Learning courses of action using the *"movie-in-the-brain"* paradigm

Rodrigo Ventura and **Luis Custódio** and **Carlos Pinto-Ferreira**
Institute for Systems and Robotics
Instituto Superior Técnico, Technical University of Lisbon
Av. Rovisco Pais, 1
1049-001 Lisboa, PORTUGAL
{yoda,lmmc,cpf}@isr.ist.utl.pt

Abstract

To exhibit social competence, an agent ought to possess the ability of deciding adequate courses of action when confronted with other agents. In this decision-making process the agent has to evaluate the behavior of other agents (and objects) along time in response to its actions and to act in order to raise the desirability of the resulting situation. To achieve this desideratum, the acting agent should learn causal relationships associating its actions to the responses got from the environment. In this paper the concept of "movie-in-the-brain" (MITB) mentioned by Damásio (Damásio 1999) was applied to the development and the implementation of a mechanism capable of establishing such causal relationships. A very simple example of an agent supervising a controller of an inverted pendulum illustrates the application of the MITB paradigm as a way of learning cause-effect relations and of improving the competence of the agent along time. Some interesting preliminary results are presented and discussed.

Introduction

The choice of a particular course of action by a situated agent depends on the desirability of the expected final results. Each time the agent makes a decision (and act), it should evaluate whether the response got from the environment corresponds to a situation which evolves towards the intended objectives. According to António Damásio, the mechanism of deciding courses of action relies on what he calls a "movie-in-the-brain" (MITB): the agent stores chunks composed by sequences of perceptions and actions, together with a measure of the corresponding desirability. When a similar situation appears in the future, the agent can make decisions on the basis of a pre-existing experience. From the point of view of developing and implementing intelligent agents, this mechanism has some interesting advantages and some drawbacks. On the one hand, the competence of the agent improves as it collects new experiences, so the required *a priori* knowledge is low. On the other hand, as the MITB relies on learning cause-effect relations, the agent exhibits sometimes a "superstitious behavior." Learning, in this case, presupposes a bootstrapping mechanism rooted on a reactive approach to action, to be utilized whenever the agent has lack of experience. It also presupposes to experiment and make, in certain cases, fatal mistakes.

The complexity of implementing a MITB has led the authors to the choice of a simple example (the supervision of the controller of an inverted pendulum) to serve as test bed for the experimentation of the MITB paradigm. Moreover, the MITB was built on the foundations of the DARE architecture for emotion-based agents. In the next section the DARE architecture is briefly described. Followed by a section explaining the proposed memory mechanism. Then the developed implementation is described followed by some preliminary results, and a comparative study with other models. Finally, the paper ends with some concluding remarks.

DARE: an emotion-based agent architecture

The DARE architecture was introduced in (Ventura & Pinto-Ferreira 1998) and further developed and implemented in (Ventura, Custódio, & Pinto-Ferreira 1998; Ventura & Pinto-Ferreira 1999; Ventura 2000; Maçãs *et al.* 2001). Applications to control systems were discussed in (Custódio, Ventura, & Pinto-Ferreira 1999), and to learning in (Vale & Custódio 2001).

The basic idea of DARE is to process stimuli simultaneously under two different perspectives: a *cognitive*, elaborative — which allows the agent to understand what is happening and what it knows about the world, and a *perceptual*, immediate — which permits it to react quickly and decide adequately in circumstances demanding urgent action. Hence, from the very same stimulus, two sets of facets are extracted: one, mostly directed to recognition and reasoning purposes, and another, aiming at assigning degrees of "threat," "danger," "pleasure," and so on, to the current situation, constructing what we call a *desirability vector* (DV).

This kind of system should be bootstrapped by the incorporation of built-in associations. In fact, there should exist some stimuli which are *essential*, innate: for instance, animals faced with their preys or predators decide either to attack or run away as a function of the vector of desirability a perceptual image suggests. This assignment should depend on the considered species.

Formally, at a given time instant t the agent is exposed to a stimulus denoted $s(t)$. From this stimulus two images are extracted: a *perceptual image* $i_p(t) = f_p(s(t))$ and a *cognitive image* $i_c(t) = f_c(s(t))$. From the perceptual image, a *desirability vector* (DV) $v_d(t) = f_d(i_p(t))$ is obtained, whose components represent the assessment the agent performs across several dimensions (e.g., valence, relevance, etc.). The built-in associations that the agent possesses correspond to the functions f_p and f_d. The figure 1 illustrates this architecture.

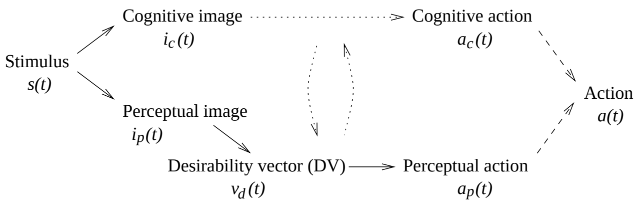

Figure 1: Simple sketch of the DARE architecture.

For a given DV, the perceptual layer obtains a reactive response action $a_p(t) = f_{ap}(v_d(t))$. Simultaneously, as a result of the cognitive processing, which treats the cognitive image on the basis of pattern matching, an action $a_c(t)$ may be obtained. The action $a(t)$ taken by the agent is either the perceptual or the cognitive action depending on the agent's assessment of the need to answer quickly to the stimulus.

Motivation and relevance to social contexts

In general terms the knowledge of a situated agent can be divided in two broad classes: *built-in* knowledge that allows the agent to cope with the environment at a basic level of competence (e.g. to be able to avoid situations that endanger its basic survivability), and *learned* knowledge acquired with the interaction with the environment. In complex and dynamic environments built-in knowledge does not suffice, given either the complexity or the unpredictability of the environment.

Learning from interaction with the environment is here understood as gathering cause-effect relationships. Given a situation the agent is facing, it should be capable to answer the question "what happens if I do this action?" Or in other words, the ability to anticipate the effects of some action.

In order to obtain these cause-effect relationships, a mechanism was added to the DARE architecture. This mechanism is based on Damásio's work (Damásio 1999), where he describes a "movie-in-the-brain" mechanism as a "rough metaphor [which] has as many sensory tracks as out nervous system has sensory portals — sight, sound, taste, and olfaction, touch, inner senses, and so on." ((Damásio 1999) page 9).

In this sense, the agent stores sequences of snapshots of its interaction with the environment in a structure called *movie-in-the-brain (MITB)*. As this interaction happens, the agent has no idea of what are the relevant sequences of snapshots it should store. Therefore, it stores as many sequences as it can.

Modeling other agents behavior is a very important feature in social contexts. However, this task can easily become an intractable task, since they can be arbitrarily complex. Furthermore, an agent has access mainly to external manifestations of behaviors. The MITB can play an important role in this context, in the sense of providing a mechanism for establishing cause-effect relationships. These include the effects on the world of an agent actions, as well as, an agent responses to world changes (including any other agent actions). An agent equipped with the ability to extract these cause-effect relationships can become able to anticipate possible effects to known causes.

Taking the scenario of a robotic soccer game as an example, there are many causes to be loosing a game for many goals. Since the MITB accumulates all the details of previous experiences, an agent might be able to identify patterns, e.g. common situations between suffered goals, and therefore to avoid them. This can be accomplished by anticipating the future behavior of the opponents, and by taking measures against it. The danger behind the opponent possession of the ball lies mostly on what can happen afterwards, rather than on the mere possession.

Implementation of the MITB

Experimental setup

In order to understand how the MITB mechanism could function, a simple testbed was developed.

The testbed consists of a supervision of a simple control system. The system to be controlled is a dynamic model of an inverted pendulum coupled to a movable car (figure 2). This is a typical textbook problem in control.

The model for the setup is a non-linear dynamic system obtained from the physical considerations. The state of the system can be described by a four variable state vector $\mathbf{x}(t) = (x(t), \dot{x}(t), y(t), \dot{y}(t))$, where $x(t)$ and $\dot{x}(t)$ are the translational position and velocity of the car, and $y(t)$ and $\dot{y}(t)$ are the angular position and angular velocity of the pendulum. The state trajectories along time are described by an equation in the form $\dot{\mathbf{x}}(t) = F(\mathbf{x}(t), u(t))$, where $u(t)$ denotes the force applied to the car, which is the actuation signal supposed to balance the pendulum, and F is a function describing the system dynamics.

The system described above is simulated by a simple numerical integration (4th order Runge-Kutta method) of the dynamic equations, where the simulation step is fixed.

This system is controlled by a simple proportional controller given by $u(t) = K_p [y_{ref} - y(t)]$, where y_{ref} is the desired angular position of the pendulum (e.g., vertical position) and K_p is a parameter called the *proportional gain* of the controller. This implicitly assumes that we are only interested in the vertical equilibrium of the pendulum, regardless of the car speed. This is in

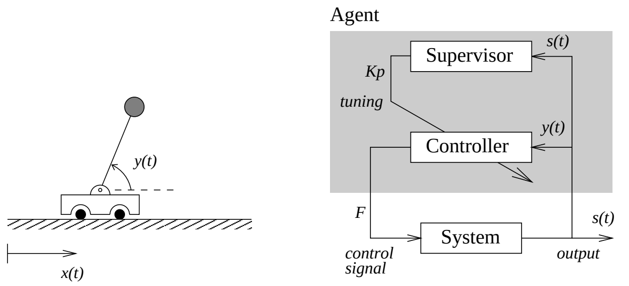

Figure 2: On the left, the system used in the testbed, and on the right, the supervisor setup.

fact a strong simplification, since when the pendulum is successfully balanced, the car may keep moving.

A control system supervisor (figure 2) is built by adding a module — the supervisor — which observes the state of the system, and tunes the controller parameter. In this case, the supervisor and the controller constitute the agent, whose stimuli are the system state and whose actions are the new controller parameter (proportional gain).

To get a realistic flavor of this setup, picture an agent watching the pendulum and measuring the objects positions and velocities by means of its sensors, and trying to balance the pendulum by exerting a force on the car. In this metaphor the controller is assumed to be part of the agent, e.g. a low-level reactive layer.

Supervisor

Fundamentals. For each simulation step the supervisor is stimulated with the state of the system, which corresponds to the stimulus $s(t) = \mathbf{x}(t)$. Then, according to the architecture previously described, the cognitive and perceptual images are extracted. In this implementation, the cognitive image equals the state vector, $i_c(t) = \mathbf{x}(t)$ (f_c is in this case the identity function), whereas the perceptual one has two components, $i_p(t) = f_p(x(t)) = (i_p^1, i_p^2)$. These components are the deviation between the pendulum angular position and the vertical position (equilibrium) $i_p^1 = y - y_{ref}$, and the sum of the absolute speeds of the car and pendulum $i_p^2 = |\dot{x}| + |\dot{y}|$.

The DV components represent basic assessments of stimuli desirability. In this case there are two components, $v_d(t) = (v_d^{val}, v_d^{urg})$, which denote *valence* ($v_d^{val} \in [-1, 1]$, positive if $v_d^{val} > 0$, neutral if $v_d^{val} = 0$, and negative otherwise) and degree of *urgency* ($v_d^{urg} \in [0, 1]$, 1 means maximum urgency).

The mappings f_d between the i_p (perceptual image) and the v_d (DV) are decomposed on two linear piecewise functions shown in figure 3.

For each time step the agent stores a memory frame $m_f(t) = \langle i_c(t), i_p(t), v_d(t), a(t) \rangle$ into the MITB. The MITB is a sequence of memory frames $\mathcal{M}(t) = [m_f(t_1), m_f(t_2), \ldots]$, where $t_k < t$ for $k = 1, 2, \ldots$ represent the time instants in which the agent have received

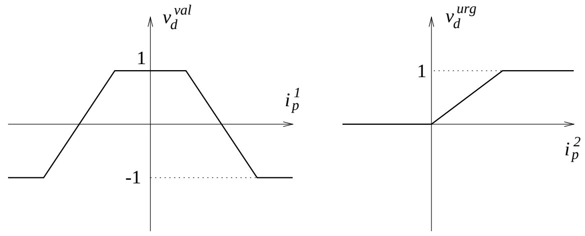

Figure 3: Profiles of the functions used to obtain the DV components.

the stimuli, up to time t.

In the following, it is always assumed a generic time t and, therefore, t will be dropped for the sake of clarity.

Topographic map. The action selection is based on a mechanism called *topographic maps* (abbreviated to topmap), inspired on an homonymous structure found in the brain (Churchland & Sejnowski 1992). The biologically inspired idea of topographic maps has been used in neural networks (Kohonen 1982), among other areas. However, in the context of this work, the idea of topographic maps was taken in a different, simpler perspective.

The topmap is a function $\mathcal{T}(x) \in \mathbb{R}$ defined in a bounded interval $x \in [x_{min}, x_{max}]$. This function is obtained by combining a set of "building-block" functions $\psi(x)$ defined and parameterized as follows:

$$\psi(x; x_0, A, \tau) = A \cdot e^{-\tau \frac{|x - x_0|}{x_{max} - x_{min}}}$$

where x_0, A, and $\tau > 0$ are parameters. This function equals A for $x = x_0$ and decays exponentially with $|x - x_0| > 0$, with a decaying coefficient τ. The idea of the topmap is to find the argument x that maximizes this function, where ψ functions with positive A contribute as "attractors" and negative A as "repulsors."

For a set of contributions $\left\langle x_0^{(n)}, A^{(n)}, \tau^{(n)} \right\rangle$, with $n = 1, \ldots, N$, a topmap is initialized with zero, $\mathcal{T}^{(0)}(x) = 0$. For each contribution (n), a slighly modified function $\psi^{(n)}(x)$ is added (pointwise) to $\mathcal{T}^{(n-1)}(x)$. The modification consists of adjusting the amplitude parameter $A^{(n)}$ in such a way that $\mathcal{T}^{(n)}(x_0) = \psi(x_0^{(n)}; x_0^{(n)}, A^{(n)}, \tau^{(n)}) = A$. In other words, to prevent that many ψ functions centered at some x_0 overload the topmap. Formally, this modification consists of transforming a contribution $\psi(x; x_0^{(n)}, A^{(n)}, \tau^{(n)})$ into:

$$\psi^{(n)}(x) = \psi(x; x_0^{(n)}, A^{(n)} - \mathcal{T}^{(n-1)}(x_0^{(n)}), \tau^{(n)})$$

The parameters x_0, A, and τ parameterize each contribution individually. However, in this implementation, all A and τ are equal. Topmaps are implemented by discretizing the interval $[x_{min}, x_{max}]$ in equally spaced (small) steps in x.

Decision making. Given a MITB containing a representation of the agent recent history, the agent performs the following steps to reach a decision at the cognitive layer:

1. *Match the present cognitive image i_c against the MITB.* The metric used in this match is a normalized Euclidian distance. Given two vectors $\mathbf{u} = (u_1, ..., u_N)$ and $\mathbf{v} = (v_1, ..., v_N)$ the distance between them is given by
$$d(\mathbf{u}, \mathbf{v}) = \sqrt{\frac{1}{N} \sum_{k=1}^{N} \left(\frac{v_k - u_k}{v_k^{max} - v_k^{min}} \right)^2}$$
provided that v_k^{min} and v_k^{max} are the minimum and maximum of the k components of all cognitive images in the MITB.
This metric accommodates for unknown vector components scaling, constrained by knowing *a priori* the normalization parameters v_k^{min} and v_k^{max}.
This matching process assigns to each memory frame m_f^M in the MITB, with cognitive image i_c^M, a match degree given by $d(i_c^M, i_c)$.
2. *Find local minima of the matching degrees.* In this application, since the changing rate of the system state is relatively slow when compared with the time step, the local minima (when looking at the MITB along the time) seems a reasonable mechanism to find a tractable small set of cognitive matches. However, it may not be as appropriate in other domains. The local minima are a subset of memory frames for which the matching degree is less or equal than the ones of the predecessor and the successor (whenever more than one memory frame satisfy this condition, just the most recent in time is considered).
3. *Pick a sub-sequence after each cognitive match.* The sub-sequence of memory frames following a cognitive match represents the immediate future after the agent was previously faced with a similar stimulus. This sub-sequence is also dependent on the actions the agent took during that corresponding period of time. This sequence of associations between cognitive and perceptual images, DVs and actions are the basis for the agent decision making process. There is a fixed parameter that limits the size of each sub-sequence.
4. *Evaluate each sub-sequence.* For the first frame, the DV and action are extracted (v_d^1 and a^1), then the first next frame for which the DV changes (in vector distance, with respect to v_d^1) more than a threshold is searched for (v_d^2 and a^2), *i.e.,* $\|v_d^2 - v_d^1\|$ greater than a threshold. If no such change is found, this sub-sequence is ignored. The amount of change is obtained from a weighted sum of the DV components difference $e_{ch} = \sum_k w_k \cdot \left[(v_d^2)_k - (v_d^1)_k\right]$.
For this implementation, the weights w_k determine to what extent the agent take into account the valence or the urgency components of the DV.
5. *Construct action topmaps with respect to the "ignorance" and to the "evaluation."*
Two topmaps are constructed: one called *ignorance*, $\mathcal{T}^{ign}(x)$, representing the degree of ignorance of the effects of a certain action $x = a(t)$, and another called *evaluation*, $\mathcal{T}^{eval}(x)$, representing whether the agent considers the effects of the action $x = a(t)$ desirable or not (positive values mean "desirable," while negative ones mean "undesirable").
The "ignorance" topmap is obtained just by combining a ψ function for the action of each cognitive match (and local minimum)m and the "evaluation" topmap is obtained in a similar fashion, but now the amplitude of the ψ function depends on the evaluation e_{ch}.
6. *Choose the "appropriate" action.* At this stage the agent has to decide whether to maximize ignorance (exploration of the environment) or evaluation (exploitation). First, the ignorance topmap is maximized: $i_{max} = \max_x \mathcal{T}^{ign}(x)$. Then, if i_{max} exceeds a threshold T_I, the agent chooses to try a different action (exploration). Otherwise, the agent chooses the action which maximizes the evaluation topmap: $a_c = \arg\max_x \mathcal{T}(x)$, where $\mathcal{T}$ is $\mathcal{T}^{ign}$ if $i_{max} > T_I$, or $\mathcal{T}^{eval}$ otherwise.

As explained previously, the agent action a is one of a_p (perceptual) or a_c (cognitive), depending on the urgency of response. In this implementation a threshold T_U is used, such that if the DV component $v_d^{urg} > T_U$, the perceptual action is chosen ($a = a_p$), otherwise, the cognitive one is chosen ($a = a_c$). The function f_{ap} that obtains the perceptual action a_p is 200 if $v_d^{val} < 0$, and 0 otherwise. This function simply turns off the controller when the valence component of the DV is positive, and uses a relatively large value otherwise. This results in a bang-bang kind of control. The objective of this choice is to show the benefits of the cognitive layer on top of a perceptual layer that is too simple to handle an inverted pendulum system.

Experimental results

Perceptual layer only (reactive approach)

In this case the (re)actions are solely result of the built-in associations, incorporated in the perceptual layer. The result is a bang-bang kind of behavior where the agent was unable to prevent the pendulum from falling for the range of initial conditions (0 to 12^o). The figure 4 shows an example of such a run.

Full architecture (pro-active)

These experiments use the full architecture, *i.e.,* cognitive and perceptual layers. The MITB is empty at the start of each run. Typically, during the first instants of time, the agent experiments several actions. After some time, either the system converges to an action that is able to successfully balance the pendulum, or lets the pendulum fall down. Figure 5 shows an example of a

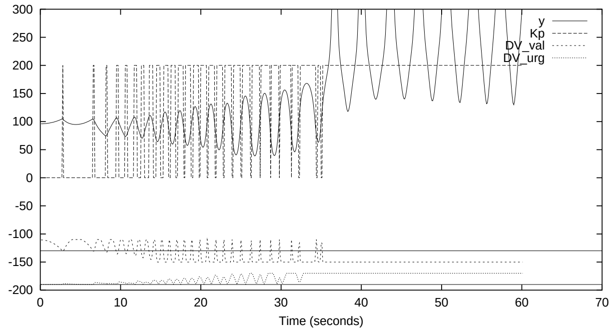

Figure 4: Perceptual layer only, single run for 6^o initial deviation. The "y" labeled trace represents the angular position, in degrees (vertical position equals 90^o); the "Kp" represents the controller gain K_p, which is the action $a(t)$ of the supervisor; the values for each of the DV components v_d^{val} and v_d^{urg} are also shown as "DV_val" and "DV_urg" over horizontal lines denoting the zero of each component.

run where the agent successfully balances the pendulum.

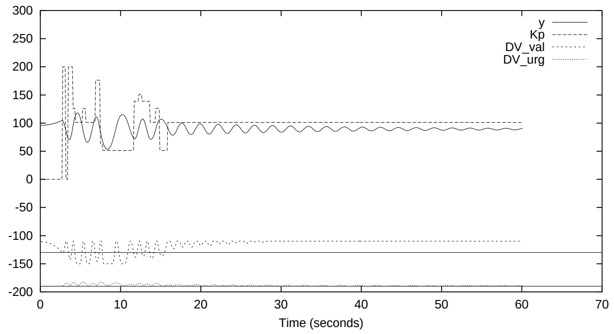

Figure 5: Full architecture, single run at the same conditions as in figure 4.

The results of a batch of 151 runs, for the range of initial conditions previously described, are the following: the pendulum fell down 29.8% of the trials, and settled (within $\pm 5^o$ margin) for 35.8% of the trials (in the remaining 34.4% the pendulum did not settled for 60 seconds); the average settling time was 39.2 seconds, with a standard deviation of 9.59 seconds.

Persistence of the MITB

For each initial condition, the MITB was first emptied, and then followed by a sequence of ten runs, each one appending memory frames to the MITB. The idea consists of observing whether the agent behavior improves as the MITB registers increasingly more experience. The results are the following, for a batch of 151 runs: the pendulum fell down 30.5% of the trials, and settled for 56.3% of the trials; the average settling time was 16.1 seconds, with a standard deviation of 13.6 seconds. Comparing with the previous experience, the average settling time improved significantly, as well as the number of runs the pendulum was successfully settled.

For a given initial condition, the first run equals the corresponding one in the previous experiments. However, for the second run, the results tend to worsen significantly. Only after several runs the results improve. One possible explanation for this behavior is that in the second run the agent tends to try out completely different actions, for the similar situations it encounters, possibly leading to worse results. However, as experience is accumulated in the MITB, the agent starts using that information to choose actions aiming at desirable situations.

A relevant feature of these experiments is that the settling time is in general inferior to the ones in the previous experiments. This shows that accumulated experience in the MITB contributes positively to a better performance of the agent.

Comparison with other models

There is a broad range of very different approaches to control problems. To name a few, Albus (Albus 1996) proposes a top-down approach, based on the desirable capabilities of a complete intelligent system, and going through an hierarchy of mechanisms, down to the low-level control loops. A radically different approach is taken by Arbib (Arbib, Schweighofer, & Thach 1994), for instance, which proposes a biologically based approach, beginning by understanding the low-level mechanisms of sensori-motor coordination, and going upwards in terms complexity.

Michie and Chambers (Michie & Chambers 1968) performed an early work on balancing a pendulum using an adaptive system. A later work by Barto *et al.* (Barto, Sutton, & Anderson 1983) approaches the same problem using reinforcement learning.

There are several qualitative distinctions between the above RL approach and the emotion-based one: first, the later bootstraps using the perceptual layer, containing a minimal ability to attempt balancing the pendulum, while the former is bootstrapped with random neural weights; second, the MITB stores knowledge in an explicit fashion, while the neural weights is implicit; and third, the RL approach learns via an exhaustive experimentation, while the later attains reasonable performance just after a few steps.

Given the availability of the C code used by Barto *et al* (Barto, Sutton, & Anderson 1983) on the web[1], some comparative experiments were conducted. The RL experiment is based on a sequence of trials, starting with the $(0, 0, 0, 0)^T$ initial state and ending with the pendulum falling or the car hitting the walls, while the emotion-based one was performed as described in the previous section.

Figure 6 shows the results for 40 trials in terms of number of steps for which the pendulum does not fall

[1] `ftp://ftp.cs.umass.edu/pub/anw/pub/sutton/pole.c`

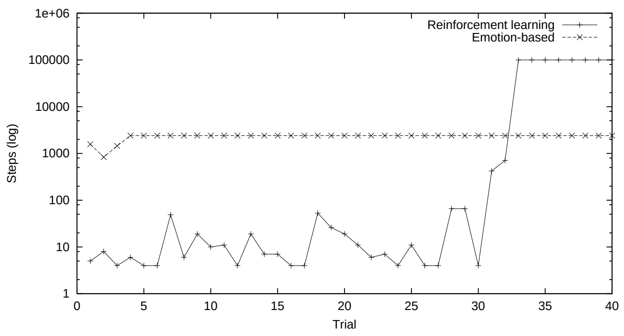

Figure 6: Comparative results of the proposed architecture and the one proposed by Barto *et al.* (Barto, Sutton, & Anderson 1983) for 40 trials.

and does not hit the wall for the RL approach. One first salient feature is that the RL approach requires about 30 trials until a reasonable performance is attained. This is a consequence of two factors: first, the neural weights are initialized to a random value, while in the emotion-based one the perceptual layer provides a basic tendency to balance the pendulum; and second, the emotion-based approach is able to put into practice what it learned in real-time, instead of relying on external reinforcements or waiting for convergence of the neural weights. Only after enough trials for the RL process to converge, it is able to outperform the emotion-based one.

Conclusions and Future Work

This paper shows some interesting results of an implementation of the "movie-in-the-brain" idea introduced by Damásio (Damásio 1999). This mechanism is used to select courses of action aiming at obtaining desirable states for the agent. The results show that the agent has a tendency to try out several actions at the beginning of each run. As the agent accumulates experience over time, in the MITB, the behavior is more consistent, showing a trend to converge to an appropriate action, *i.e.,* that successfully balances the pendulum, in a shorter amount of time, when compared with the beginning of a run.

However, the computational resources of agents are usually limited. In order to cope with complex and dynamic environments, a MITB as described would result in an intractable large "movie." Therefore, a long-term memory mechanism has to be addressed in the future.

Acknowledgments

The authors are grateful for the suggestions and comments made by Sandra Gadanho on an early version of this paper.

References

Albus, J. S. 1996. The engineering of mind. In *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB-96)*, 23–32. The MIT Press.

Arbib, M. A.; Schweighofer, N.; and Thach, W. T. 1994. Modeling the role of cerebellum in prism adaptation. In *Proceedings of the Third International Conference on Simulation of Adaptive Behavior (SAB-94)*, 36–44. The MIT Press.

Barto, A. G.; Sutton, R. S.; and Anderson, C. W. 1983. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on System, Man, and Cybernetics* SMC-13(5):834–846.

Churchland, P. S., and Sejnowski, T. J. 1992. *The Computational Brain.* The MIT Press.

Custódio, L.; Ventura, R.; and Pinto-Ferreira, C. 1999. Artificial emotions and emotion-based control systems. In *Proceedings of 1999 7th IEEE International Conference Emerging Technologies and Factory Automationm (ETFA'99)*, volume 2, 1415–1420. IEEE.

Damásio, A. R. 1999. *The Feeling of What Happens: body and emotion in the making of consciousness.* Harcourt Brace.

Kohonen, T. 1982. Self-organized formation of topologically correct feature maps. *Biological Cybernetics* 43:59–69.

Maçãs, M.; Ventura, R.; Custódio, L.; and Pinto-Ferreira, C. 2001. Experiments with an emotion-based agent using the dare architecture. In *Proceedings of the Symposium on Emotion, Cognition, and Affective Computing, AISB'01 Convention.*

Michie, D., and Chambers, R. A. 1968. BOXES: An experiment in adaptive control. In Dale, E., and Michie, D., eds., *Machine Intelligence 2.* Edinburgh: Oliver and Boyd. 137–152.

Vale, P., and Custódio, L. 2001. Learning individual basic skills using an emotion-based architecture. In *Proceedings of the Symposium on Emotion, Cognition, and Affective Computing, AISB'01 Convention.*

Ventura, R., and Pinto-Ferreira, C. 1998. Emotion-based agents. In *Proceedings AAAI-98*, 1204. AAAI.

Ventura, R., and Pinto-Ferreira, C. 1999. Emotion-based agents: Three approaches to implementation (preliminary report). In Velásquez, J., ed., *Workshop on Emotion-Based Agent Architectures (EBAA'99)*, 121–129.

Ventura, R.; Custódio, L.; and Pinto-Ferreira, C. 1998. Emotions — the missing link? In Cañamero, D., ed., *Emotional and Intelligent: The Tangled Knot of Cognition*, 170–175.

Ventura, R. 2000. Emotion-based agents. Master's thesis, Instituto Superior Técnico, Technical University of Lisbon, Lisbon, Portugal. (URL: http://www.isr.ist.utl.pt/~yoda/aims/papers/ventura-msc.ps.gz).