1998-02

Application of Biological Learning Theories to Mobile Robot Avoidance and Approach Behaviors

Application of biological learning theories to mobile robot avoidance and approach behaviors

Carolina Chang and Paolo Gaudiano

February 1998

Technical Report CAS/CNS-98-006

Boston University Center for Adaptive Systems and
Department of Cognitive and Neural Systems
677 Beacon Street
Boston, MA 02215

Submitted to *Journal of Complex Systems*

Application of biological learning theories to mobile robot avoidance and approach behaviors

Carolina Chang and Paolo Gaudiano

Boston University Neurobotics Lab
Department of Cognitive and Neural Systems
677 Beacon Street, Boston, MA 02215

E-mail: {cchang,gaudiano}@bu.edu

Abstract

We present a neural network that learns to control approach and avoidance behaviors in a mobile robot using the mechanisms of classical and operant conditioning. Learning, which requires no supervision, takes place as the robot moves around an environment cluttered with obstacles and light sources. The neural network requires no knowledge of the geometry of the robot or of the quality, number, or configuration of the robot's sensors. In this article we provide a detailed presentation of the model, and show our results with the *Khepera* and *Pioneer 1* mobile robots.

1 Introduction

When an animal has to survive in a complex, unknown environment, it must somehow learn to recognize informative cues in the environment, and to predict the consequences of its own actions. Biological organisms are a clear example that this sort of learning is possible in spite of what, from an engineering standpoint, seem to be insurmountable difficulties: noisy sensors, unknown kinematics and dynamics, nonstationary statistics, and so on.

We are interested in understanding how animals are able to solve complex problems such as learning to navigate in an unknown environment, so that we may apply what is learned from biology to the control of robots. In particular, in this article we describe a neural network model of classical and operant conditioning that learns to control the avoidance and approach behaviors of a wheeled mobile robot.

The neural network that we describe here is based on a theoretical model of classical and operant conditioning first proposed by Grossberg in 1971 (Grossberg, 1971, 1982). The model shows how an organism, in this case a robot, can learn without supervision to recognize simple stimuli in its environment and to associate them with different actions. In particular, our model is trained by letting a robot move around in an environment containing some objects that lead to punishment (obstacles with which the robot collides) and some other objects that lead to reward (lights). Briefly stated, whenever the robot receives punishment because of a collision, an inhibitory association is learned between the activity of neurons representing the range sensors and neurons representing the robot's movements. After training, a given pattern of sensor activations will tend to suppress movements that would yield punishment. Similarly, an excitatory association can be learned when the robot receives a reward (e.g., sufficiently high light intensity), so that the robot will tend to promote movements toward light sources.

Avoiding obstacles and approaching light sources is not a new achievement, nor is the application of learning theories to these problems. We believe that our approach contains several novel aspects, and that it is useful and powerful for several reasons. First, our model can learn approach and avoidance behaviors simultaneously and quite rapidly. Second, it is based on an egocentric frame of reference, so that learning in one environment generalizes to any environment. But the most important feature of our model, we believe, is that it requires no implicit or explicit knowledge about the shape of the robot, the quality and sensitivity of the sensors, or the configuration of the sensors on the robot. Hence, our model minimizes the need for calibration, and it can be of great use in applications were multiple robot platforms may be used, or where the characteristics of the sensors are unknown or variable. To demonstrate this point we have used exactly the same network to learn approach and avoidance behavior in two very different mobile robot platforms: the *Khepera*, and the *Pioneer 1*.

Preliminary partial results of our work have been presented in condensed form at recent meetings (Gaudiano et al., 1996a; Chang & Gaudiano, 1997; Gaudiano & Chang, 1997). In this article we present a complete, detailed description of the model and of the results obtained with real robots. We begin the presentation with a brief introduction to two forms of animal learning known as classical and operant conditioning, and with a general description of the theory on which our model is based. Following a detailed description of our model we then present our results with the two different types of robot. We close the article with discussion and conclusion sections.

2 Classical and Operant Conditioning

Psychologists have identified classical and operant conditioning as two primary forms of learning that enable animals to acquire the causal structure of their environment. In the classical conditioning paradigm, learning occurs by repeated association of a *conditioned stimulus* (CS), which normally has no particular significance for an animal, with an *unconditioned stimulus* (UCS), which has significance for an animal and always gives rise to an *unconditioned response* (UCR). For example, a rat that is repeatedly shocked (UCS) shortly after a red light is turned on (CS) will associate the red light with fear (UCR), meaning that eventually, presentation of the red light alone elicits a *conditioned response* (CR) resembling the fear response elicited by the shock itself. Hence, *classical conditioning* is the putative mechanism for learning to recognize informative stimuli in the environment.

In the case of *operant conditioning*, an animal learns the consequences of its own actions. More specifically, the animal learns to exhibit more frequently a behavior that has led to reward in the past, and to exhibit less frequently a behavior that has led to punishment. For example, a pigeon can be trained to peck at an illuminated key in order to receive a small food reward, while a human being might learn to stop at a red light in order to avoid getting in an accident.

In the field of neural networks research, it is often suggested that neural networks based on associative learning laws can model the mechanisms of classical conditioning, while neural networks based on reinforcement learning laws can model the mechanisms of operant conditioning. However, both of these classes of models are too simple to function in realistic, unstructured environments. This is not to say that associative learning and reinforcement learning do not exist in some form or another in biological organisms. Instead, the problem seems to lie in the use of rather simple, "monolithic" networks designed around each particular neural network law. At least two fundamental problems arise from these sorts of neural networks: first, the majority of neural networks function only as long as the inputs and outputs are controlled and timed carefully with respect to each other; second, most neural networks have no means of learning to discriminate "good" inputs from "bad" inputs on the basis of an internal value system.

The first of these problems has been aptly dubbed the *synchronization problem* by (Grossberg, 1971, 1982): how can learning between a CS and a UCS occur reliably even though they are presented at different times on different trials? The problem of discriminating "good" from "bad" can be discussed in the context of *motivation*, the internal force that produces actions on the basis of the momentary balance between our needs and the demands of our environment (Dorman & Gaudiano, 1994): somehow humans and animals are able to estimate the affective value of different stimuli, and learning is constrained to those cues and events that are affectively meaningful to them.

The ability to identify and discriminate what is good from what is bad is essential for an organism to survive in an unstructured environment. In practice, neural networks are rarely left to fend for themselves in the real world, learning to recognize which things are good and which are bad. Our work demonstrates that this sort of autonomy can be achieved, at least in part, with neural models that are rooted in behavioral and physiological studies.

3 Controlling a mobile robot through operant conditioning

In 1971, Grossberg proposed a detailed neural network theory of classical and operant conditioning which was designed to account for a variety of behavioral data on learning in vertebrates. The model was refined in several subsequent publications. (Grossberg & Levine, 1987), and (Grossberg & Schmajuk, 1987) report on detailed computer simulations of different components of the conditioning circuit.

Before providing details of the model and our own implementation of it, we provide an intuitive description of the main elements of the model. Fig. 1 is a schematic of the overall structure of Grossberg's theory. In the figure, populations of neurons are represented by boxes, while the interconnections between populations are represented by lines. We use the term "population" to refer to a collection of simulated neurons performing a given function; this is comparable to the term "layer" used in other contexts.

The essential departure from a typical associative memory model is in the use of motivational signals to modulate learning. At the core of the model are several assumptions, which (Grossberg, 1982) describes in terms of psychological *postulates*. The first design consideration of the model is that those stimuli that are initially not significant to the organism (i.e., CSs) are unable to generate emotional or behavioral responses, whereas a few stimuli that are innately significant to the organism (i.e., UCSs) always lead to an emotional and behavioral response (UCR). This is represented in Fig. 1 by the modifiable connections (semi-circles) between the CS population and the Reward/Punishment population, and by modifiable connections between the gated CS population and the Behavior Generation population. In contrast, the UCS operates through fixed, strong connections to these populations, which are represented by thick arrows in the figure. The gated CS nodes require joint activation of the sensory (i.e., CS) and emotional (i.e., Reward/Punishment) input in order to be activated. Hence, as long as the connections emanating from the CS are weak, the gated CS nodes cannot be activated by a CS alone, and behaviors cannot be generated by the CS.

Another psychological postulate states that a CS can learn to generate emotional and behavioral responses on its own by repeated pairing with a UCS. To satisfy this criterion, each UCS activates two populations: the Reward/Punishment population and the Behavior Generation population. The Reward/Punishment population, which Grossberg refers to as *drive* nodes, carry the emotional valence of the UCS. For instance, shock is a UCS that elicits fear, while food is a UCS that elicits pleasure. In a similar fashion, different UCS stimuli can generate different behaviors: shock generates an avoidance behavior, while food generates an approach behavior.

Through repeated pairing with a UCS, a CS can acquire the ability to generate emotional and behavioral responses that resemble those of the US with which it is paired. So, for instance, a bell that is repeatedly paired with shock will eventually elicit fear and avoidance behavior when presented alone, while a light repeatedly paired with the arrival of food will eventually elicit pleasure and approach behavior (as Pavlov's dogs learned to salivate in response to the ticking sound of a metronome).

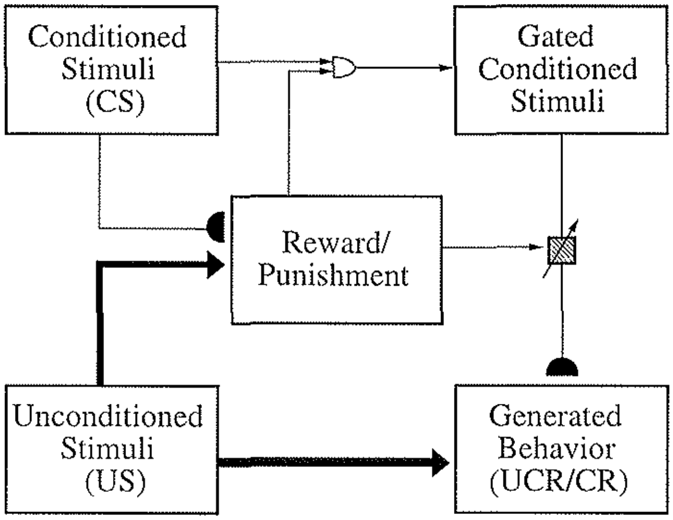


Figure 1: Schematic Conditioning model

In summary, the Reward/Punishment (or drive) nodes restrict learning to stimuli that are paired with emotionally significant events. This is an important departure from traditional connectionist approaches where every input-output pair presented to the network is learned.

Fig. 2 illustrates our detailed implementation of the circuit schematized in Fig. 1. Each block in the diagram is replaced by a more detailed representation of the corresponding neural population. The resulting neural network is able to control a mobile robot in order to exhibit the obstacle avoidance behavior.

In this model the *sensory cues* (i.e., CSs) are stored in Short Term Memory (STM) within the population labeled S. This population includes competitive interactions to ensure that the most salient cues are contrast enhanced and stored in STM while less salient cues are suppressed. In the present model the CS nodes correspond to activation from the robot's range sensors.

The *drive node* D corresponds to the Reward/Punishment component of Fig. 1. Learning can only occur when the drive node is active. The cells in population P correspond to the Gated Conditioned Stimuli, and are represented as triangular nodes to denote that they are *polyvalent* cells. Polyvalent cells require the convergence of two types of input in order to become active. As described in the schematic model, these inputs come from the CS population and from the Reward/Punishment (i.e., drive) node.

According to Grossberg's theory, the drive node is also polyvalent: it needs the joint activation of a stimulus, and an internal *homeostatic* signal in order to become active. An example of a homeostatic signal is hunger, which indicates the body's internal need for food when the body detects a low concentration of sugar in the bloodstream. In this case, an animal will not eat even in the presence of food unless it is hungry. A different situation arises in the case of aversive stimuli: an animal should always perform an avoidance behavior in the presence of an aversive stimulus. One way to interpret this is that there is a homeostatic signal corresponding to a sort of "survival instinct," which is active at all

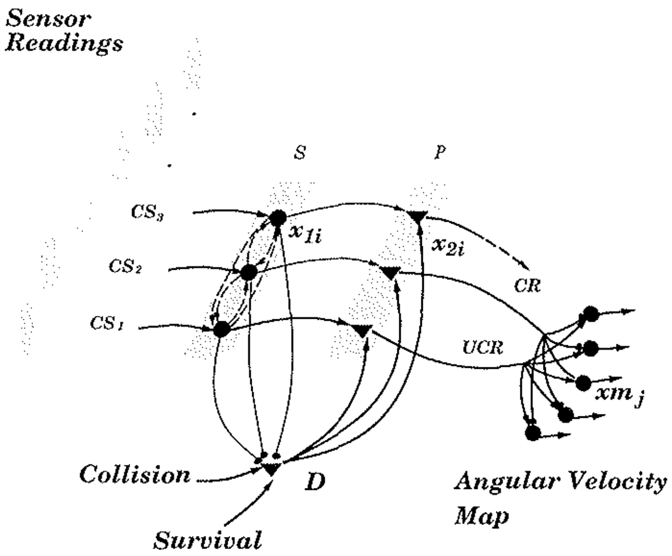


Figure 2: Conditioning model for obstacle avoidance. The robot's range sensor activities represent the CSs. A collision detector activates the UCS. Motor learning occurs at a population coding the robot's target angular velocity. After conditioning, the pattern of activity across the range sensors can predict a collision and modify the robot's angular velocity to avoid obstacles.

times. In our model, the UCS corresponds to the robot colliding with an obstacle, which can be detected through a bump sensor, or when any one of the range sensors indicates that an obstacle is at the sensor's minimum range, or when the robot's wheels fail to move. Assuming that the survival instinct signal is always on, the drive node associated with aversive stimuli (and thus with avoidance behaviors) only requires a relevant sensory input in order to become active.

Finally, the neurons at the far right of Fig. 2 represent the network's responses (conditioned or unconditioned), and are thus connected to the motor system. In a normal organism there may be many such networks, some giving rise to emotional responses (e.g., changes in skin conductance) and others generating actual motor behaviors. In our model the responses are generated as a range of angular velocities that drive the robot's movements. Each node in the motor population encodes a particular angular velocity. For instance, the leftmost node corresponds to turning left at the maximum rate, the central node corresponds to straight line movements, and so on. A more detailed description of this population is given below.

4 Model Details

In this section we provide a detailed description of the proposed model for the obstacle avoidance and light approach behaviors. After describing how learning of a single behavior is achieved, we discuss how an extended neural network is able to learn to exhibit multiple behaviors.

4.1 Single Drive Network

The motivation behind Fig. 2 is that whenever the robot collides with an obstacle, learning in the circuit will modify the connections between the pattern of sensor activity and the angular velocity of the robot when the collision took place. After learning, sensor activity will lead to *inhibition* of those angular velocities that previously caused collision under similar sensor pattern activation. In other words, because collisions correspond to punishment, the network learns to decrease the occurrence of actions that lead to punishment, as with the typical operant conditioning paradigm.

While it moves, the robot takes measurements from its range sensors. Contrast-enhancement enables sensors detecting closer objects to activate more strongly their corresponding nodes at population S.

Originally, the S population was modeled by Grossberg as a *recurrent competitive field*, which removes noise while contrast enhancing the input pattern. (Grossberg, 1971, 1982). In our implementation, we have simplified the competition of activations x_{1i} of population S, given by:

$$x_{1i}(t) = \frac{I_i(t)}{\sum_j I_j(t)} \tag{1}$$

Here I_i represents a sensor value which codes proximal objects with large values, and distal objects with small values. For instance, I_i corresponds to "raw" measurements of infrared sensors, while it corresponds to the complement of the raw measurements (i.e., maximum range minus the actual measurement) when ultrasound sensors are used. This is because infrared returns are larger for closer objects, while ultrasound returns are smaller for closer objects.

Notice that this is the only consideration we have to make for the network to work with different types of sensors. The network requires no knowledge of the geometry of the robot or the quality, number, or distribution of sensors over the robot's body.

Initially, the drive node D is only activated by the UCS, namely, when a collision is detected. However, after learning, sufficiently large patterns of sensor activity can also elicit activity y at the drive node D since the sensory nodes (CSs) have learned to predict impending collisions (UCS). Activation y of the drive node D is given by:

$$y(t) = \sum_i x_{1i} z_{1i}(t) - T_y + UCS(t) \tag{2}$$

where UCS(t) represents the collision status at time t ($UCS = 1$ if a collision just occurred, and $UCS = 0$ otherwise), z_{1i} is the adaptive weight connecting the sensory node x_{1i} to the drive node, and T_y is a threshold that controls how easily the drive node is activated.

The activation x_{2i} of the polyvalent cells at population P (or gated CSs), is given by:

$$x_{2i}(t) = x_{1i}(t) f(y(t)) \tag{3}$$

where $f(y(t))$ is defined as:

$$f(y(t)) = \begin{cases} 1 & \text{if } y(t) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

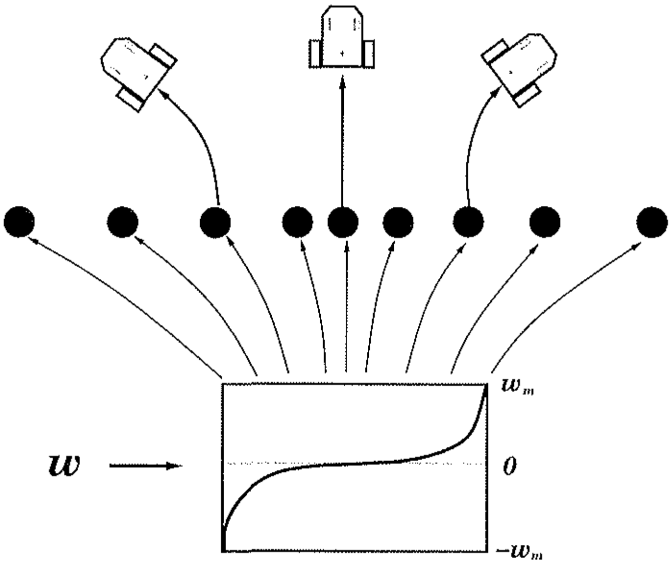


Figure 3: Angular velocity map. Each node represents an angular velocity w developed by the robot. A sigmoidal transformation function leads to a higher density of nodes for velocity values close to zero.

Notice that equation 3 codes the need for joint activation from the sensory nodes (CS) and the drive node (Punishment/Reward), in order for the gated CSs to become active, as explained in section 3.

Activation of the drive node allows two different kinds of learning to take place: the learning that couples sensory nodes with the drive node, and the learning that inhibits the movements performed just before the robot collided.

The first type of learning follows an associative learning law with decay. This learning enables the most active sensory nodes to accrue strength in their connections z_{1i} to the drive node (on the left side of Fig. 2), so that eventually the sensory nodes will be able to activate the drive signal on their own, and thus to activate the polyvalent cells P, and ultimately a motor response. The primary purpose of this learning scheme is to ensure that learning occurs only for those CS nodes that were active within some time window prior to the collision (UCS). The associative learning law is given by:

$$z_{1i}(t) = Lz_{1i}(t-1) + Px_{1i}(t)f(y(t)) \tag{5}$$

where L is the weight decay, and P is the learning rate.

The second type of learning, which is also of an associative type but inhibitory in nature, is used to map the sensor activations to the angular velocity map. Fig. 3 illustrates the scheme we used to represent angular velocities. In this figure, the leftmost node represents an angular velocity of $-w_m \frac{rad}{s}$, and the rightmost node represents an angular velocity of $w_m \frac{rad}{s}$, where w_m is the maximum angular velocity developed by the robot. The central node corresponds to a straight line movement (angular velocity equals zero). The map includes a sigmoidal transformation, whereby angular velocities close to zero are represented by a greater number of nodes for finer control.

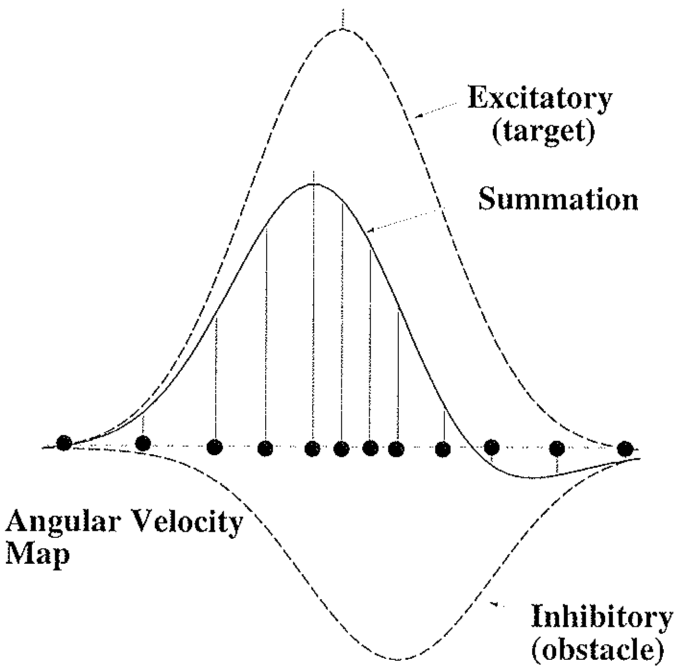


Figure 4: The peak shift property. The positive Gaussian distribution represents the desired angular velocity, whereas the negative distribution represents the activation from the conditioning circuit. The summation of the two distributions determines the angular velocity that will be used to drive the robot. Notice how the peak of the excitatory Gaussian is shifted by the inhibitory Gaussian.

For a particular desired angular velocity[1] α, the corresponding node n_d in the angular velocity map is given by:

$$n_d(t) = \begin{cases} \frac{N}{2} + \frac{N(a_w+0.5w_m)\alpha(t)}{w_m(1.0+\alpha(t))} & \text{if } \alpha(t) > 0 \\ \\ \frac{N}{2} + \frac{N(a_w+0.5w_m)\alpha(t)}{w_m(1.0-\alpha(t))} & \text{otherwise} \end{cases} \tag{6}$$

where N is the number of nodes in the angular velocity map, w_m is the maximum angular velocity, and a_w is a constant that controls the slope of the sigmoid function.

When driving the robot, activation is distributed as a Gaussian centered on the desired angular velocity. The use of a Gaussian leads to smooth transitions in the angular velocity even with few nodes, and it is also crucial for our obstacle avoidance scheme, as depicted in Fig. 4. When an excitatory Gaussian is combined with an inhibitory Gaussian at a slightly shifted position, the resulting net pattern of activity exhibits a maximum peak that is shifted in a direction *away* from the peak of the inhibitory Gaussian. Hence, the presence of an obstacle to the right causes the robot to shift to the left, and *vice versa*. Further details on this operation can be found elsewhere (Gaudiano et al., 1996b).

[1] By *desired* angular velocity we simply mean the angular velocity required to move the robot toward a particular target.

By using an inhibitory learning law, the polyvalent cells corresponding to the sensory nodes acquire negative connection weights that learn to generate a pattern of inhibition that matches the sensor activity profile active at the time of collision. For instance, if the robot was turning right and collided with an obstacle, the range sensor neuron most active shortly before the collision will learn to generate an inhibitory Gaussian centered upon the right-turn node in the angular velocity population. This learning law leads to the development of negative weights, as given by:

$$zm_{i,j}(t) = zm_{i,j}(t-1) - Mx_{2i}(t)\left[\frac{gx_j(t)}{1+(j-G(t))^2} + zm_{i,j}(t-1)\right] \tag{7}$$

where $zm_{i,j}$ represents the adaptive weight from the polyvalent cell i to the node j of the angular velocity map. M is the learning rate, and gx_j is a Gaussian function centered on the desired movement direction, as described by:

$$gx_j(t) = e^{-(j-n_d(t))^2/\sigma^2} \tag{8}$$

$G(t)$ is the index of the node in the angular velocity map for which gx_j is maximal, and σ is the standard deviation of the Gaussian.

Once learning has occurred, the activation of the angular velocity map is given by two components (Fig. 4). An excitatory component, which is generated directly by the sensory system, reflects the angular velocity required to reach a given target in the absence of obstacles. We have shown previously how this signal can be derived from the sensors (Gaudiano et al., 1996b); for simplicity here we assume that the angular velocity is proportional to the angle between the robot's current heading and the target. A second, inhibitory component, generated by the conditioning model in response to sensed obstacles, moves the robot away from the obstacles as a result of the activation of sensory signals in the conditioning circuit. The equation that describes this behavior is:

$$xm_j(t) = gx_j(t) + \sum_i x_{2i}(t)zm_{i,j}(t-1) \tag{9}$$

The node in the angular velocity map that has maximal activation after the summation of the excitatory and inhibitory Gaussians determines the angular velocity that the robot will perform in its next movement. The angular velocity coded by the winning node is computed as follows:

$$w(t) = \begin{cases} \frac{w_m[J(t)-N/2]}{N(a_w+0.5w_m)-w_m[J(t)-N/2]} & \text{if } J(t) > N/2 \\ \\ \frac{w_m[J(t)-N/2]}{N(a_w+0.5w_m)+w_m[J(t)-N/2]} & \text{otherwise} \end{cases} \tag{10}$$

where $J(t)$ is the index of the node with maximal activation.

In principle, eq. 10 is the inverse function of eq. 6. Together, these two equations account for the transformation from angular velocities to nodes in the angular velocity map, and *vice versa*.

Notice that after learning, the desired (i.e., α) and resulting (i.e., w) angular velocities might differ. In the presence of obstacles, the learned inhibitory Gaussian causes the peak in the angular velocity map to shift, moving the robot away from obstacles, and also away from its instantaneous desired direction.

The output of the angular velocity population is then decomposed algorithmically into left and right wheel angular velocities. In an alternative approach by (Gaudiano et al., 1996b), the transformation from the angular velocity population to actual wheel velocities can be done adaptively with another neural network.

The technique we have just described for obstacle avoidance (i.e., using a difference of Gaussians) is related to the technique widely known as *potential fields* (Khatib, 1986; Latombe, 1991), though the methods differ in various details. For instance, we only utilize a one-dimensional map of neurons representing instantaneous desired angular velocities, rather than actually building a potential function based on sensor activities. Nonetheless, the approach used here is similar to potential fields and other methods that "weigh" the presence of obstacles sensed around the robot. In fact, this part of the circuit should be easy to replace with an alternative, but comparable method. The present method has the desirable features of being computationally expedient, easy to implement, and robust to parameter manipulation. We also chose this particular technique because of our prior experience with it (Gaudiano et al., 1996b; ?; Zalama, Gaudiano, & López-Coronado, 1995).

The neural network model described above has been used to develop an obstacle avoidance behavior in our robots. Learning of the inhibitory Gaussian distribution in order to produce a peak shift in the activation of the angular velocity map lead to the desired avoidance behavior.

In another experiment, we used the same neural network design of Fig. 2 in order to develop an approach behavior in the robot. Notice that this behavior is completely opposite to the obstacle avoidance behavior since it requires the robot to move *towards* the location of the source of sensory stimulation (instead of *away*).

In this case, detection of an increase in the level of light present in the environment corresponds to the UCS. For the CSs, we used the light intensity measurements of the Khepera's infrared sensors. Light is regarded as a reward, e.g., food, which activates a "pleasure" drive, and elicits the approach behavior.

To generate the approach behavior, the learning associative law of eq. 7 was modified in order to build excitatory Gaussian distributions that would move the robot towards the location of sensory stimulation, as depicted in Fig. 5. The new learning law is given by:

$$zm_{i,j}(t) = zm_{i,j}(t-1) + Mx_{2i}(t)\left[\frac{gx_j(t)}{1+(j-G(t))^2} - zm_{i,j}(t-1)\right] \tag{11}$$

Equation 11 allows the network to develop positive weights in the connections from the polyvalent cells to the angular velocity map. Recall that previously, collision elicited punishment signals that lead to inhibition, and thus, to the obstacle avoidance behavior. With eq. 11, instead of being punished, the robot is rewarded each time there is a certain increase in the sensors' activation.

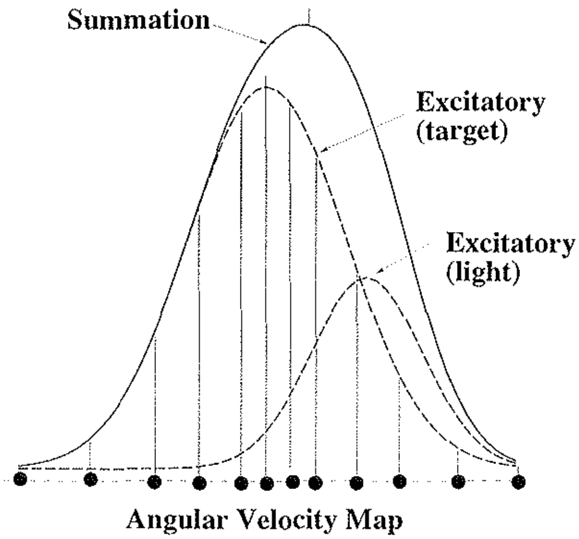


Figure 5: Peak shift for the approach behavior. The learned excitatory Gaussian distribution shifts the peak of the angular velocity map activation toward the source of sensory stimulation, leading to an approach behavior.

4.2 Multi-Drive Network

So far we have described an implementation of the conditioning circuit that can learn to either avoid obstacles or approach the source of sensory stimulation. As presented, the model is unable to learn both behaviors simultaneously. In order to be able to learn multiple behaviors, the original network of Fig. 2 was expanded as shown in Fig. 6.

The expanded model consists of sensory nodes for both kinds of cues (i.e., proximity of objects, and intensity of light). For simplicity, the figure shows only the connections of one node of each type. Note that in principle, there is no difference between the two types of nodes other than the kind of sensory information they are concerned with. To be able to elicit two opposite behaviors, the network contains two drive nodes, i.e., fear and pleasure. "Survival" is the internal homeostatic signal associated with fear and the avoidance behavior, while "hunger" is the signal associated with pleasure and the approach behavior. Each drive releases *incentive motivation* to a specific population of polyvalent cells. In the figure, Population $P-$ is associated with the fear drive and avoidance behavior, and population $P+$ is associated with the pleasure drive and approach behavior.

In the expanded conditioning circuit, the two drives compete in a *sensory-drive heterarchy* (Grossberg, 1971). That is to say, the combination of sensory activation and drive activation determines which drive will release incentive motivation, thus allowing an appropriate motor response to take place. For simplicity, we have assumed that both internal homeostatic signals (i.e., survival and hunger) are always active. This reduces the problem of drive competition to a matter of sensory activity: after learning, the strongest cues will win the competition leading to the release of the associated behavior. In a later section we also describe a modification of the appetitive (pleasure) drive to mimick a hunger signal.

The population of polyvalent cells that receives incentive motivation from the winning

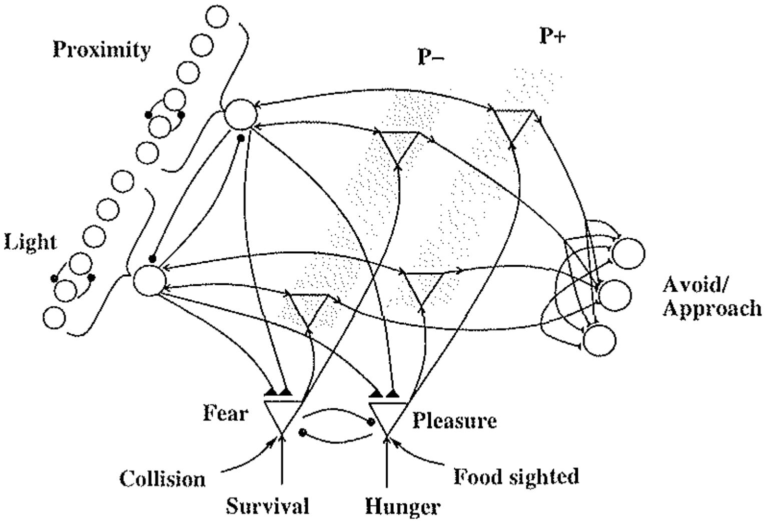


Figure 6: Expanded Neural Network. The neural network consists of two drives. The fear drive is associated with the avoidance behavior, whereas the hunger drive is associated with the approach behavior.

drive node elicits the motor response to which it is associated. Hence, at each time only one motor response is released, i.e., either the avoidance or the approach behavior.

Note that when a drive node wins the competition, it releases incentive motivation to all the polyvalent cell populations. That is, it releases incentive motivation to the polyvalent cells connected with the proximity sensory nodes and also to the cells connected with the light detection sensory nodes. This happens because initially the CSs have no special meaning; they cannot predict the occurrence of a particular UCS. It is only through learning that the network starts to discover the causal structure of the environment. When a collision occurs or food is sighted, all sensory nodes are allowed to learn. However, after repeated associations, only sensory nodes constantly active during learning would actually have their connections strengthened. Similarly, only the polyvalent cells with strong connections to the motor units will have strong influence on the motor response produced by the system. Activation of the fear drive allows learning of the avoidance behavior by the development of negative weights that describe an inhibitory Gaussian, as described earlier. At the same time activation of the pleasure drive permits learning of the approach behavior towards the source of food (e.g., light), described by an excitatory Gaussian.

The two forms of learning take place simultaneously while the robot is moving through the environment. The dynamics of the network determine which nodes learn in which conditions. In general, as long as lights and obstacles are not all overlapping, there is no problem with simultaneous learning of both types of behaviors. Likewise, after learning, the expanded neural network is capable of exhibiting multiple behaviors depending on the events that take place in the environment.

Before showing our experimental results on real mobile robots with single and multi-drive networks, we describe briefly an extension of the model to take into account a more complex form of homeostatic signal.

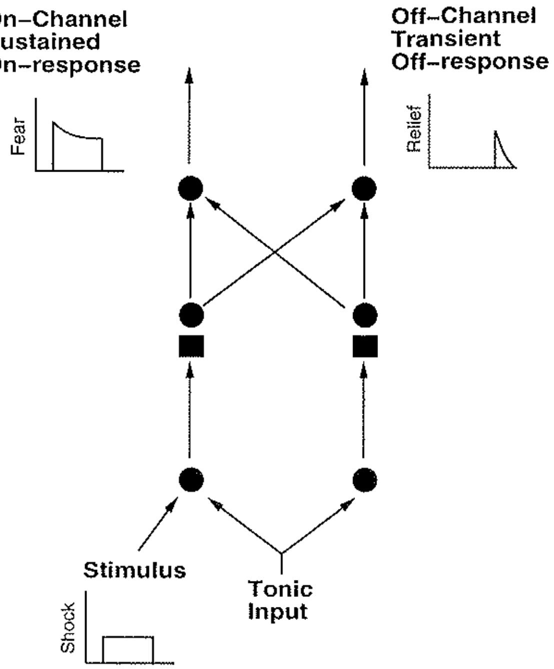


Figure 7: The gated dipole consists of two channels with mutually inhibitory connections. A sustained habituating response and a transient antagonistic rebound are elicited by a stimulus onset and offset, respectively.

4.3 Drive Activation Cycle

Although a survival instinct might be always present in organisms, it seems unrealistic to assume that hunger is always active. For this reason, we modified the proposed network in order to account for a more natural homeostatic signal activation. We wanted the hunger homeostatic signal to become inactive after food intake, and to activate again after a variable period of time, which depends on the size of the last food intake.

To this end, we employed *gated dipoles*, a neural circuit proposed by Grossberg in 1972 to model some aspects of classical and operant conditioning in vertebrates. In brief, the gated dipole is a microcircuit consisting of two channels organized in a mutually inhibitory, or *opponent* fashion (Fig. 7). The first channel, called the *on-channel*, generates a sustained yet habituative response (e.g., fear) to the onset of a stimulus (e.g., shock). In contrast, the *off-channel* produces a transient response (e.g., relief) to the offset of the same cue. The activation of the off-channel due to the stimulus offset is known as *antagonistic rebound*.

Depending on the dipole's internal laws, sustained activation of the stimulus produces diverse responses in the circuit. Dipoles that use linear laws have sustained on-responses due to the stimulus onset. On the other hand, some nonlinear laws can cause a transient on-response followed by activation of the off-channel, even before the stimulus offset. A detailed description of the gated dipoles can be found elsewhere (Grossberg, 1972, 1982).

To implement the "hunger" homeostatic signal, two coupled gated dipoles were used (Fig. 8). A nonlinear dipole describes the amount of "food" (e.g., light) that the robot has consumed. Positive activation represents the time the robot is eating while non-positive

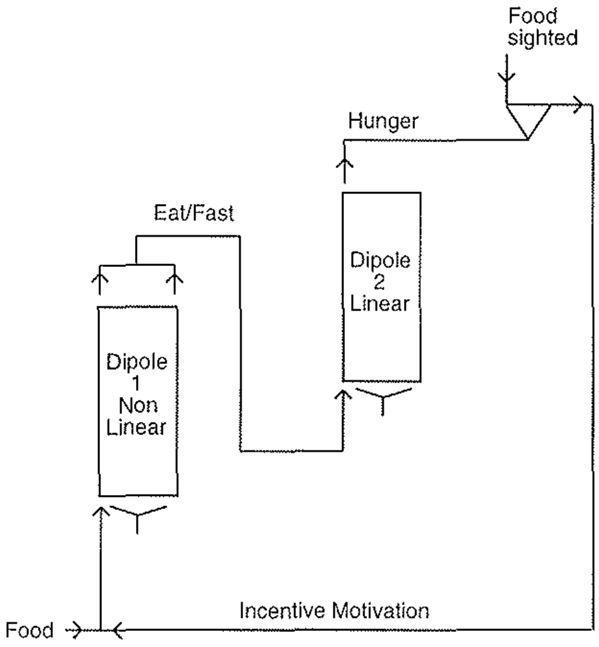


Figure 8: Two coupled gated dipoles were integrated into the extended conditioning model in order to describe the dynamics of the "hunger" homeostatic signal.

values indicates the time the robot is not eating. This information is used by a second, linear dipole to control the onset/offset of the hunger homeostatic signal. A period of time without eating (i.e., when the activation of the first dipole activation is close to zero) triggers the hunger signal, which remains on until food has been consumed (at which time the first dipole on-channel activation changes from positive to negative). At that moment, the hunger signal switches off, and a "fullness" sensation occurs (i.e., negative activation of the on-channel of the second dipole). This fullness sensation decreases with time, eventually leading to reactivation of the hunger signal.

Coactivation of the hunger homeostatic signal and conditioned stimuli that predict the arrival of food triggers the onset of the appetitive, or hunger drive. This drive competes with the fear drive in the sensory-drive heterarchy. If the hunger drive wins the competition, it releases incentive motivation, leading to an approach behavior in the presence of food. If the drive is not strong enough, no approaching behavior is released even in the presence of food.

Notice that the amount of time that the robot is not hungry depends on the amount of food it was able to take. In Fig. 9, at times near t=100 and t=1050 the robot was able to eat enough food to last about 250 time units without being hungry again. However, small food intakes near t=550 and t=850 lead to shorter fullness periods, of about 150 time units.

We should point out that the proposed mechanism is not meant to be an accurate simulation of the typical hunger-satiety cycles seen in humans and animals. However, the point is to show how the model can be extended to include more realistic and useful situations in which drives and sensory stimuli interact in a more complex fashion. For instance, we envision replacing the lights with charging stations (which of course would

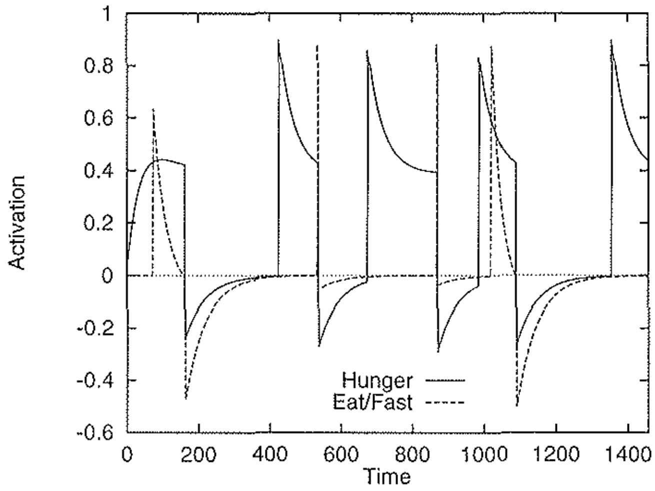


Figure 9: The hunger homeostatic signal activation cycle. Positive values of the "hunger" curve indicate that the hunger signal is active. Positive values of the "Eat/Fast" curve indicate food intake. Food is consumed only when the hunger signal is active. Hunger deactivates after food intake. The time elapsed before reactivation of the hunger signal depends on the size of the last food intake.

have to be distinguishable by the sensors), in which case the hunger drive would be directly correlated to the charge level of the battery, and after learning the robot would ignore charging stations until the battery level became sufficiently low.

5 Experimental Results

We have implemented the model just described on two real mobile robots. The Pioneer 1 (Real World Interface, Jaffrey, NH), shown in Fig. 10(a), is a small (14″ wide, 18″ long, 9″ tall), two-wheel differential-drive robot with five forward-facing and two side-facing sonar range finders. The Khepera (K-team SA, Préverenges, Switzerland), shown in Fig. 10(b), is a miniature (2.2″ diameter) differential-drive robot with eight infrared proximity sensors, six of which cover the frontal 180$^{\text{O}}$, and two sensors facing backwards. In our experiments we have ignored the two rear-facing infrareds, using only the six frontal sensors. We have previously reported our results using simulators (Gaudiano et al., 1996a; Chang & Gaudiano, 1997). We focus here on the results using real robots.

It is worthwhile to note that the implemented model requires essentially no modifications in order to run on these two robots, the only difference being that the infrared sensors on the Khepera return larger values for closer objects, while the sonars on the Pioneer 1 return smaller values for closer objects (of course, there is also a difference in the number of CS nodes).

In our model, the range sensors initially do not propagate activity to the motor population because the initial weights are small or zero. The robot is trained by allowing it to make random movements in a cluttered environment. The goal of the training phase is to give each CS node the opportunity to sample several movements that lead to collisions.

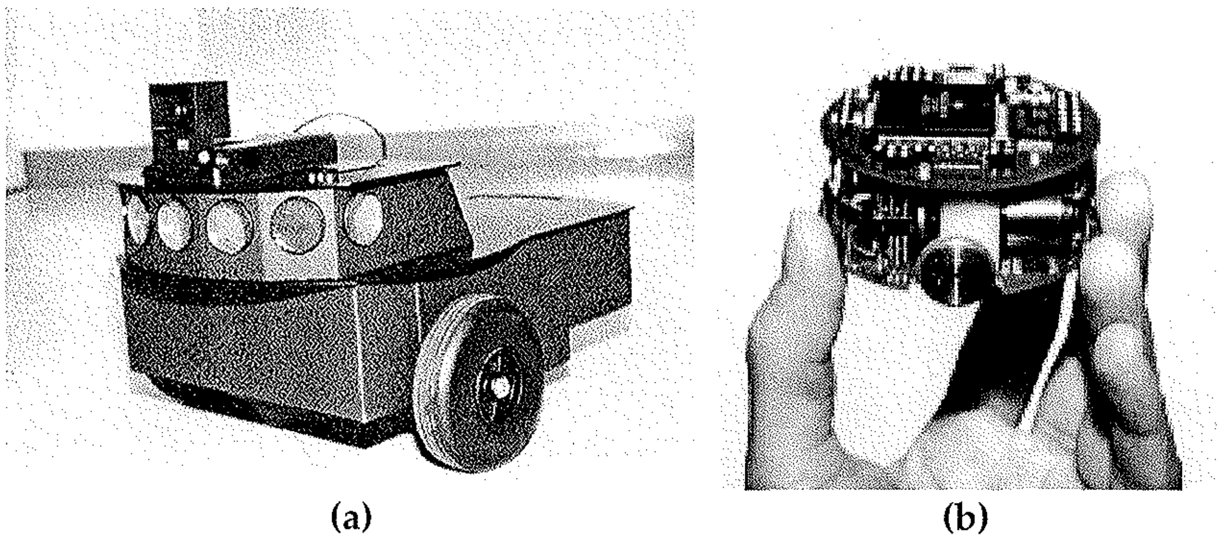

(a) (b)

Figure 10: Two robotics platforms. **(a)** The Pioneer 1 robot; **(b)** the Khepera robot.

In practice we found that it is sufficient for each CS node to be active during only a handful of collisions, when using 11 nodes in the angular velocity map. In order to generate a wide range of movements, during the training phase we turn on each node in the angular map for a brief time until a collision is registered, then switch to a new angular map node and repeat the process. We can achieve good avoidance behavior in this way with only a few collisions for each node. Fig. 11 illustrates the learning process. We obtained this curve in the following way: starting with all the weights in the network set to zero, we turn on one node in the angular map and let the robot collide with an obstacle, generating a small amount of learning, then turn on another node, and so on. At regular intervals during the training phase we temporarily disable learning and allow the robot to move from a new starting position for a total of 500 steps through the algorithm, and measure for how many of the 500 steps the robot detected a collision. On the first trial, before any learning has taken place, as soon as the robot collides it remains stuck against the obstacle, so the number of collisions is very close to 500. By the time we have trained through 50 collisions (total: meaning that each of the six sensors, on average, has sampled fewer than ten collisions) the robot is able to navigate with virtually no collisions.

The inhibitory weights developed by the neural network are depicted in Fig. 12. The adaptive connections between the sensory nodes and the angular velocity map develop in such a way that angular velocities that make the robot turn to the right (nodes close to 10) are inhibited when the sensors located at the right side of the robot are active (sensory nodes 4 and 5). Similar yet opposite inhibitory weights develop for left turns when obstacles are sensed at the left side. In the middle of the figure (nearly straight-forward movements with obstacles located straight ahead), a Gaussian-like inhibitory curve accounts for the fact that in such cases, turns to either the left or the right are needed to avoid collisions.

After sufficient training, the robot is able to wander in a cluttered and constantly changing environment while avoiding collisions with obstacles. Fig. 13 is a digital image captured using a frame grabber board that receives signals from a camcorder mounted

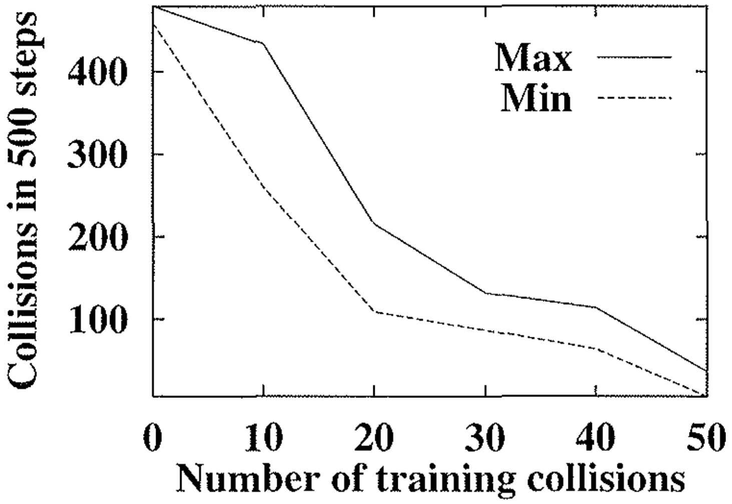


Figure 11: Learning in the Khepera robot, measured as the number of collisions in 500 steps as a function of the total number of collisions experienced during training. *Min* and *Max* refer to the best and worst learning curves out of a set of five training trials.

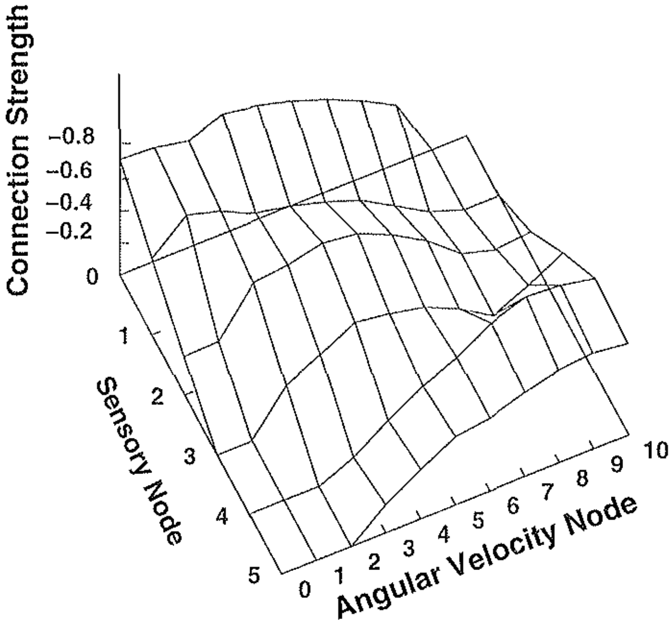
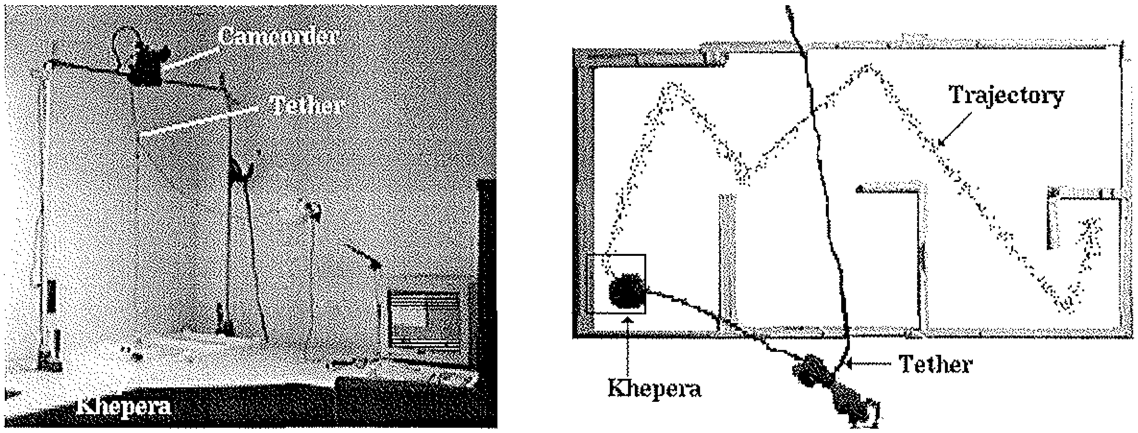


Figure 12: Adaptive connections between the sensors and the angular velocity map developed by the Khepera robot for the obstacle avoidance behavior.



(b) (a) (b)

Figure 13: Obstacle avoidance performed by the Khepera robot. An overhead camcorder and a tracking algorithm were used to capture the robot's movements. The robot communicates with a computer through a serial cable (tether). (a) Experimental setup. (b) The tracking algorithm localizes the robot and surrounds it with a moving square window. A trace of the robot's trajectory is drawn as the robot moves.

above the Khepera's environment. A tracking algorithm traces the trajectory described by the robot (black dots). The robot makes wide and fast turns when it gets very close to the obstacle due to the short range of activation of the infrared sensors (the range of the Khepera's IR sensors can vary, but in our environment it is limited to no more than 3cm). Nonetheless, the resulting movements succeed in preventing collisions.

The robot is also capable of reactive target reaching. Without having a map of the environment, in most cases the robot is able to reach arbitrary target positions, when the angle and the distance between the current position and the goal are specified. Appropriate nodes of the angular velocity map activate depending on the angle between the robot's heading direction and the target. As it travels, the robot updates its position and direction by relying on its odometry. The inhibitory learned Gaussian forces the robot to deviate from its desired trajectory when the proximity of objects is sensed.

The same kinds of results were obtained when we trained the Pioneer 1 robot to develop the obstacle avoidance behavior. About 60 collisions where required during the training phase to fully learn to avoid obstacles. Fig. 14 shows some images of the Pioneer 1 moving in our Lab while avoiding obstacles. A main difference between the two robots is that the Pioneer 1 robot developed more gradual avoidance movements than the Khepera. This happened because its ultrasound sensors are accurate in the range of several feet, whereas the Khepera's infrared sensors can only detect objects in the range of 1 inch.

In the obstacle avoidance behavior, through punishment signals the neural network learned to build inhibitory Gaussians centered in the direction of the obstacle. For the light approaching behavior, the Khepera's infrared sensors are used to detect and approach a source of light. Instead of being punished, the robot was rewarded each time an increase in the detected amount of light exceeded a given threshold. Hence, the network learned to enhance excitatory connections that would move the robot towards the

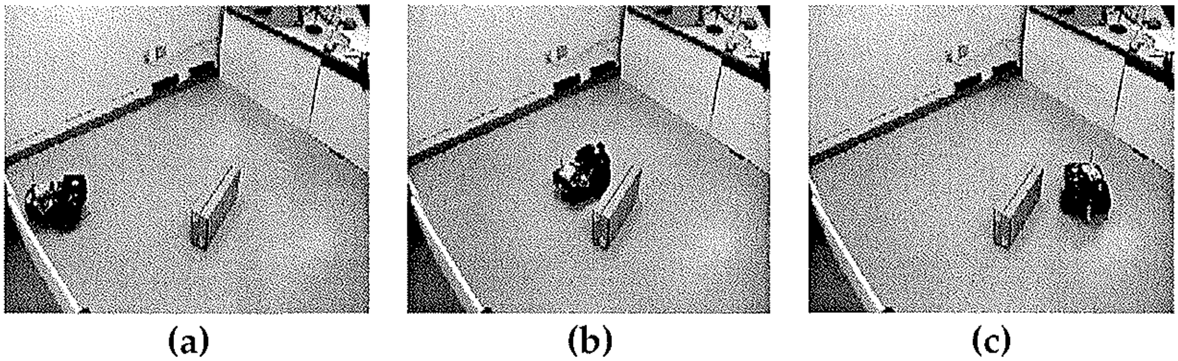


Figure 14: Obstacle avoidance behavior of the Pioneer 1 robot.**(a)** An obstacles is located in the robot's current direction of movement. **(b)** The robot makes a turn to the left to avoid a collision with the obstacle. **(c)** The robot surrounds the obstacle without colliding.

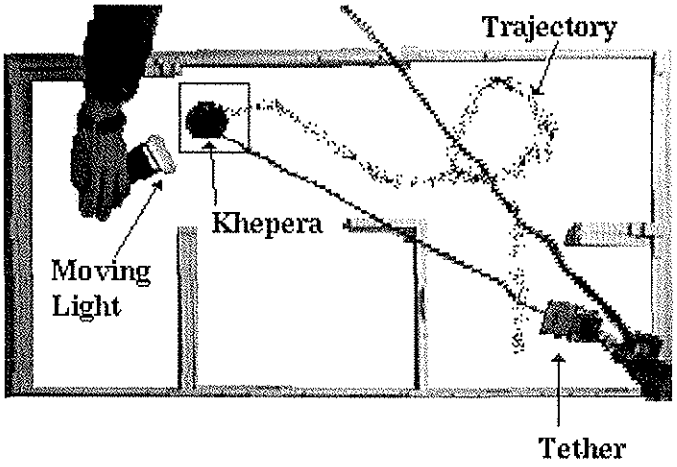


Figure 15: The Khepera robot follows a moving flashlight directed by the experimenter.

location of light (Fig. 5), by using the approach of Gaussian summation described in section 4.1. As in the previous case, after enough training the robot successfully learned to turn towards the source of light. If the light moves, the approaching behavior enables the robot follow the trajectory of the light, as shown in Fig. 15.

Simultanouns learning of the avoidance and approach behaviors in the extended conditioning model developed quite nicely. With the use of the gates dipoles for the hunger homeostatic signal, the robot approaches the source of light only when it is hungry. When the robot is not hungry or no source of light is detected, the robot keeps moving and exhibiting the obstacle avoidance behavior. Even when approaching a source of light, the robot avoids obstacles found in its path.

6 Discussion

The neural network model we have proposed for avoidance and approach behaviors in real mobile robots has been inspired by Grossberg's work on classical and operant conditioning. Without the need of supervision, the model is able to learn rapidly to avoid obstacles and to approach sources of light. Furthermore, since the model is largely independent of the nature and configuration of sensors, it can be implemented on very different robotics platforms, as demonstrated by our experimental results.

The ability to work in the real world, with real sensors, in different robotics platforms, demonstrates the model's success and robustness. In our opinion, this success is due primarily to our use of models of neural and behavioral aspects of animal learning. However, we are not the first to foresee the potential use of models of animal learning in robotics, nor is ours the only implementation of Grossberg's models in real robots. Very impressive results have been reported by (Baloch & Waxman, 1991), using the robot MAVIN. They utilized a variant of Grossberg's conditioning circuit as a part of the overall control scheme. The model they used is complex as it focuses on MAVIN's visual navigation, attacking a variety of problems with specific solutions. The main difference is perhaps in our approach, since we are more interested in achieving rapid adaptive control that is independent of the platform on which the model is being used.

Closely related to our work is the implementation of the Schmajuk and DiCarlo model reported by (Bühlmeier & Manteuffel, 1997). The basic network of Grossberg's conditioning circuit is also used for an obstacle avoidance task. However, this model differs from ours in several aspects. First, there is only one kind of learning, namely, the prediction of the UCS by the CS (classical conditioning). All the robot responses are prewired in the network, therefore all responses are regarded as reflexes in which no learning takes place. Second, as a consequence of this prewiring, knowledge of the robot's geometry is required, since the reflexes make the right wheel turn back if collision is detected in the left front side of the robot, and so on. In contrast, our model requires neither knowledge of the robot configuration nor design of reflex behaviors. Instead, the obstacle avoidance and light approach behaviors arise due to learning.

From a different approach, (Pfeifer & Verschure, 1992) also achieve obstacle avoidance by means of a form of associative learning that is modulated by appetitive and aversive stimuli. As with the model of Bühlmeier and Manteuffel, built-in avoidance reflexes require knowledge about the sensor location on the robot's body. Similarly, the model does not learn to generate new behaviors, but simply to generate built-in behaviors at the right time. Another major difference between the model of Pfeifer and Versuche's and our model is that they specifically design the network in such a way that the obstacle avoidance behavior directly inhibits the approach behavior. Therefore, there is also a pre-built preference for the avoidance reflex over the approach reflex. In our case, no prewiring is required, since competition between the combination of drives and active cues determines which behavior will be released. Therefore, our network could eventually be expanded to include more than two competing behaviors, all this without the need for further design considerations.

Given the common points between our work and the models just summarized, a combined approach might be possible in order to extend further the learning capabilities of our neural network. For instance, we could combine basic reflexes of the type used by other authors in order to try to learn more complex behaviors, such as recognizing stimuli that cannot be represented by a single sensor.

7 Conclusions

We have described a model that learns to generate avoidance and approach behaviors for a wheeled mobile robot by using a form of "self-supervised" learning. In section 4.1 we provided a detailed explanation of our implementation of the conditioning model. The robot progressively learned to avoid obstacles without the need for external supervision, but simply through "punishment" signals produced by the collision of the robot during random exploratory motion. One of the main properties of the model is that it is necessary to know neither the robot's geometry nor the configuration of the range sensors on the robot's surface, because the robot learns from past experiences to avoid directions of movement that lead to collisions. In our experiments with two different robotics platforms, i.e., Khepera and Pioneer 1, the same neural network learned to avoid obstacles, thanks to the model's platform-independence. Moreover, learning in one environment generalized to any environment since it is based on the robot's egocentric frame of reference. Experimental results with the Khepera robot showed that the neural network is also capable of learning to generate an approach behavior. Instead of "punishment" signals, "reward" signals were used to learn approach a source of light.

In section 4.2 we extended the model of conditioning to account for multiple behaviors. Training for the avoidance and approach behaviors can be done simultaneously, even though these behaviors are quite opposite. With the addition of the hunger activation cycle we showed how drives and sensory stimuli can interact in complex situations. Although the coupled gated dipoles described in section 4.3 are not meant to be an accurate model of animal feeding dynamics, they allowed us to show how the robot can choose among different behaviors depending on the moment-by-moment combination of sensorial information and internal needs.

Using the extended model, we plan to utilize the visual system of the Pioneer 1 robot to learn to approach interesting objects found in the environment. This behavior would be equivalent to the light approaching behavior we have achieved using the Khepera robot. The use of visual information instead of infrared measurements would show further the platform-independence of our model.

We also want to combine our model with a neural network for low-level control developed by (Gaudiano et al., 1996b), which allows the robot to learn in an unsupervised manner its inverse and forward kinematics using sensory feedback. The network constantly adapts to miscalibrations produced by wheel slippage, changes in the wheel sizes, and changes in the distance between the wheels. As is well known (Borenstein et al., 1996), odometry leads to accumulation of errors in the computation of the robot's position. Although odometry sufficed for our target reaching experiments, complementary methods for navigation would be needed to maintain or improve positioning accuracy.

Finally, though versatile, purely reactive navigation is not enough for the target reaching task since the robot can get stuck in local minima paths. For this reason, we want to develop higher-level navigation schemes. Planning strategies combined with "frustration" when the performed plans fail should endow the robot with more powerful navigation skills.

Acknowledgement

We would like to thank Eduardo Zalama for his contribution to earlier versions of this work, and William Streilein for his comments on an earlier draft of this paper. Support for this research was provided by the mediaCenter mbH in Friedrichshafen, Germany, and by grants from the Office of Naval Research (00014-96-1-0772 and N00014-95-1-0409).

References

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

Bekey, G. A., & Goldberg, K. Y. (Eds.). (1993). *Neural Networks in robotics*. Kluwer, Boston, MA.

Borenstein, J., Everett, H. R., & Feng L. (1996). *Navigating Mobile Robots Systems and Techniques*. A K Peters, Ltd., MA.

Bühlmeier, A., & Manteuffel, G. (1997). Operant Conditioning in robots . In Omidvar, O. & Van der Smagt, P. (Ed.), *Neural Systems for Robotics*. Academic Press, pp. 195–225 MA.

Carpenter, G. A., & Grossberg, S. (Eds.). (1991). *Pattern Recognition by Self-Organizing Neural Networks*. MIT Press, Cambridge, MA.

Chang, C., & Gaudiano, P. (1997). Neural competitive maps for reactive and adaptive navigation. In *Proceedings of the 2nd International Conference on Computational Intelligence and Neuroscience* pp. 19–23 March 1997, Research Triangle Park, NC.

Dorman, C., & Gaudiano, P. (1994). Motivation. In Arbib, M. (Ed.), *Handbook of brain theory and neural networks*. MIT Press, Cambridge, MA. In press.

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 97* pp. 13–18 July 1997, Monterey, California, U.S.A.

Gaudiano, P., Zalama, E., Chang, C., & López Coronado, J. (1996a). A model of operant conditioning for adaptive obstacle avoidance. In Maes, P., Mataric, M. J., Meyer, J., Pollack, J., & Wilson, S. W. (Eds.), *From Animals to Animats 4*, pp. 373–381 Cambridge, MA. MIT Press.

Gaudiano, P., Zalama, E., & López Coronado, J. (1996b). An unsupervised neural network for real-time, low-level control of a mobile robot: noise resistance, stability, and hardware implementation. *IEEE SMC*, *26*, 485–496.

Grossberg, S. (1971). On the dynamics of operant conditioning. *Journal of Theoretical Biology*, *33*, 225–255.

Grossberg, S. (1972). A neural theory of punishment and avoidance, II: Quantitative theory. *Mathematical Biosciences*, *15*, 195–240.

Grossberg, S. (1982). A psychophysiological theory of reinforcement, drive, motivation and attention. *Journal of Theoretical Neurobiology*, *1*, 286–369.

Grossberg, S., & Levine, D. (1987). Neural dynamics of attentionally modulated Pavlovian conditioning: blocking, interstimulus interval, and secondary reinforcement. *Applied Optics*, *26*, 5015–5030.

Grossberg, S., & Schmajuk, N. A. (1987). A neural network architecture for attentionally-modulated Pavlovian conditioning: Conditioned reinforcement, inhibition, and opponent processing. *Psychobiology*, *15*, 195–240.

Grossberg, S. (Ed.). (1982). *Studies of Mind and Brain: neural principles of learning, perception, development, cognition and motor control.* Reidel, Boston.

Grossberg, S. (Ed.). (1989). *Neural Networks and Natural Intelligence.* MIT Press, Cambridge, MA.

Khatib, O. (Ed.). (1989). Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, *5*, 90–98.

Latombe, J.-C. (1990). *Robot Motion Planning.* Boston: Kluwer Academic Publishers.

Miller, W. T., Sutton, R. S., & Werbos, P. J. (Eds.). (1990). *Neural networks for control.* M.I.T. Press, Cambridge, MA.

Pfeifer, R., & Verschure, P. (1992). Distributed adaptive control: a paradigm for designing autonomous agents. In Varela, F. J., & Bourgine, P. (Eds.), *Toward a practice of autonomous systems*, pp. 21–30. MIT Press, Cambridge, MA.

Zalama, E., Gaudiano, P., & López-Coronado, J. (1995). A real-time, unsupervised neural network model for the low-level control of a mobile robot in a nonstationary environment. *Neural Networks*, *8*(TR-94-002), 103–123.