# Learning Obstacle Avoidance with an Operant Behavior Model

D. A. Gutnisky
B. S. Zanutto*
Instituto de Ingeniería de Biomédica
FI-Universidad de Buenos Aires
Paseo Colón 850
CP 1063, Buenos Aires
Argentina
silvano@fi.uba.ar
and
Instituto de Biología y Medicina Experimental—CONICET

**Keywords**
Operant learning, neural networks, reinforcement learning, Q-Learning, animats, artificial neural networks

**Abstract** Artificial intelligence researchers have been attracted by the idea of having robots learn how to accomplish a task, rather than being told explicitly. Reinforcement learning has been proposed as an appealing framework to be used in controlling mobile agents. Robot learning research, as well as research in biological systems, face many similar problems in order to display high flexibility in performing a variety of tasks. In this work, the controlling of a vehicle in an avoidance task by a previously developed operant learning model (a form of animal learning) is studied. An environment in which a mobile robot with proximity sensors has to minimize the punishment for colliding against obstacles is simulated. The results were compared with the Q-Learning algorithm, and the proposed model had better performance. In this way a new artificial intelligence agent inspired by neurobiology, psychology, and ethology research is proposed.

## 1 Introduction

Much of the research on autonomous robots is focused on solving a relatively small variety of problems, providing in this way a common framework to compare how different strategies are able to solve them. Obstacle avoidance is one of the most common tasks in which new algorithms are compared with known ones. Some of them, because of their simplicity, robustness, and mathematical tests of convergence, are used as benchmarks. A good example is Q-Learning [28, 34], a reinforcement learning algorithm, whose application varies from box-pushing [21] to elevator dispatching tasks [6].

Reinforcement learning is the problem faced by an agent that must learn behavior through trial-and-error interaction with a dynamic environment [18]. It relies on the association between a goal and a scalar signal, which can be either a reward or a punishment. The objective is to find the policy that maximizes rewards (or minimizes punishments). In the particular case of obstacle avoidance, when a mobile robot contacts an obstacle, it is punished.

Algorithms such as Q-Learning are not restricted by biological constraints and are built with the aim of solving different kinds of problems. Touretzky and Saksida [31] said that mobile robots trained by methods such as Q-Learning have not come close to matching the sophistication, versatility, and adaptation of animals. They suggested that closer attention to the animal training literature and a serious attempt to model the effects described there might yield benefits of immediate value to robot learning

∗ Requests for reprints should be send to Dr. Zanutto.

researchers, and also provide a new, computationally oriented perspective on animal learning.

On the other hand, simple artificial animals (*animats*), which operate as autonomous adaptive robots in the real world, can serve both as models of biological behavior and as a radical alternative to conventional methods of designing intelligent systems [9]. According to McFarland and Bösser [22], it has been recently realized that robots might better be designed along the zoological lines of primitive animals than along the traditional lines of autocratic control. Dean [10] says that the animat approach is the most recent attempt to simulate the adaptive behavior characteristic of animals as well as the acquisition of this competence. Moreover, he points out that both in analysis and design, the animat approach borrows heavily from ethology, psychology, neurobiology, and evolutionary biology. Animat studies can also be useful for studying emergent behavior in biological systems [30]. For artificial intelligence (AI) and robotics researchers, understanding the mechanisms behind adaptive behavior is secondary to creating them, but natural scientists can hope for tools and concepts to aid understanding of biological systems. It is clear that if we knew how animals controlled their behavior, this might provide us new ideas about how to make robots do it.

In line with this proposed approach, recent research showed how models inspired by psychological research can be used to control mobile robots [5, 14, 32]. The mobile robot Mavin [2] was one of the first attempts to control a mobile robot based on observations derived from the behavioral literature.

Here a previously developed model of operant learning capable of avoiding obstacles is proposed to control agents. It differs from other models in its behavioral and neurobiological bases and in the psychological experiments it explains. This article is a new contribution to the increasing interest in building biologically plausible models for application to robot research.

We have investigated the capability of the model of operant behavior developed by Lew et al. and Zanutto and Lew [20, 35] to learn obstacle avoidance. The hypotheses of this adaptive neural network model of aversive and appetitive behavior come from theories of behavior, experimental results on animals, and neurobiological evidence. Its performance is compared with Q-Learning, the algorithm most used in AI to perform this task.

## 2 Operant Conditioning

Psychologists have identified operant conditioning as a primary mechanism that enables animals to acquire relevant characteristics of their environment in order to get rewards or to avoid punishments. Operant conditioning is a closed-loop experimental procedure in the sense that stimuli received by the animal are contingent upon its behavior. The animal learns to perform the actions that lead to a reward more frequently and the ones that lead to a punishment less frequently. For example, a rat can be trained to press a key when it sees a red light as conditioned stimulus (CS) in order to receive a food reward (unconditioned stimulus, or US).

Our operant conditioning model is able to learn from both appetitive and aversive stimuli. In the experiment of obstacle avoidance, the animat learns to avoid collision, which is interpreted as a punishment.

## 3 Model of Operant Conditioning

Psychological experiments suggest that behavior is driven by changes in the expectation about the future salient events, mainly reward and punishment. In operant and classical conditioning, the CS anticipates the US. Rescorla and Wagner [23] proposed that animals learn by comparing what they expect from a given situation with what actually happens. As Staddon [29] has pointed out, animals act because the CS allows them to elaborate an expectation or prediction of the unconditioned stimulus. Furthermore, there are neural substrates of prediction and reward, such as the involvement of dopamine neurons of the ventral tegmental area (VTA) and substantia nigra, identified with the processing of prediction and reward [25].

From this point of view, Lew et al. and Zanutto and Lew [20, 35] presented a neural network model (ZL) that, based on biologically plausible hypotheses, explains relevant features of operant conditioning for appetitive and aversive stimuli. In [20] it was shown that the model predicts features of appetitive stimulus such as the matching law, response selection, the partial reinforcement extinction effect, spontaneous recovery, and the successive contrast effect. In [35] it was shown how the model explains experiments to support the one-factor theory, the two-factor theory, and the cognitive theory. Finally, this model also explains imitation in the same terms as Schmajuk and Zanutto [24].

The model is shown in Figure 1. The inputs to the model are all the CSs and the US. The outputs are all the possible responses of the animal (Rs). The network has three functional blocks that simulate the stimuli and responses: the short-term memory, the prediction neurons, and the response neurons.

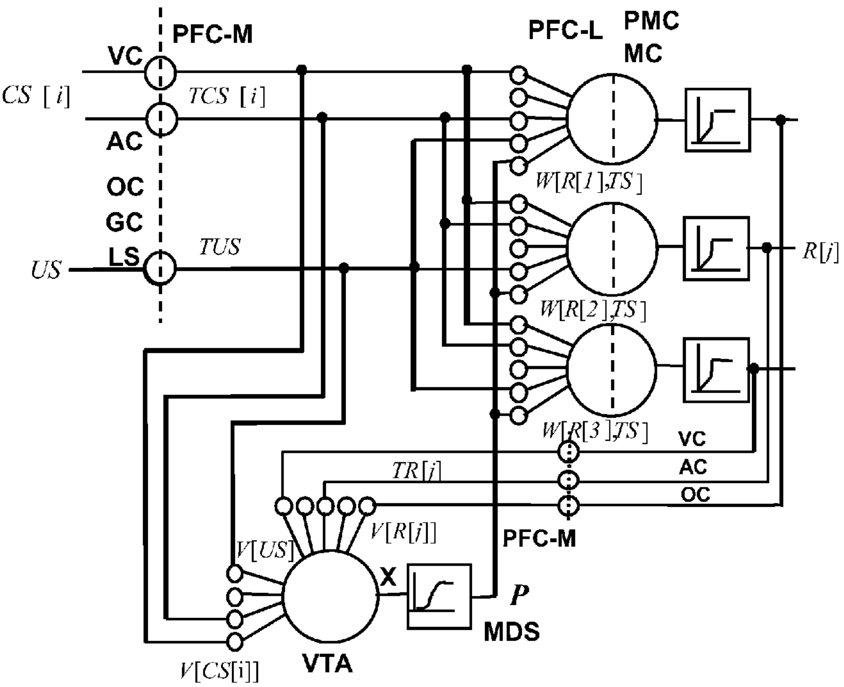

Figure 1. Neural network model. There is an artificial neuron computing the prediction *P*, and one for each response *R*[*j*]. Here *P* is a nonlinear function applied to the output of the prediction neuron (*X*). VC: visual cortex; AC: auditory cortex; OC: olfatory cortex; GC: gustatory cortex; LS: limbic system; PFC: prefrontal cortex; PMC: premotor cortex; MC: primary motor cortex; MDS: mesocortical dopaminergic system; VTA: ventral tegmental area. The traces (TCS[*i*]) representing the short-term memory of the conditioned stimuli are computed by a group of cells in the PFC-M [15]; the unconditioned stimulus (TUS), and the responses (TR[*j*]) are inputs of the neurons computing *P* and *R*[*j*]. The synaptic weights *V*[CS[*i*]], *V*[*R*[*j*]], and *V*[US] represent the associations between the inputs and *P*. The synaptic weights *W*[*R*[*j*], TS] are associations between *P*, TCS[*i*] and TUS, and the PFC-L [4]. The responses performed by the animal are inputs to the different sensory cortices (VC, AC). These responses generate short-term memory in the PFC-M (bottom of the figure). Finally, the traces from response neurons are feedback to the prediction neuron.

In the model, all the neurons make a summation of their inputs weighted by their synaptic weights, and then a nonlinear function is applied, as is commonly done in neural network models. In addition, synaptic weights are limited in their maximum strength due to biological constraints.

A single artificial neuron computes the role of dopamine neurons of the VTA and substantia nigra (SN) in prediction and reward. The prediction neuron has all the stimuli and the response traces as inputs. The prediction neuron synaptic weights are modified by the Rescorla-Wagner rule, except the one that corresponds to the US, which remains fixed. The response neurons have all the CS and US traces and the prediction as inputs. If the prediction exceeds a certain threshold, the learning in the responses will be Hebbian in the appetitive, and anti-Hebbian in the aversive case. The reverse rule is applied if the prediction is below the threshold. When one of the response neurons exceeds a certain level, the associated response is executed.

The hypothesis that dopamine can mediate learning in motor association and primary motor areas was also incorporated in the operant learning model of Donahoe et al. [11–13]. However, this last model does not take into account that dopamine neurons in the VTA and SN signal the difference between actual and predicted rewards [25]. The Rescorla-Wagner rule states that the discrepancy, or error, between the actual and predicted rewards determines whether learning occurs when a stimulus is paired with a reward. The role of prediction errors is demonstrated by the observation that learning is blocked when the stimulus is paired with a fully predicted reward. Waelti et al. [33] used this blocking procedure to show that the responses of dopamine neurons to conditioned stimuli are governed differentially by the occurrence of reward prediction errors rather than stimulus-reward associations alone. Daly and Daly [7, 8] incorporated the major predictions of frustration theory [1] in the Rescorla-Wagner model. In contrast with the operant theory presented in [20, 35], Daly and Daly's model does not explicitly determine which response is emitted given the associative strength at a particular moment in time. As explained above, the operant model used here has a neuron computing the prediction, whose learning is a real-time version of the Rescorla-Wagner rule. As in [11], the simulated dopaminergic system in the VTA and SN controls the learning of the model's responses, although it differs in the learning rule applied.

There are two main approaches to making formal theories of behavior based on neuroscience research. The first one consists of explaining behavior from realistic models of biological neurons and synapses. This approach has not been able to explain the most basic conditioning paradigms, due to the complexity of the mammals' nervous systems. The second one starts explaining behavior by building models at a higher level of abstraction constrained by anatomical, physiological, and neurobiological evidence. Although this approach sometimes lacks details at the cellular level, it is good for testing hypotheses that can be verified by the animal's behavior. Once the model is able to explain most relevant behavioral results without contradicting biological evidence, higher levels of abstraction can be replaced with more concrete ones. The model used here follows this latter approach, in which artificial neurons represent clusters of biological neurons.

### 3.1 Traces of Stimulus Computation

There are two types of traces, one corresponding to stimuli, and the other to responses. In both cases, the traces represent a short-term memory [15]. Short-term memories allow associating a CS with a US when their presentations are not simultaneous.

The stimuli traces receive all the stimuli coming from the visual cortex (VC), auditory cortex (AC), olfactory cortex (OC), gustatory cortex (GC), and limbic system (LS) as inputs. The outputs of short-term traces are inputs to response neurons and to the prediction, and the traces from response neurons are feedback to the prediction neuron.

The equation to calculate the short-term traces ($T_S$) of the stimuli ($S$) of the CS as well as the US at instant $n$ is a first-order linear difference equation:

$$T_S(n) = T_S(n-1)\cdot(1-\varepsilon) + \varepsilon\cdot S(n) \qquad \text{if} \quad S(n) > 0 \tag{1}$$

$$T_S(n) = T_S(n-1)\cdot(1-\beta) \qquad \text{if} \quad S(n) = 0 \tag{2}$$

When a stimulus is presented ($S_{n-1} > 0$), its short-term trace raises exponentially at a rate determined by $\varepsilon$, up to a maximum equal to the stimulus value. When the stimulus disappears ($S_{n-1} = 0$), its short-term trace decays exponentially at a rate determined by $\beta$.

There is a similar equation for the response traces:

$$T_{\mathrm{R}}(n) = T_{\mathrm{R}}(n-1)\cdot(1-\beta) + \varepsilon\cdot(1 - T_{\mathrm{R}}(n-1))\cdot \mathrm{R}(n) \tag{3}$$

### 3.2 Prediction Computation

The inputs to the prediction are all the short-time traces of the CSs, USs, and Rs. Each response neuron has the output of the prediction neuron ($P$) as input. The prediction neuron has the additional function of controlling the response neurons' learning:

$$X(n) = V_{\mathrm{US}}(n)\cdot T_{\mathrm{US}}(n) + \sum_{i=1}^{N_{\mathrm{CS}}} V_{\mathrm{CS}i}(n)\cdot T_{\mathrm{CS}i}(n) + \sum_{i=1}^{N_R} V_{\mathrm{Ri}}(n)\cdot T_{\mathrm{Ri}}(n) \tag{4}$$

$$P(n) = \frac{\xi}{1 + e^{-\upsilon\cdot(X(n)-\sigma)}} \tag{5}$$

Here $P$ is the nonlinear output function, $V$ the synaptic weights, and $T$ the corresponding traces of the US, CS, and R. The number of CSs is $N_{\mathrm{CS}}$, and the number of responses is $N_{\mathrm{R}}$. The synaptic weight $V_{\mathrm{US}}(n)$ remains fixed at 0.2. The maximum value of $P$ is $\xi$; when $X_n = \sigma$, $P$ reaches half of its maximum value. The slope of $P$ is controlled by $\upsilon$.

The updating of the weight in the prediction neuron is based on the Rescorla-Wagner model [23]:

$$\mathrm{VX}_S(n) = \mathrm{VX}_S(n-1) + \eta(\mathrm{US})\cdot T_S(n)\cdot(\mathrm{US}(n) - X(n)) \tag{6}$$

$$V_S(n) = \frac{2}{1 + e^{-\kappa\cdot \mathrm{VX}_s(n)}} - 1 \tag{7}$$

The associative strength is represented by $\mathrm{VX}_S(n)$. Equation 7 clamps the synaptic weights in the range from $-1$ to 1. The VX values are bounded between 10 and $-10$ in order to limit the maximum associative strength. Here the salience (i.e., the relative importance of a stimulus) is represented by the stimulus trace $T_S(n)$; this means that a CS's salience depends on its memory trace. The rate of learning is represented by $\eta(\mathrm{US})$, which depends on whether the US is present or not, due to attentional modulation: $\eta(\mathrm{US}) = \eta_i$ if $\mathrm{US} > 0$, and $\eta(\mathrm{US}) = \eta_d$ if $\mathrm{US} = 0$.

### 3.3 Computation of Responses

There is an output neuron for each of the possible responses of the animal. The output of these neurons is determined by

$$Y_j(n) = W_{j\mathrm{Pred}}(n)\cdot P(n) + W_{j\mathrm{US}}(n)\cdot T_{\mathrm{US}}(n) + \sum_{i=1}^{N_{\mathrm{CS}}} W_{j\mathrm{CS}i}(n)\cdot T_{\mathrm{CS}i}(n) + \mathrm{noise}(n) \tag{8}$$

$$R_j(n) = \begin{cases} 0 & \text{if } Y_j(n) < 0 \\ Y_j(n) & \text{if } 0 \leq Y_j(n) \leq \mu \\ 1 & \text{if } Y_j(n) > \mu \end{cases}$$

where $W_{j\text{Pred}}(n)$ is the $j$th response synaptic weight corresponding to the output of the prediction neuron at instant $n$, $W_{j\text{US}}(n)$ is the one corresponding to the US, and $W_{j\text{CS}i}(n)$ is the one corresponding to each of the CSs. The output of the prediction neuron is $P_n$, the US's short-term memory is $T_{\text{US}}(n)$, and the CS's short-term memories are $T_{\text{CS}i}(n)$.

The animal executes a response R($j$) whenever Y($j$) exceeds the threshold $\mu$. At any instant, only one response can be executed. The updating is done asynchronously. At each instant, one neuron is selected at random, and only its weights are updated.

The equation to compute the learning of these neurons is based on the Hebb rule [16], which states that the change in synaptic efficacy is proportional to the presynaptic and postsynaptic activity. If the US is predicted, the learning will be Hebbian in the appetitive case. The association between stimuli and the selected response will be reinforced because it produces a positive reinforcement. The weights of the executed neuron are updated in the following way:

$$W_{jq}(n) = \Psi \cdot W_{jq}(n-1) + \phi \cdot \Omega \cdot Q(n) \cdot T_{\text{R}j}(n) \tag{9}$$

where $Q$ is the corresponding input ($P$, $T_{\text{US}}$, or $T_{\text{CS}i}$), $q$ is the respective index ($P$, US, or $i$), and $T_{\text{R}j}(n-1)$ is the short-term trace of the input $j$th response at instant $n-1$. The coefficient $\Psi$ is a constant that represents the synaptic weight forgetting rate. The learning rate is controlled by $\phi$ and $\Omega$. However, $\Omega$ can take two possible values: when the US is appetitive and $P < \lambda$, then $\Omega = -\lambda$, and if $P \geq \lambda$ then $\Omega = \lambda$. The reverse rule is applied in the case that the US is aversive. The constant $\lambda$ is the learning threshold; if the prediction is high enough, it means that the active CSs will signal that a US is likely to come; if the US is aversive, the responses that do not avoid the punishment will be weakened.

## 4 Obstacle Avoidance with the Model of Operant Behavior

### 4.1 Introduction

We developed an application to simulate an environment with a number of obstacles in which a mobile robot with proximity sensors must avoid them. The inputs to the agent are five digital sensors and a signal indicating if the mobile robot has collided or not. The outputs are three possible actions: to move forward, to turn right 45°, and to turn left 45°. The control of the mobile vehicle can be a computational implementation of our model of operant behavior (ZL's model) or a Q-Learning one. We tested both models in the same environment and compared their ability to learn how to avoid obstacles.

### 4.2 Description of the Simulator's Computational Implementation

The virtual environment is a 200 cm square with eight obstacles and a labyrinth. The simulated mobile robot is a 10 cm square. There are three equidistant IR sensors in front of the mobile vehicle and two sensors on each side (see Figure 2). The simulated IR sensors measure the amount of IR light reflected back from the obstacles inside a bidimensional vision cone. Whenever the received energy is above a certain threshold, the sensor returns a logic level of 1, and otherwise 0. In order to compute the energy received by each sensor, the intersections between the obstacles and the limits of the

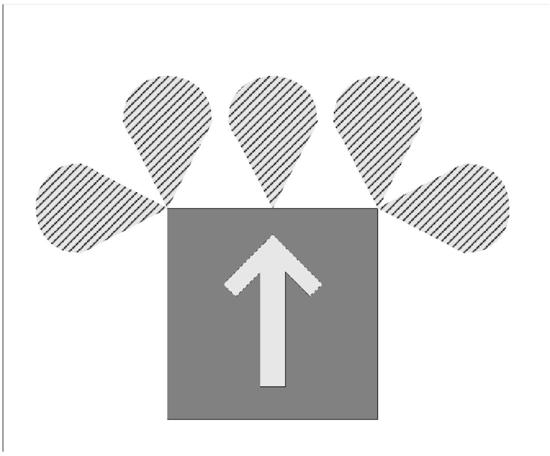

Figure 2. The mobile vehicle sensors' distribution and their cone of vision. There are five IR sensors in the front of the mobile: three facing forward, and the other two facing to the left and the right. The figure is out of scale.

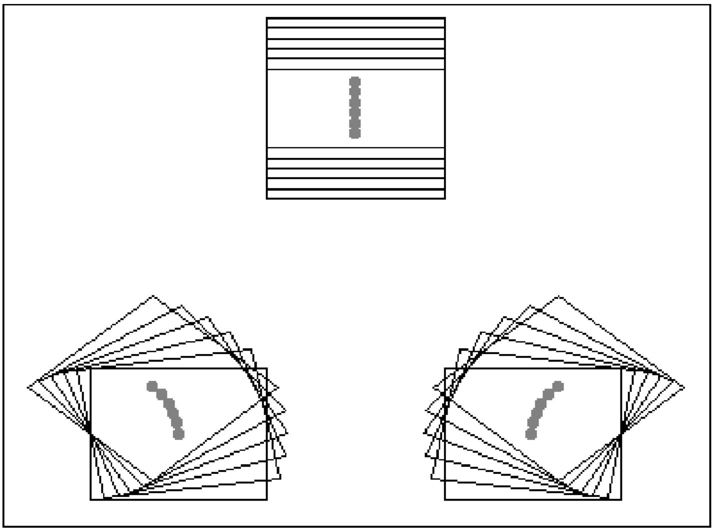

Figure 3. The three allowed mobile vehicle moves, each one involving a displacement of the mass center.

vision cone are calculated. Then, each segment is integrated with respect to the inverse square of the distance. The reach of the sensors is 10 cm.

The agent executes one of the allowed actions, to move forward, to turn left, or to turn right. Both turns involve a displacement of the mobile robot's center of mass (see Figure 3). Once the action is decided upon and performed, the sensor's information does not change, and the agent cannot make another response until the movement is completed. Then the agent must perform another action.

One of the purposes of this article is to show how the model of operant conditioning is able to avoid obstacles, given digital IR sensors as input. However, learning complex rules, as in this case, involves other capacities beyond operant ones. An operant behavior theory need only explain the experimental results of operant learning. In this case, it seems unlikely that an animal in an operant box having the sensor information as CSs could learn what to do to avoid a punishment. That the operant model used here was not able to avoid obstacles with the raw sensor information is right for the theory, but does not accomplish the proposed task. However, this does not mean that operant behavior is not involved in the learning of obstacle avoidance, but that higher-level visual processing has to be used to simulate when animals are in the context of avoiding obstacles. We do not focus on studying how the visual information that arrives at association areas (i.e., the prefrontal cortex and premotor cortex) is processed. Instead, we are interested in how an action is selected based on higher-level information that arrives at association areas and the reinforcement that is obtained.

To address this inconvenience, we had to add a preprocessing layer between the sensor information and the input to the agent. No direct information from the sensors is provided to the agents. Instead, we provided the agent with three different signals as CSs (see Figure 4) that indicate in which direction there is a free path, and another CS representing the context (CX) that is present all the time. Three circumstances can arise. The first one is when there is only one direction without near obstacles. The second one is when close obstacles surround the agent and there is no clear free path. In this case, the only active signal will be CX. The third one is when there is more than one direction without near obstacles. In this last case, one of the alternatives is selected by default. The selection of a default direction did not have any influence on the performance of any agent.

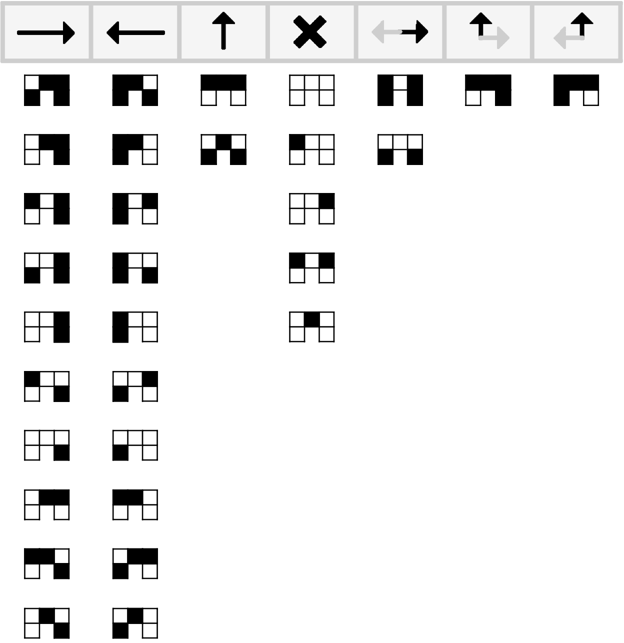

Figure 4. Free-path CSs for all the possible combination of sensor states. Empty squares represent sensors detecting an obstacle. The arrows in each column represent the free-path direction provided to the agents for the sensor states described below them. The cross represents the cases in which there is no free path (no signal is provided to the agents). The columns headed by more than one arrow correspond to the cases that have more than one free path. In these cases only one of the possible directions is provided to the agents. Gray arrows indicate the direction signal selected by default. In the case that no sensor is detecting any obstacle, no signal is provided. These arbitrary selections did not influence the agents' performance. Performing a move in the direction of the free path does not guarantee avoiding a collision. A collision can happen if, despite taking the correct action to avoid an obstacle, the mobile robot is too close to it. In addition, performing a move in a direction different from the free-path signal may not lead to a collision.

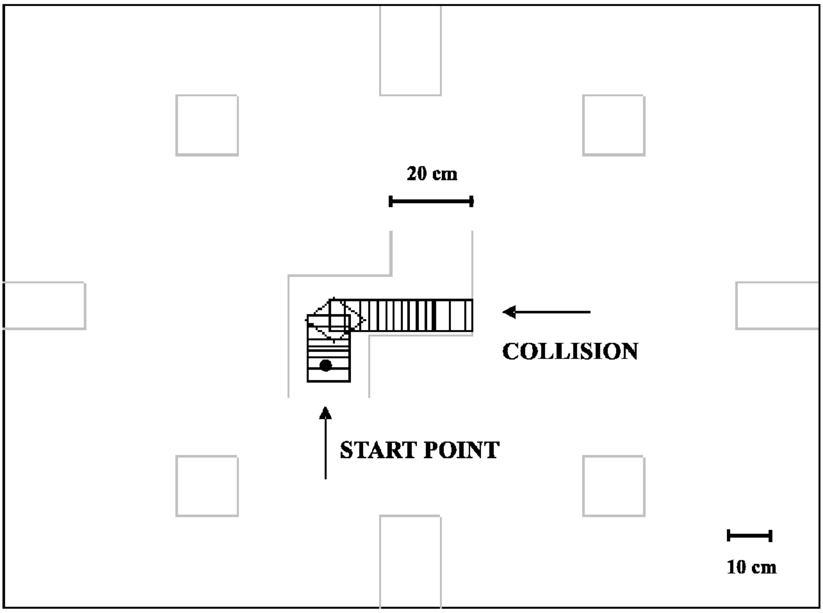

Figure 5. The virtual environment is a 200 cm square with eight obstacles and a labyrinth. The simulated mobile robot is a 10 cm square. The reach of the sensors is 10 cm. Some difficulties in learning the task arise because the only reinforcement signal is when the agent collides. In this case, if the agent goes ahead instead of turning left when it has a free path to the left, it makes up to twelve moves in the start of the trial (the starting direction of movement is north). This makes the learning more difficult, as there is only one punishment in twelve moves. In this way, the mistaken action lasts longer under Hebbian learning than under anti-Hebbian learning. This local minimum is avoided by having enough exploratory behavior.

Some difficulties in learning the task arise because the only reinforcement signal is when the agent collides. For example, when the agent receives a CS indicating that there is a free path to the left, the agent can be misled to go ahead. It receives only one punishment in twelve moves, so the mistaken action lasts longer under Hebbian learning than under anti-Hebbian learning (see Figure 5). This local minimum can be avoided by having enough exploratory[1] behavior.

Reinforcement learning algorithms, such as Q-Learning, rely on the assumption that the underlying task corresponds to a Markov decision process (MDP). However, this hypothesis does not hold true when the state of the environment is not completely known to the learning agent [26]. A partially observable Markov decision process (POMDP) is a generalization of a MDP that restricts the state information available to the learner [17]. As in the present article, in a POMDP the control agent has sensors that return some estimate of the state of the environment. These algorithms, applied in non-MDP situations, can sometimes work well, as in [3], but can go very wrong, as demonstrated in [26]. Singh et al. [26] demonstrated that in a POMDP the best stochastic policy can be arbitrarily better than the best stationary deterministic policy. They also showed that reinforcement learning algorithms do not degrade gracefully with increasing non-Markovianness in a POMDP. Different approaches to deal with POMDPs can be found in [17, 19, 26].

In this way, lack of information in position sensors and in the reinforcement feedback makes it impossible for some of the conditions for convergence of Q-Learning to be fulfilled. Thus, we expected that Q-Learning would not be 100% effective in avoiding obstacles.

[1] Here, exploratory behavior refers to the random selection of the model's responses.

### 4.3 Agent's Parameters

In each trial the agent controls the mobile vehicle during 400 steps or until it collides, starting from the same position. The experiment consists of 150 trials. Fifty repetitions of the experiment are performed in order to obtain a mean number of steps performed in each trial. Each repetition starts with a naïve agent.

The simulated environment provides the agents with the states of their sensors, and then agents report the action taken to the environment. The agents receive a signal indicating if they have collided or not. Finally, all this cycle is repeated. Q-Learning can work straightforwardly with this structure; however, the operant model differs in how it processes the information. The operant model works in real time, while Q-Learning does not. Q-Learning takes an action once the state is known, and its updating rule is applied once the punishment is received. In contrast, the operant model can take an action at any moment, and it updates the synaptic weights in real time. The presentation of a CS, the response performed, and the reinforcement received occur at precise instants. However, the internal processing of the time by the model does not influence the environment. The model receives the sensor information. When a response neuron is selected, the action is performed, and if the agent collides, it receives a punishment that is processed internally. Once an action is selected, it cannot be changed during the turn, and the sensor information is not updated until the move is completed.

We define a time step as an updating of the model equations. The operant model computes the whole cycle of sensor information, action taken, and reinforcement obtained in 60 time steps. Sensor information is presented during the first 25 time steps, the response has a duration of 5 time steps, and the reinforcement is presented for 10 time steps (see Table 2).

We tried different proportions of forced exploration in Q-Learning ($\varepsilon$-greedy exploration); however, the best performance is achieved without forcing any exploration. In the case of the operant model, exploration is performed in the following way:

1. If in the first 15 time steps no action is executed, the noise level is raised from 0 to 5.
2. Between trials 0 and 30, the probability of executing a response at random is 0.4; thereafter it is 0.

Tables 1–4 show the parameters of the experiments of both the operant model and the Q-Learning model.

Table 1. Mobile vehicle parameters.

| | |
|---|---|
| Length | 10 cm |
| Width | 10 cm |
| Sensor threshold | 0.1 |
| Turning radius | 5 cm |
| Opening angle | 60° |

Table 2. In the model of operant behavior, in time steps, for each move, the durations of the CS and US, and the response time.

| | |
|---|---|
| Move | 60 |
| CS | 25 |
| US | 10 |
| Response | 5 |

Table 3. Operant behavior model parameters.

| | | |
|---|---|---|
| $\beta$ | 0.005 | Short-term memory decay rate |
| $\varepsilon$ | 0.25 | Short-term memory rise rate |
| $\xi$ | 1.2 | Prediction neuron maximum value |
| $\upsilon$ | 10 | Prediction neuron nonlinear output parameter |
| $\sigma$ | 0.4 | Prediction neuron nonlinear output parameter |
| $\mu$ | 0.35 | Response neuron threshold |
| $\kappa$ | 0.2 | Slope parameter of the clamping equation of the response neuron's synaptic weights |
| $\eta_i$ | 0.4 | Rise rate of $P$ neuron synaptic weights |
| $\eta_d$ | 1.5 | Decay rate of $P$ neuron synaptic weights |
| $\Psi$ | 0.99977 | Decay rate of response neuron synaptic weights |
| $\Omega$ | 0.6 | Hebbian learning threshold |
| $\lambda_{\text{pos}}$ | 0.16 | Hebbian learning constant value when the output of the prediction neuron is below the threshold |
| $\lambda_{\text{neg}}$ | 2 | Hebbian learning constant value when the output of the prediction neuron is above the threshold |
| $V$[US] | 0.2 | Fixed weight of the prediction neuron synaptic weight corresponding to the US input |
| US intensity | 6 | |
| CS intensity | 1 | |
| CX intensity | 0.15 | |

Table 4. Q-Learning parameters. $n(i, a)$ is the number of times that the agent executed the action a in the state $i$.

| | |
|---|---|
| Punishment | 1 |
| $\eta_n(i_n, a_n)$ | $\frac{1.6}{n(i,a)}$ |

## 4.4 Results

We compared the performance of Q-Learning and of our operant model agent in different environments, with similar results. Here, we show only one of the simulated environments. In the case of Q-Learning, we studied two possibilities: in the first one the inputs were the state of the five sensors (32 states), and in the other one the inputs were the three free-path preprocessed signals (4 states, because only one signal can be activated at any instant).

Figures 6–8 show a complete experiment without collision for each agent (32-state Q-Learning, 4-state Q-Learning, and operant behavior model agent, respectively). Figure 9 shows the average number of steps performed by the agents per trial. It can be seen that the 32-state Q-Learning agent has poor performance, and the one with highest performance is the model of operant behavior, although it learns more slowly than the 4-state Q-Learning agent (each point in the figure is an average of five consecutive trials). Figure 10 compares the average performance achieved in the last 30 trials for each type of agent (when it is considered that agents achieve stationary performance). Figures 9 and 10 show that the operant model had better performance than Q-Learning ($P < 0.001$, chi-square test).

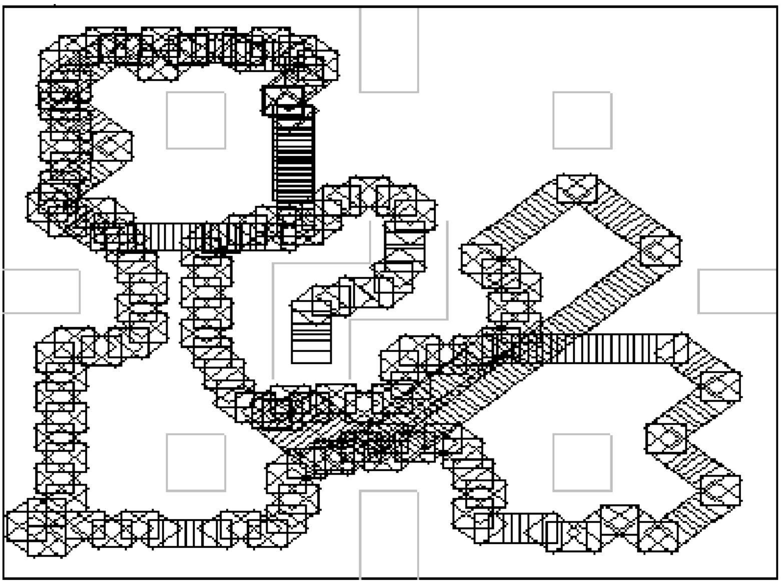

Figure 6. One example of a route without collisions performed by the 32-state Q-Learning agent.

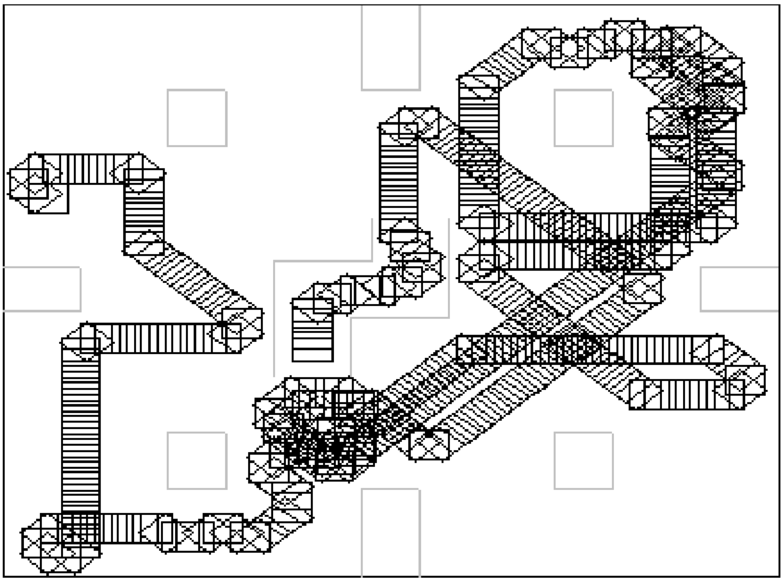

Figure 7. One example of a route without collisions performed by the four-state Q-Learning agent.

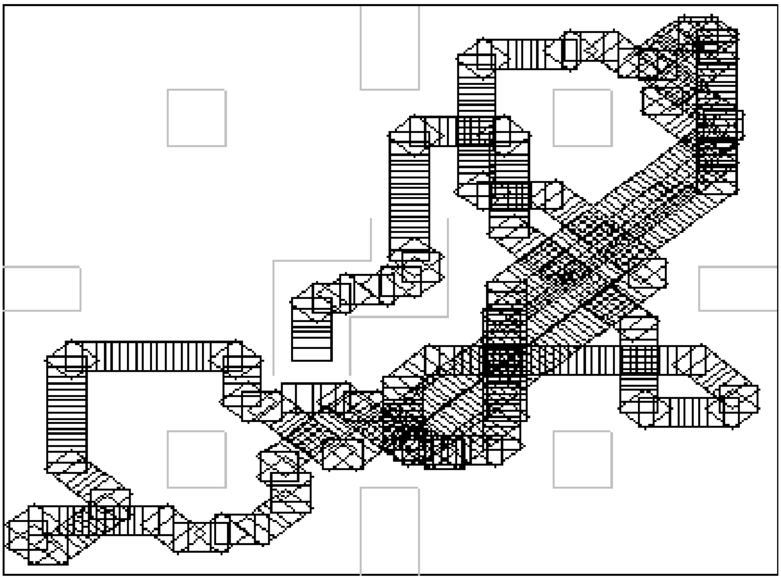

Figure 8. One example of a route without collisions performed by the operant behavior agent.

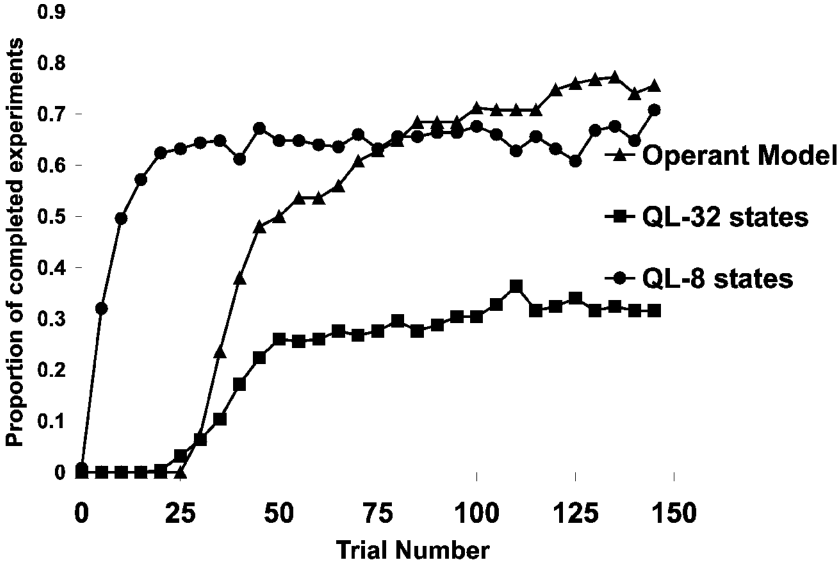

Figure 9. The experiment consisting of 150 trials is run from the same initial conditions 50 times. The figure shows the proportion of experiments that are finished without collisions as a function of the trial number for each type of agent (each mark represents the average of five trials). There is at least one policy that guarantees that there is no collision.

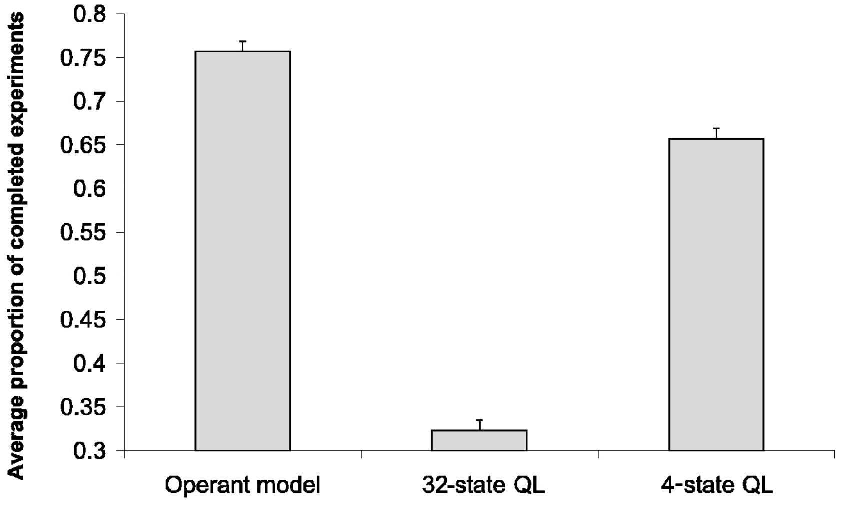

Figure 10. Proportion of experiments that are finished without collisions as a function of the trial number for each type of agent. The proportions are estimated by repeating the whole experiment 50 times. Proportions of completed experiments are estimated after the agents have learned (last 30 trials).

## 5 Discussion

The aim of this article has been to show that an operant learning model could learn to avoid obstacles. Two types of controller and two types of input were tested. The results showed that both factors influenced the performance. Although the preprocessing layer simplifies the problem to the agents, neither of them achieved a perfect performance. In this case, the only optimal policy is to move in the direction of the free path and to take an arbitrary action when there is no free path. In this way, when 4-state Q-Learning and the operant model learn the optimal policy, the two behave similarly. The only difference between them is that they can maintain different proportions of random choices for each state. On the other hand, there are other optimal policies if agents are provided with direct sensor information. For example, when there was more than one free path, one of them was chosen by default, although the selection of the other alternatives did not change the performance. Thus, the agent can learn other possible mappings with the same results. However, both the operant model without the preprocessing layer (data not shown) and the 32-state Q-Learning achieved poor performance.

Figure 9 and Figure 10 show that the operant model has better performance than Q-Learning. One of the reasons is that each agent was built to solve different problems. Q-Learning executes a Markov decision process, while our operant model simulates how animals learn to respond in the presence of different stimuli in order to receive appetitive stimuli or to avoid aversive ones.

When the hypotheses of the convergence of Q-Learning are not fulfilled, its performance cannot be predicted, thus raising a question about its robustness in such cases. However, robustness is one of the most desirable features of an algorithm in order to be able to solve many different problems with the same strategy. Probably the best example of this property is observed in animals. Previous works proposed models of operant conditioning that can learn to avoid obstacles [2, 14, 32]; however, the model used here is based on biologically plausible hypotheses that allowed us to explain the most relevant behavioral results in our previous work [20, 24, 35].

The model presented here cannot solve some tasks that Q-Learning can. For example, this model cannot be applied to operant paradigms that have nonlinear solutions, as this task cannot be performed without a preprocessing layer. Modifications to the model could allow solving problems that are more complex than the one presented here, or the same problem with better performance. This work has the purpose of showing that neural network principles applied to the description of animal behavior in different learning tasks might also be applied to build highly adaptive autonomous agents [24].

Operant conditioning is similar to reinforcement learning in that it allows an agent to adapt its behavior to get rewards from the environment when it performs a correct action. Also, state-action associations can be translated into psychological terms as stimulus-response associations. Stimulus-response-only architectures establish correlations between the response taken and the reward received. Stimulus-response associations are able to distinguish between clearly defined high and low rewards, but not when a low reward in one situation can be the highest reward attainable in another situation. Q-Learning relies on the prediction of future rewards as a function of the input stimuli and the selected response. Dyna architectures incorporate internal models. The neural network presented here can be understood as a Dyna architecture, which combines response-selection mechanisms with an internal model of the world. The model of the environment generates predictions of future events based on the combination of environmental stimuli and the agent's own response. Based on these predictions, the model selects the best response in a given situation.

The architecture of the neural network of the operant model we have presented relies on a neuron that predicts the reinforcement; the main objective is to control the learning of the response neurons in order to make the moves that avoid the punishment due to collisions. The neural network makes a simple model from the inputs, actions, and punishment that it receives. Modification of the response neurons' synaptic weights occurs not only because of the arrival of the punishment but by the prediction of its arrival. The ability to anticipate relevant events has an important adaptive property in nature. However, reinforcement learning architectures do not learn an internal model of the world's dynamic (what causes what), just the policy (what to do) and the return predictions (how well I am doing). Sutton [27] pointed out that this is an important limitation because potentially much more can be learned from an environment model than just by trial and error. Also, the punishment is just a scalar, while the sensory input signal has much potential information as a source of training.

Despite its simplicity, the present work shows that this neural network is suitable to control autonomous robots. With simple hypotheses borrowed from psychological and neurobiological experiments, the proposed operant model is a clear example of how research in control of autonomous robots can benefit from research in psychology and neuroscience.

## 6 Conclusion

We have shown how a model of operant behavior learns to avoid obstacles by being punished for collisions and that its final performance is better than that of Q-Learning. Also, it is shown that a model that has been built with the aim of explaining real behavioral experiments in animals can perform tasks in an environment different from that in which animal experiments are made.

**Acknowledgments**
The authors are grateful to Rina Zelmann and Nadia Yavitz for their helpful comments on previous versions of the manuscript. This project was supported in part by the

fellowship “Beca interna EGD Dr. Ambrosio Tognoni” from the Rotary Club of Buenos Aires.

## References

1. Amsel, A. (1958). The role of frustrative nonreward in continuous reward situations. *Psychological Bulletin, 55*, 102–119.

2. Baloch, A., & Waxman, A. (1991). Visual learning, adaptive expectations, and behavioral conditioning of the mobile robot Mavin. *Neural Networks, 4*, 271–302.

3. Barto, A. G., Sutton, R. S., & Anderson, C. W. (1983). Neuronlike elements that can solve difficult learning problems. *IEEE Transactions on Systems, Man, and Cybernetics, 13*, 835–846.

4. Bates, J. F., & Goldman-Rakic, P. S. (1993). Prefrontal connections of medial motor areas in the rhesus monkey. *Journal of Comparative Neurolobiology, 336*, 211–228.

5. Chang, C., & Gaudiano, P. (1998). Application of biological learning theories to mobile robot avoidance and approach behaviors. *Journal of Complex Systems, 1*, 79–114.

6. Crites, R. H., & Barto, A. G. (1996). Improving elevator performance using reinforcement learning. In D. Touretzky, M. Mozer, & M. Hasselmo (Eds.), *Neural Information Processing Systems*, Vol. 8.

7. Daly, H. B., & Daly, J. T. (1982). A mathematical model of reward and aversive nonreward: Its application in over 30 appetitive learning situations. *Journal of Experimental Psychology: General, 111*, 441–480.

8. Daly, H. B., & Daly, J. T. (1984). DMOD—A mathematical model of reward and aversive nonreward in appetitive learning situations: Program and instruction manual. *Behavior Research Methods, Instruments, & Computers, 16*, 38–52.

9. Damper, R. I., French, R. L. B., & Scutt, T. W. (2000). ARBIB: An autonomous robot based on inspiration from biology. *Robotics and Autonomous Systems, 31*(4), 247–274.

10. Dean, J. (1998). Animats and what they can tell us. *Trends in Cognitive Sciences, 2*(2), 60–67.

11. Donahoe, J. W., Burgos, J. E., & Palmer, D. C. (1993). A selectionist approach to reinforcement. *Journal of the Experimental Analysis of Behavior, 60*, 17–40.

12. Donahoe, J. W., & Palmer, D. C. (1994). *Learning and complex behavior.* Boston: Allyn & Bacon.

13. Donahoe, J. W., Palmer, D. C., & Burgos, J. E. (1997). The S-R issue: Its status in behavior analysis and in Donahoe & Palmer’s Learning and Complex Behavior. *Journal of the Experimental Analysis of Behavior, 67*, 193–211.

14. Gaudiano, P., & Chang, C. (1997). Adaptive obstacle avoidance with a neural network for operant conditioning: Experiments with real robots. In *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)* (pp. 13–18). Monterey, CA.

15. Goldman-Rakic, P. S. (1987). Circuitry of primate prefrontal cortex and regulation of behavior by representational memory. In F. Plum (Ed.), *Handbook of Physiology: The Nervous System* (pp. 373–417). Bethesda, MD: American Physiological Society.

16. Hebb, D. O. (1949). *The organization of behavior: A neuropsychological theory.* New York: Wiley.

17. Jaakola, T., Singh, S. P., & Jordan, M. I. (1995). Reinforcement learning algorithm for partially observable Markov decision problems. In G. Tesauro, D. Touretzky, & T. Leen (Eds.), *Advances in neural information processing systems*, Vol. 7 (pp. 345–352). Cambridge, MA: MIT Press.

18. Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research, 4*, 237–285.

19. Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence, 101*, 99–134.

20. Lew, S. E., Wedemeyer, C., & Zanutto, B. S. (2001). Role of unconditioned stimulus prediction in the operant learning: A neural network model. In *Proceedings of IEEE Conference on Neural Networks* (pp. 331–336).

21. Mahadevan, S., & Connell, J. (1991). Automatic programming of behavior-based robots using reinforcement learning. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, Anaheim, CA.

22. McFarland, D., & Bösser, T. (1993). *Intelligent behavior in animals and robots*. Cambridge, MA: Bradford Books, MIT Press.

23. Rescorla, R. A., & Wagner, A. R. (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. In A. H. Black & W. F. Prokasy (Eds.), *Classical conditioning II: Current research and theory*. New York: Appleton-Century-Crofts.

24. Schmajuk, N., & Zanutto, B. S. (1997). Escape, avoidance and imitation: A neural network approach. *Adaptive Behavior, 6*, 63–129.

25. Schultz, W., Dayan P., & Montague, R. (1997). A neural substrate of prediction and reward. *Science, 275*, 1593–1598.

26. Singh, S. P, Jaakola, T., & Jordan, M. I. (1994). Learning without state-estimation in partially observable Markovian decision processes. In W. W. Cohen & H. Hirsh (Eds.), *Proceedings of the Eleventh International Conference on Machine Learning* (pp. 284–292). San Francisco, CA: Morgan Kaufmann.

27. Sutton, R. S. (1991). Reinforcement learning architectures for animats. In J.-A. Meyer & S. W. Wilson (Eds.), *From animals to animats: Proceedings of the First International Conference on Simulation of Adaptative Behavior* (pp. 288–296). Cambridge, MA: MIT Press.

28. Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.

29. Staddon, J. E. R. (1983). *Adaptive behavior and learning*. Cambridge, UK: Cambridge University Press.

30. Steels, L. (1991). Towards a theory of emergent functionality. In J. A. Meyer & S. W. Wilson (Eds.), *From animals to animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior* (pp. 451–461). Cambridge, MA: Bradford.

31. Touretzky, D. S., & Saksida, M. L. (1997). Operant conditioning in skinnerbots. *Adaptive Behavior, 5*(3/4), 219–247.

32. Verschure, P. F., & Voegtlin, T. (1998). A bottom up approach towards the acquisition and expression of sequential representations applied to a behaving real-world device: Distributed adaptive control III. *Neural Networks, 11*, 1531–1549.

33. Waelti, P., Dickinson, A., & Schultz, W. (2001). Dopamine responses comply with basic assumptions of formal learning theory. *Nature, 412*, 43–48.

34. Watkins, C. J. C. H. (1989). Learning with delayed rewards. Ph.D. dissertation, Psychology Department, Cambridge University.

35. Zanutto, B. S., & Lew, S. (2000). A neural network model of aversive behavior. In M. H. Hamza (Ed.), *Proceedings of the IASTED Neural Networks NN'2000* (pp. 118–123). Zürich: IASTED/ACTA Press.