Actor-Critic Models of Animal Control - A critique of reinforcement learning

Florentin Wörgötter
Department of Psychology, University of Stirling, Stirling FK9 4LA, Scotland,
worgott@cn.stir.ac.uk

Abstract

In this article we will compare traditional reinforcement learning techniques with a novel correlation based algorithm. We will discuss several problems which occur in reward-based reinforcement learning and outline alternative solutions. An example of a robot control task shown at the end will support our claims.

1 Introduction

Control tasks for animals or machines ("systems") require sensible actions that follow from the current state of the system. As a consequence the state of the system may change and a new action will be elicited, and so on. This procedure describes a sequence of states and actions which follow each other in time. Such control mechanisms can either be hardwired into the system, but many times it is more appropriate to design a learning algorithm that tries to infer the next action from the previous sequence of states. Especially in animal (or robot-) control the complexity of the world may prevent hardwiring and learning is required instead to assure enough flexibility.

Currently there is only one group of algorithms widely used to this end: reward-based reinforcement learning (Sutton and Barto, 1998). The central paradigm for all of these algorithms is that a system learns the most sensible actions by performing that particular action from which it receives maximal reward.

Substantial efforts have been undertaken during the last 15 years to develop a highly successful theory of reinforcement learning (Sutton, 1988; Dayan, 1992; Dayan and Seynowski, 1994; Kaelbling et al., 1996; Sutton and Barto, 1998). However, in spite of this, convincing proofs of the usefulness of reinforcement learning in technical applications are still rare. In general one finds that these

algorithms are not very successful in space- and time-continuous control tasks, to which all animal and robot control tasks and many industrial ones belong. This article will discuss possible reasons for this problem and we will try to provide an alternative solution, where many of these problems do not occur.

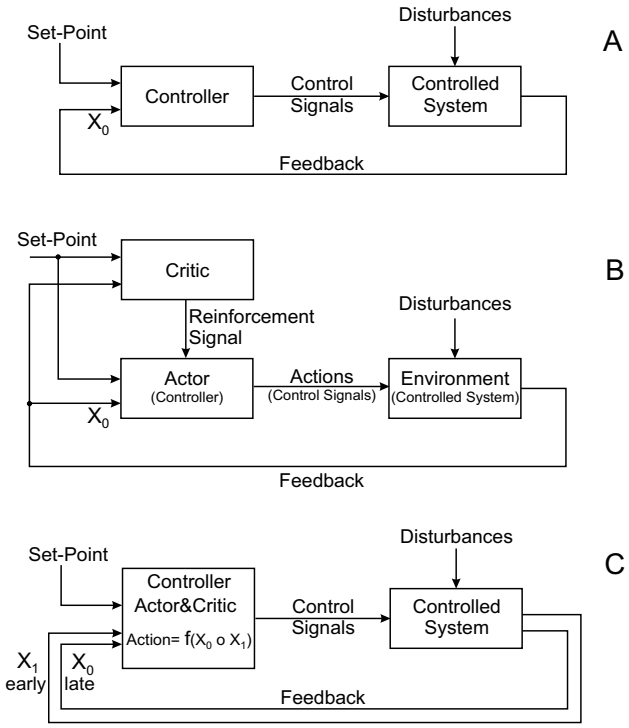

Figure 1: Actor-Critic architecture. A) Conventional feedback loop controller, B) Actor-Critic control system, where a Critic influences action selection by means of a reinforcement signal. C) Correlation based Actor-Critic architecture, where the control signal is derived from the correlations between two temporally related input signals.

2 Actor-Critic models

Reinforcement learning control-applications are usually embedded in so called Actor-Critic architectures and as such these models are strongly related to control theory (Witten, 1977; Barto et al., 1983; Sutton, 1984). Fig. 1 A shows a conventional feedback control system. A controller provides control signals to

a controlled system which is influenced by disturbances. Feedback allows the controller to adjusts it signals. In addition, a set-point is defined. In the equilibrium (without disturbance), the feedback signal X_0 will take the negative value of the set-point, which represents the "desired state" of the complete system. In the simplest case (set-point=0), this is zero, too. The set-point can essentially be associated with the control goal of the system, reaching it by means of the feedback could be interpreted in the way that the system has attained homeostasis. Part B of this figure shows how to extend this system into an actor-critic architecture. The critic produces evaluative, reinforcement feedback for the actor by observing the consequences of its actions. Most of the time temporal difference learning (TD-learning, Sutton 1988) is used to create the reinforcement signal.

TD-learning essentially calculates the error δ between a the predicted value of a state s and the actually observed (currently existing value), holding it against the future return R. One defines:

$$R_T = \sum_{t+1}^{T} r(t), \tag{1}$$

$$\delta(t) = R_T - V_s(t), \tag{2}$$

$$V_s(t) \leftarrow V_s(t) + \delta(t) \tag{3}$$

where R_T is called *return* and represents the total future reward when performing an action sequence that starts at state s. V_s is the momentarily existing *value* of state s. The prediction error δ gives the mismatch of the currently assumed value V_s as compared to the expected return. The error is zero if these two quantities match. In this case the correct value V has already been associated to its state s. Thus, the update rule corrects the momentarily existing value by means of adding the error: If we expect more return, V_s should rise, otherwise it should fall. Recursive methods like the TD(0) algorithm exist to calculate this in an efficient way (Sutton and Barto, 1998).

The Critic takes the form of a TD-error which gives an indication if things have gone better or worse than expected with the preceding action. Thus, this TD-error can be used to evaluate the preceding action: If the error is positive the tendency to select this action should be strengthened or else, lessened. Thus, actor and critic are adaptive through reinforcement learning. This relates these techniques to advanced model-based feed-forward control and feed-forward compensation techniques. Many different ways exist to actually implement this (see Sutton and Barto (1998) for an example) and actor-critic architectures have become especially influential when discussion animal control. Several authors have suggested

that actor-critic architecture might describe decision making sub-systems in animals. Accordingly, several models have been designed which try to capture the functionality of the basal ganglia in terms of Actor-Critic architectures (Houk et al., 1995; Suri and Schultz, 1998; Berns and Sejnowski, 1998; Brown et al., 1999; Contreras-Vidal and Schultz, 1999; Suri et al., 2001). To discuss these models here would, however, exceed the scope of this article (see Joel et al. 2002; Wörgötter and Porr 2004)

3 Are reinforcement Actor-Critic architectures realistic in view of animal control

In this section we will ask which requirements must be met in order to successfully implement an Actor-Critic architecture for animal control.

Before learning can start we must make sure that: 1) The environment must provide feedback to the system and 2) the system must be able to explore the environment while it should at the same time maximize its return during learning.

3.1 Evaluative versus non-evaluative feedback - The credit structuring problem

The *credit structuring problem* especially "haunts" many practical applications. TD-methods (and their relatives) always rely on rewards $r(t)$. This, however, requires a process that "places" the rewards appropriately (credit structuring problem). To this end we need an external observer who has some prior knowledge about the structure of the problem and about the goal of the prediction (or control-) problem.

Only in very simple cases credit-structuring can be done directly. Most of the time real-world MPD-like problems are too complex to allow for hand-placing rewards. In addition, many times goals are at first only generally known. For example, the goal "learn to win in chess", hence placing a reward at the end of a successful game, will lead to unacceptably long times to convergence (if at all) of learning.

Credit-structuring is a major challenge especially in time- and space-continuous control tasks, to which all animal behavioral tasks belong, where there are an infinite number of state-action pairs, to which rewards or punishments would have to be associated beforehand by defining a mapping function from the state-action

space to the reward/punishment space defining the so-called "reinforcement function" (Santos and Touzet, 1999a,b), which requires appropriate structural credit assignment also during learning (Mahadevan and Connell, 1992).

Such procedures for credit structuring can be called *external evaluative feedback*: An external structure explicitly provides the rewards.

A better strategy may be to assume that the agent/algorithm performs credit structuring on its own. This can be achieved, for example, by associating only very general aversive or attractive properties (like good or bad taste of food, pain or pleasure, etc.) to states and let the agent "experience" them via some sensor inputs. We would call such a procedure *internal evaluative feedback*: The agent itself provides the value of the rewards. There is, however, a hitch. We still need an external structure which explicitly defines the reference frame for what should be rewarding (or punishing). But this is easy, isn't it? For example: Food is rewarding, pain not. However, the rewarding properties of food will for an animal very much depend on its internal and non-observable state of satiety. It may depend on its tiredness, on its mating-drive if the opposite sex is present, on its fear in how to approach the food, etc., etc. Even pain is relative and situation dependent. All these variables are in a normal situation inaccessible to an external observer and it may be just as dangerous to impose ones own external reference frame(s) onto the agent as it is to directly define its rewards. Thus, also internal evaluative feedback poses a problem when trying to design autonomous agents in a complex world with complex, opaque and possibly conflicting internal goals. This problem had initially been pointed out by Klopf (1988) in the context of classical conditioning and it is know earlier as the "frame-problem of AI" from the work of Dennett (1984).

The only possible solution out of this is to try to design a system which operates strictly with *non-evaluative* feedback where any external structure (here, the environment) will only provide value-free signals.

Why is this an important problem? Santos and Touzet (1999b) explain that wrong credit structuring (i.e., defining the wrong reinforcement function) can easily prevent or delay convergence or - even worse - can lead to a convergence to an undesired behavior. Thus, this problem needs to be addressed before any learning can take place. The examples above, however, have clearly shown that this is a non-trivial task.

3.2 How to choose the right starting policy?

Control system built by means of traditional RL methods need a built-in starting policy without which they would not do anything. Here we observe that in real world applications a wrong starting policy may (literally) be lethal for the agent. Evolution on the other hand as made sure that animals start at least with a repertoire of built in behaviors that assures survival. If this problem can be solved for the artificial agent we still find that many RL-algorithms converge with different speed under different starting policies.

3.3 The exploration - exploitation dilemma

The goal of the system is to maximize the return, so naively it should always chose the one action which leads maximal immediate reward. This, however, prevents exploration. Maybe a suboptimal action performed now will lead to a much higher cumulative reward later? As a consequence all RL control-algorithm are faced with the decision to what degree they should balance exploration and exploitation. There are no easy solutions existing to this problem and a large body of literature is devoted to it and the different convergence properties which arise (see for example Bertsekas 1987; Kaelbling 1990; Thrun 1992; Wyatt 1996; Kaelbling et al. 1996; Dearden et al. 1998; Gaskett et al. 1999; Meuleau and Bourgine 1999; Wiering 1999; Smart and Kaelbling 2000; Wyatt 2001; Kearns et al. 2002; Kearns and Singh 2002).

3.4 The credit assignment problem

Finally, we are faced with the situation that rewards and punishments are "rare in the world". Or, more specifically, the controlled agent will normally perform many moves through the state-action space where immediate punishments or rewards are (almost) zero and where more relevant events are rather distant in the future. Nonetheless, to assure that it will learn a good (or even optimal) behavioral strategy, credit would have to be assigned to every *possible* and not only to every actually performed move. In the limit of a continuous state-action space, however, all possible state-action combinations can never be explored, thus credit assignment will always be sparse and - in the worst case - we might miss important state-action pairs. This problem is called the *temporal credit-assignment problem* (Russell and Norvig, 1995; Sutton and Barto, 1998). Thus, credit assignment relies on the actual moves of the agent and the sparse structure of rewards

requires generalization methods (function approximation). It has been observed over and over again that a bad choice of the approximation can strongly delay or even prevent convergence (see Santos and Touzet 1999a,b for a discussion, (Gordon, 1995; Baird, 1995; Tsitsiklis and van Roy, 1996; Bertsekas and Tsitsiklis, 1996) for observations on divergence).

3.5 Maximizing rewards or minimizing disturbances?

Return maximization is the central paradigm of reinforcement learning and this gives rise to the above discussed problems of credit structuring and credit assignment. So far actor-critic architectures have also been implemented adhering to this paradigm. Here we observe, however, that actor-critic architectures would allow implementing a different learning goal as well. Negative feedback loops (Fig. 1 A) have an important homeostatic property: Whenever a deviation from the desired state (set-point) occurs the feedback loop will try to correct it. This property could be used to define a minimization instead of a maximization principle: The learning goal could be to try to minimize deviations from the desired state given by the set-point, i.e., to minimize disturbances of the homeostasis of the feedback loop. These two paradigms are in most cases not equivalent. Most often one finds that maximal return is associated with a single point on the decision surface. Minimal disturbance, however, will cover a whole dense manifold of points all of which represent solutions of the learning problem. As a consequence convergence should be faster and the system will not get stuck in local extrema.

4 Correlation based control is non-evaluative

Fig. 1 C suggest a schematic architecture which accommodates both ideas: 1) non-evaluative feedback and 2) disturbance minimization. This architecture utilizes the basic feedback loop controller from Fig. 1 A, but it assumes that the environment will, in a temporal sequence learning situation, provide temporally correlated signals about upcoming events like those mentioned in the introduction (e.g., smell predicts taste, etc.). This architecture follows the learning goal: Learn to keep the later signal (x_0), whatever it is, *minimal* by employing the earlier signal (x_1) to elicit an appropriate action. In conventional actor-critic architectures, the critic provides an evaluation of the action (e.g. "good" or "bad") and this evaluation influences future action selections. Evaluations however are, as discussed above, always *subjective*. In correlation based control the situation is fundamen-

tally different: Here the system relies on the *objective* difference between "early" and "late", which arises (without internal or external interference) from the structure of the input signals. Evaluations do not take place at this point. Instead the "re-"action of the feedback control loop in response to x_0, be it an attraction or a repulsion reaction, will be shifted forward in time to occur earlier now in response to x_1. Thus, in this system, evaluations do not take place during learning instead they are *implicitly* built into the (sign of the) reaction behavior of the inner x_0-loop: repulsion or attraction. As a consequence, Critic and Actor are not necessarily separate entities anymore and can be merged into the same architectural building block.

The central advantage of Actor-Critic architectures is that they can be set up in a way which is operational from the very beginning if using the reflex loop control configuration (stable starting policy, Millan (1996); Porr et al. (2003), Fig. 1 A). On top of this, correlation based temporal sequence learning (Fig. 1 C) offers the additional advantage that credit-structuring takes place without effort: Every signal which enters the reflex loop will drive the learning and if there is no signal the situation is stable (desirable) for the moment anyway. Rewards do not exist in this scheme.

4.1 Isotropic Sequence Order learning (9)

In this section we discuss a novel algorithm with which correlation based temporal sequence learning can be implemented. This is achieved with a differential Hebbian learning rule (ISO-learning) operating on all synaptic weights (Porr and Wörgötter, 2002, 2003a,b). The main distinguishing features of isotropic sequence order learning (ISO-learning) are: 1) All input lines are treated equal unlike in the older models of Sutton and Barto (1981) and Klopf (1986). 2) Inputs are band-pass filtered 3) Learning is purely correlation based and synapses can grow or shrink depending on the temporal sequence of their inputs. 4) Inputs can take any form of being analogue or pulse-coded.

Fig. 2 A shows the structure of the algorithm for N inputs. Inputs x_k are band-pass filtered (h) prior to summation. Thus, the output of the neuron is given by:

$$v = \sum_{k=0}^{N} \omega_k \bar{x}_k \quad \text{where} \quad \bar{x}_k = h_k * x_k \tag{4}$$

The asterisk denotes the convolution between inputs and band-pass filters.

To get an idea what these filter do, lets consider pulse-inputs. In this case

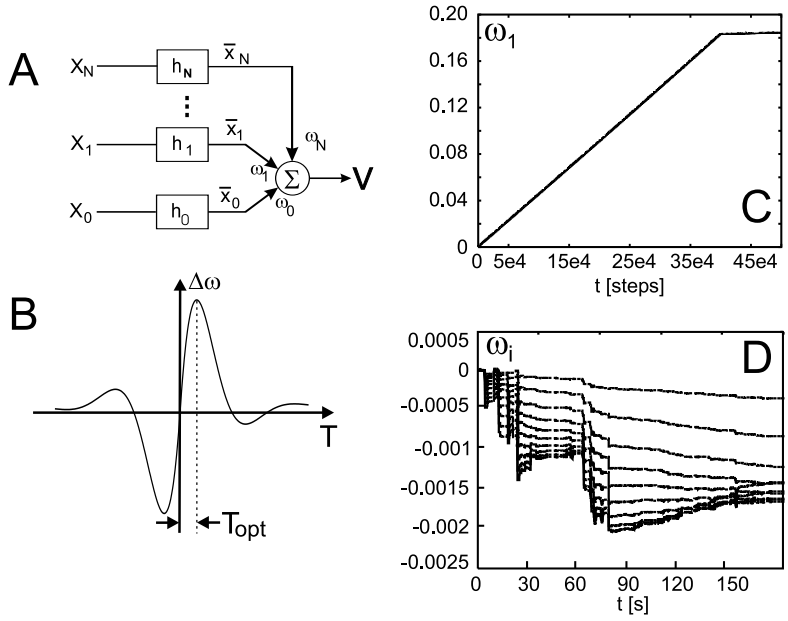

Figure 2: Isotropic sequence order learning. A) Structure of the algorithm for N inputs. For notations see text. A central property of ISO-learning is that all weights can change. B) Weight change curve calculated analytically for two inputs with identical resonator characteristics (h). The optimal temporal difference for learning is denoted as T_{opt}. C) Linear development of ω_1 for two active inputs (x_0, x_1). At time-step 40000 input x_0 is switched off and, as a consequence of the orthogonality property of ISO-learning, ω_1 stops to grow. D) Development of 10 weights ω_1^i, $i = 0 \ldots 9$ in a robot experiment (Porr and Wörgötter, 2003a). All weights are driven by input x_1 but are connected to different resonators h_1^i, which create a serial compound representation (Sutton and Barto, 1990) of x_1 (see Fig. 3 A). The robot's task was obstacle avoidance. At around $t = 150\ s$, it has successfully mastered it, and the input x_0, which corresponds to a touch sensor is not anymore triggered. As a consequence we observe that the weights ω_1^i stop to change (compare to C).

the filtered signals will consist of damped oscillations (Grossberg and Schmajuk, 1989; Grossberg, 1995; Grossberg and Merrill, 1996) which span across some temporal interval until they fade. Thus, band-pass filtering essentially amounts to applying a trace to all inputs, which "stretches" them in time to allow for temporally delayed correlations to take place.

Synaptic weights are changed according to a differential Hebbian learning rule:

$$\frac{d\omega_k}{dt} = \mu \bar{x}_k v' \quad \mu << 1 \tag{5}$$

where v' is the derivative of the output. First we note that the system is linear and weight-changes can be calculated analytically (Fig. 2 B).

It can be shown that inputs will not influence their own synapses and learning is strictly heterosynaptic. As a consequence of this, a very nice feature emerges for pairs of synapses: Weight-change will stop as soon as one input becomes silent (Fig. 2 C). This leads to an automatic self-stabilizing property for the network in control applications (Fig. 2 D).

4.2 Non-evaluative ISO-control: merging Critic & Actor

In this section we will describe how ISO-learning can be used to implement correlation based control introduced in an abstract way in section 4. The central assumption of ISO-control is that any control system should start with a stable negative feedback loop (Fig. 1 C) for example a reflex-loop. Feedback controllers, however, suffer from a major disadvantage: They will always only react *after* a disturbance has taken place (inner loop in Fig. 3 A). Thus, the desired state (e.g. $x_0 = 0$, see also Fig. 1 C) cannot be maintained all the time. Or in other words, disturbances will not yet be minimal when employing feedback control. ISO-control can improve on this if a temporal correlation exist between the primary disturbance and some other earlier occurring signal (denoted by the delay τ between the inner and the outer loop in Fig. 3). The ISO-learning algorithm allows for learning this correlation and, as a result, the primary reflex reaction will be "shifted forward" in time, now occurring earlier; i.e. before the primary reflex would have been triggered. Thus, if learning is successful the primary reflex will be fully avoided and disturbances are now minimal (ideally zero). Note that in the architecture shown in Fig. 3 A Critic and Actor are not anymore separate (compare Fig. 1 B,C).

This principle has been employed in several real-robot experiments (Fig. 3 B,C) which can be viewed at http://www.cn.stir.ac.uk/predictor. We simulate touch and range-finder signals. Before learning the simulated robot will perform a built-in retraction reaction when touching an obstacle (primary reflex reaction). All weights ω_k are initially zero except the weights which belong to the touch sensor inputs, which we set to one. Thus, the output is at this stage just the signal $v = \bar{x}_0$, where $\bar{x}_0$ is the band-pass filtered touch senor input $\bar{x}_0 = h_0 * x_0$. This signal is sent sign-inverted (negative feedback!), but otherwise unaltered, to the motors[1], which leads to a retraction reaction. The range-finders provide the

[1]This description is slightly simplified, because we employ steering and accelerating control,

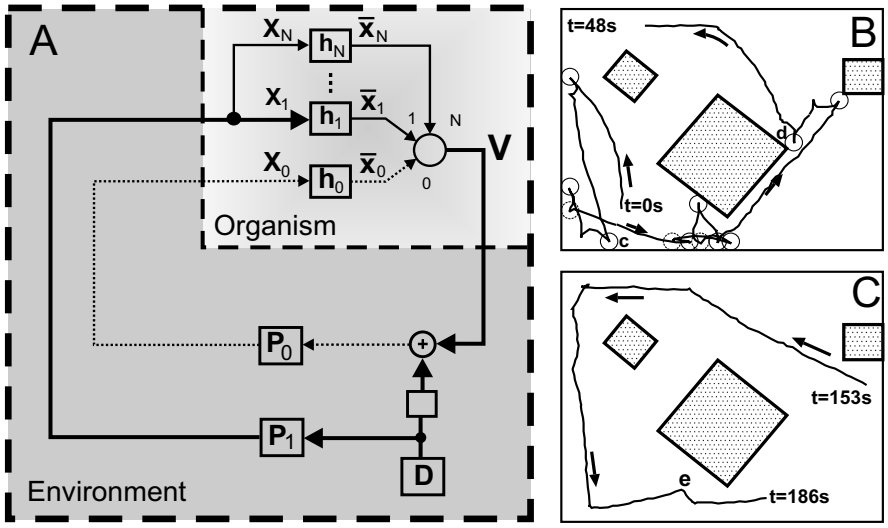

Figure 3: Applying ISO-learning in a control task. A) This architecture is reminiscent of an Actor-Critic architecture (see Fig. 1 C), but here the system does not use evaluative feedback ("rewards") from the environment. Instead it relies only on correlations between the inputs. Hence, it is assumed that the "organism" receives temporally correlated inputs, where x_1 arrives earlier (e.g.; a signal from a range finder reflected from an obstacle) and x_0 arrives later (e.g.; a signal from a touch sensor triggered at the moment of touching the obstacle). P_0, P_1 denote environmental transfer functions, D a signal ("disturbance"), which arrives at the inputs: undelayed at x_1 and with delay τ at x_0. Other symbols are as in Fig. 2, here we have also implemented a filter bank of 10 filters with different frequencies all driven by the same input x_1, this way creating also something like a serial compound representation (Sutton and Barto, 1990). This, however, was done purely to speed up learning and to create smoother output signals. Note that a filter bank approach will still lead to weight stabilization (Fig. 2 D). After successful learning, the output V will fully compensate the disturbance D at the summation node of the inner loop leading to $x_0 = 0$, which is equivalent to a functional elimination of the inner loop. The system has learned the inverse controller of the inner loop (Porr et al., 2003). B,C) Trajectory of a real robot early (B) and late (C) during learning in an area with three obstacles (boxes). Collisions are denoted by the small circles (forward=solid, backward=dashed). Only forward collisions can be used for learning. In such an environment the robot never needed more than 10 forward collisions to learn the task. This way it is as fast as the best RL-algorithms which require sophisticated credit-structuring and temporal assignment mechanisms. (Touzet 1999, also Touzet pers. communication).

necessary earlier signal because they respond before the touch sensor is triggered.

thus two sets of neurons. The correct cross-wiring is described in Porr and Wörgötter (2003a). Here of importance is that ISO-control essentially works without any signal post-processing or conditioning.

ISO-learning learns this correlation. After learning the output is $v = \sum \omega_k \bar{x}_k$, where $k \geq 1$, because the touch sensors (x_0) are not anymore triggered. This signal will now in same way as before, but earlier, lead to a retraction reaction and the primary reflex will be avoided.

Interestingly, the principle of disturbance minimization by reflex avoidance can be employed in the same way to learn a food-retrieval task (see Porr and Wörgötter 2003b). Here a behavior emerges which looks like reward-retrieval (or return-maximization) but which really follows the disturbance-minimization principle.

5 Conclusion

The goal of this article was to point out that realistic autonomous animal control cannot easily be achieved with reward-based learning mechanisms. At first, rewards must not be externally defined, because in this case the controlled system will operate according to the intentions of its designer and it will not follow its own ones, thus, failing to be autonomous. Not even externally defined reference frames (evaluative feedback) are allowed, because in this case the same problem occurs only at a higher level of abstraction. After all, how can the designer make sure that he/she has taken care of everything which is relevant for the system (frame-problem, Dennett (1984)). This cannot be achieved as long as the designer is unable to "see the world with the system's eyes", which, however, is never the case even between two humans. Thus, ultimately only reward-free learning mechanisms (e.g. correlation based learning), which rely on non-evaluative feedback from the environment will assure autonomous development. Only in a second stage, the system may find a way to make *its own* rewards explicit, possibly through semantic associations between correlated signals. For example, the system could develop a mechanism by which it associated the objective feature "Early" with the semantic evaluation "Good" (and "Late" with "Bad"). Autonomous agents, however, will have to do this on their own and only afterwards reward-based learning becomes feasible again.

6 Acknowledgements

This work was supported by the SHEFC INCITE and the EU ECOVISON grants. The authors wish to thank P. Dayan, A. Saudargiene and L. Smith for helpful

discussions.

References

Baird, L. C. (1995). Residual algorithms: Reinforcement learning with function approximation. In *Proc. of the Twelfth Int. Conf. on Machine Learning*, pages 30–37, San Francisco, CA. Morgan Kaufmanns.

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

Berns, G. S. and Sejnowski, T. J. (1998). A computational model of how the basal ganglia produce sequences. *J. Cogn. Neurosci.*, 10(1):108–121.

Bertsekas, D. and Tsitsiklis, J. (1996). *Neuro-dynamic Programming*. Athena Scientific, Belmont, MA.

Bertsekas, D. P. (1987). *Dynamic programming:Deterministic and stochastic models*. Prentice-Hall, Englewood Cliffs, NJ.

Brown, J., Bullock, D., and Grossberg, S. (1999). How the basal ganglia use parallel excitatory and inhibitory learning pathways to selectively respond to unexpected rewarding cues. *J. Neurosci.*, 19(23):10502–10511.

Contreras-Vidal, J. L. and Schultz, W. (1999). A predictive reinforcement model of dopamine neurons for learning approach behavior. *J. Comput. Neurosci.*, 6:191–214.

Dayan, P. (1992). The convergence of TD(λ). *Mach. Learn.*, 8(3/4):341–362.

Dayan, P. and Seynowski, T. (1994). TD(λ) converges with probability 1. *Mach. Learn.*, 14:295–301.

Dearden, R., Friedman, N., and Russell, S. (1998). Bayesian Q-learning. In *Proceedings of AAAI-98*, Madison, WI.

Dennett, D. C. (1984). Cognitive wheels: The frame problem of AI. In Hookway, C., editor, *Minds, machines and evolution*, pages 129–151. Cambridge University Press.

Gaskett, C., Wettergreen, D., and Zelinsky, A. (1999). Q-Learning in continuous state and action spaces. In *Proceedings of the 12th Australian Joint conference on artificial Intelligence*, Sydney, Australia. Springer-Verlag.

Gordon, G. J. (1995). Stable function approximation in dynamic programming. In *Proc. of the Twelfth Int. Conf. on Machine Learning*, pages 261–268, San Francisco, CA. Morgan Kaufmanns.

Grossberg, S. (1995). A spectral network model of pitch perception. *J. Acoust. Soc. Am.*, 98(2):862–879.

Grossberg, S. and Merrill, J. (1996). The hippocampus and cerebellum in adaptively timed learning, recognition and movement. *J. Cogn. Neurosci.*, 8:257–277.

Grossberg, S. and Schmajuk, N. (1989). Neural dynamics of adaptive timing and temporal discrimination during associative learning. *Neural Networks*, 2:79–102.

Houk, J. C., Adams, J. L., and Barto, A. G. (1995). A model of how the basal ganglia generate and use neural signals that predict reinforcement. In Houk, J. C., Davis, J. L., and Beiser, D. G., editors, *Models of information processing in the basal ganglia*, pages 249–270. MIT Press, Cambridge, MA.

Joel, D., Niv, Y., and Ruppin, E. (2002). Actor-critic models of the basal ganglia: new anatomical and computational perspectives. *Neural Networks*, 15:535–547.

Kaelbling, L. P. (1990). *Learning in embedded systems*. PhD thesis, Department of Computer Science, Stanford University, Stanford University.

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

Kearns, M., Mansour, Y., and Ng, A. Y. (2002). A sparse sampling algorithm for near-optimal planning in large markov decision processes. *Mach. Learn.*, 49:193–208.

Kearns, M. and Singh, S. (2002). Near-optimal reinforcement learning in polynomial time. *Mach. Learn.*, 49:209–232.

Klopf, A. H. (1986). A drive-reinforcement model of single neuron function. In Denker, J. S., editor, *Neural networks for computing: AIP Conf. Proc.* , volume 151. New York: American Institute of Physics.

Klopf, A. H. (1988). A neuronal model of classical conditioning. *Psychobiol.*, 16(2):85–123.

Mahadevan, S. and Connell, J. (1992). Automatic programming of behavior-based robots using reinforcement learning. *Artificial Intelligence*, 55:311–365.

Meuleau, N. and Bourgine, P. (1999). Exploration of multi-state environments: Local measures and back-propagation of uncertainty. *Mach. Learn.*, 35(2):117–154.

Millan, J. R. (1996). Rapid, safe, and incremental learning of navigation strategies. *IEEE Transactions on Systems, Man and Cybernetics*, 26(3):408–420.

Porr, B., von Ferber, C., and Wörgötter, F. (2003). ISO-learning approximates a solution to the inverse-controller problem in an unsupervised behavioral paradigm. *Neural Comp.*, 15:865–884.

Porr, B. and Wörgötter, F. (2002). Isotropic sequence order learning using a novel linear algorithm in a closed loop behavioural system. *Biosystems*, 67(1–3):195–202.

Porr, B. and Wörgötter, F. (2003a). Isotropic sequence order learning. *Neural Comp.*, 15:831–864.

Porr, B. and Wörgötter, F. (2003b). Isotropic sequence order learning in a closed loop behavioural system. *Proc. Roy. Soc. B*, in press.

Russell, S. and Norvig, P. (1995). *Artificial intelligence: A modern approach*. Prentice Hall, London, UK.

Santos, J. M. and Touzet, C. (1999a). Dynamic update of the reinforcement function during learning. *Connection Science, spec. issue on adaptive robots*, 11(3/4).

Santos, J. M. and Touzet, C. (1999b). Exploration tuned reinforcement function. *Neurocomputing,special issue on NEURAP98*, 28(1-3):93–105.

Smart, W. D. and Kaelbling, L. P. (2000). Practical reinforcement learning in continuous spaces. In *Proc. of the Seventeenth Int. Conf. on Machine Learning*, San Francisco. Morgan Kaufmann.

Suri, R. E., Bargas, J., and Arbib, M. A. (2001). Modeling functions of striatal dopamine modulation in learning and planning. *Neurosci.*, 103(1):65–85.

Suri, R. E. and Schultz, W. (1998). Learning of sequential movements by neural network model with dopamine-like reinforcement signal. *Exp. Brain Res.*, 121:350–354.

Sutton, R. and Barto, A. (1981). Towards a modern theory of adaptive networks: Expectation and prediction. *Psychol. Review*, 88:135–170.

Sutton, R. S. (1984). *Temporal credit assignment in reinforcement learning*. PhD thesis, University of Massachusetts, Amherst.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Mach. Learn.*, 3:9–44.

Sutton, R. S. and Barto, A. G. (1990). Time-derivative models of Pavlovian reinforcement. In Gabriel, M. and Moore, J., editors, *Learning and computational neuroscience:Foundation of adaptive networks*. MIT Press.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Bradford Books, MIT Press, Cambridge, MA, 2002 edition.

Thrun, S. B. (1992). The role of exploration in learning control. In White, D. A. and Sofge, D. A., editors, *Handbook of intelligent control: Neural, fuzzy and adaptive approaches*. van Nostrand Reinhold, Florence, KY.

Touzet, C. (1999). Neural networks and Q-learning for robotics. http://www.sciences-cognitives.org/scico/annuaire/Touzet_Claude/Touzet_IJCNN_Tut.pdf. Tutorial at IJCNN'99, Washington, DC, USA.

Tsitsiklis, J. N. and van Roy, B. (1996). Feature-based methods for large scale dynamic programming. *Mach. Learn.*, 22:59–94.

Wiering, M. (1999). *Explorations in efficient reinforcement learning*. PhD thesis, Universiteit van Amsterdam, Amsterdam, The Netherlands.

Witten, I. H. (1977). An adaptive optimal controller for discrete-time Markov environments. *Information and Control*, 34:86–295.

Wörgötter, F. and Porr, B. (2004). Temporal sequence learning for prediction and control - a review. *Neural Comp.*, 000:000–000. submitted.

Wyatt, J. (1996). *Exploration and inference in learning from reinforcement*. PhD thesis, Department of Artificial Intelligence, University of Edinburgh, Edinburgh, UK.

Wyatt, J. (2001). Exploration control in reinforcement learning using optimistic model selection. In *Proc. of the Eighteenth Int. Conf. on Machine Learning*, pages 593–600, Williamstown, MA.