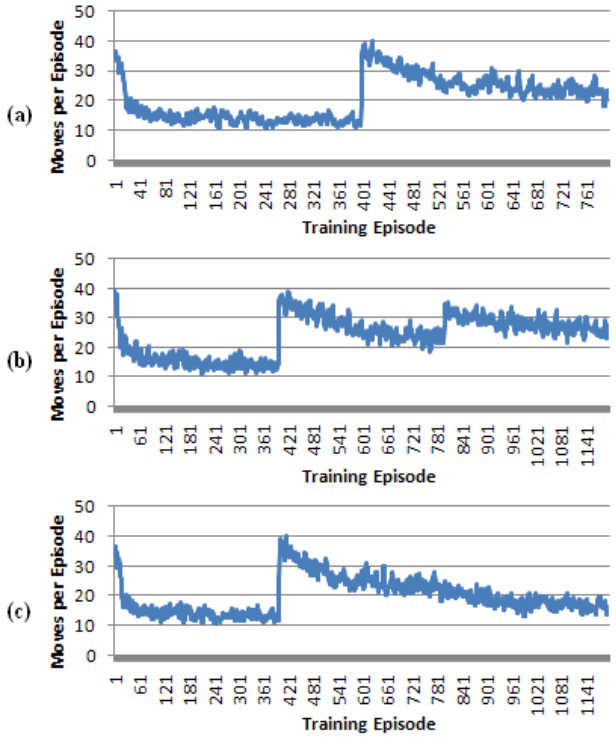

Fig. 8. (a) Plot of moves per episode over 800 training episodes during which the platform was moved after episode 400. (b) Plot of moves per episode over 1200 training episodes during which the platform was moved after episode 400 and episode 800. (c) Plot of moves per episode over 1200 training episodes during which the platform was moved after episode 400, and then moved back to where it originally started after episode 800. Results were averaged over 10 trials in (a), (b), and (c).

B. *Forgetting*

In order to obtain better adaptability, the effect of adding "forgetting" to the TD system was explored, with the main goal being to allow the system to perform better in non-stationary environments by reducing the hindrance of prior learning. The "forgetting" is based on keeping track of past performance, namely rewards received.

Specifically, a "short term" reward and a "long term" reward are calculated based on rewards received during the current trial. These two rewards are then used to control the "forgetting" process.

The original update equation for the weights, w_i, of the TD system [8] is given as:

$$w_i^{t+1} = w_i^t + \alpha \delta^t e_i^t$$

where α is the learning rate, δ^t is the TD error at time t, and e_i^t is the eligibility trace for w_i at time t. With "forgetting," the update equation becomes:

$$w_i^{t+1} = \left[w_i^t + \alpha \delta^t e_i^t - \mu \right] \left(1 - f_p\right) + \mu$$

where μ is the initial mean of the weights for the TD system, and f_p is defined as follows:

$$f_p = \begin{cases} 0 & if\ r_s \geq r_l - \varepsilon \\ (r_l - r_s)\eta_f & otherwise \end{cases}$$

Where r_l and r_s are the long term and short term rewards respectively, ε is a small positive constant, and η_f is the forgetting rate.

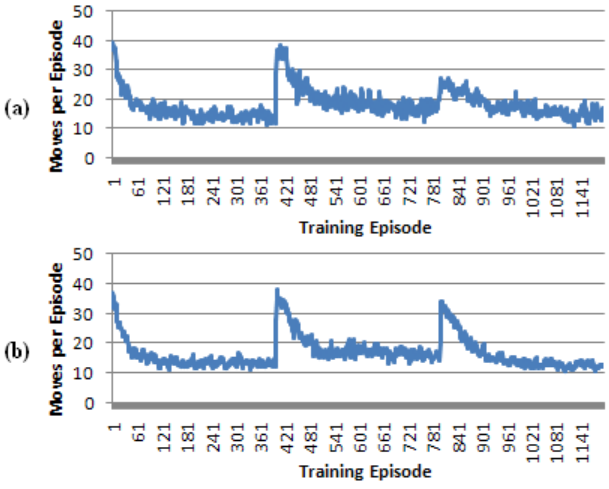

Fig. 9. Performance with "forgetting" added to TD learning system. (a) Plot of moves per episode over 1200 training episodes during which the platform was moved after episode 400 and episode 800. (b) Plot of moves per episode over 1200 training episodes during which the platform was moved after episode 400, and then moved back to where it originally started after episode 800. Results were averaged over 10 trials in (a) and (b).

Under normal circumstances, where $r_s \geq r_l - \varepsilon$, the weight update is unchanged. However, if $r_s < r_l - \varepsilon$, then the system effectively "forgets" (by an amount proportional to the difference between r_l and r_s) by moving all the weights closer to the initial mean value.

Fig. 9 (a-b) shows the same experiments conducted in Fig. 8 (b-c), with the addition of "forgetting" to the TD system. As can be seen from the results, with "forgetting," the TD system is effectively able to adapt to the new platform locations as though it had not undergone any previous learning, thus making it much more adaptable in non-stationary environments.

VII. Conclusion

This paper presented the results of using a biologically inspired TD learning approach to learn a spatial memory task on a physical robot, and the results of testing the adaptability of that approach to non-stationary environments. Experiments showed that the robot is able, using the TD approach, to learn the necessary associations between perceptual states and actions to successfully locate the hidden platform. Furthermore, experiments conducted in simulation showed that with the addition of "forgetting" the system is able to achieve good performance in non-stationary environments.

Future work will be aimed at further investigation of the adaptability of the TD approach, and investigation of other representations of the perceptual space.

References

[1] Morris, R. "Development of a water-maze procedure for studying spatial learning in the rat," *Journal of Neuroscience Methods,* 1984, vol. 11, pp. 47-60.

[2] A.D. Redish and D.S. Touretsky, "The role of the hippocampus in solving the Morris water maze," *Neural Computation*, 1998, vol. 10, no. 1, pp. 73-111.

[3] M.A. Brown, and P.E. Sharp, "Simulation of spatial learning in the Morris water maze by a neural network model of the hippocampal formation and nucleus accumbens," *Hippocampus*, 1995, vol. 5, pp. 171-188.

[4] D.J. Foster, R.G.M. Morris, and Peter Dayan, "A Model of Hippocampally Dependent Navigation, Using the Temporal Difference Learning Rule," *Hippocampus*, 2000, vol. 10, pp. 1-16.

[5] J.L. Krichmar, D. A. Nitz, J.A. Gally, and G. M. Edelman, "Characterizing functional hippocampal pathways in a brain-based device as it solves a spatial memory task," In *Proc National Academy of Science USA*, 2005, vol. 102, pp. 2111-2116.

[6] M.A. Busch, M. Skubic, J.M. Keller, and K.E. Stone, "A Robot in a Water Maze: Learning a Spatial Memory Task," In *2007 IEEE Intl. Conf. on Robotics and Automation*, Rome, Italy, 10-14 April 2007, pp. 1727-1732.

[7] Kohonen, T. 1990. The self-organizing map. *Proc. IEEE* 78, 1464-1480.

[8] J.L. Phillips and D.C. Noelle, "A Biologically Inspired Working Memory Framework for Robots," In *Proc. of the 27th Annual Meeting of the Cognitive Science Society*, Stresa, Italy, July 2005.

[9] Gerkey, B.P., Vaughan, R.T. and Howard, A. 2003. The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In *Proc. IEEE Intl. Conf. Advanced Robotics*, Coimbra, Portugal.

[10] G. Bradski, et. al. Intel Open Source Computer Vision Library: http://www.intel.com/technology/computing/opencv/

[11] Sutton, R.S. "Learning to predict by the methods of temporal differences," *Machine Learning*, 1988, vol. 3, pp. 9-44.