Anticipating Rewards in Continuous Time and Space without Discretization

Arnaud J. Blanchard and Lola Cañamero

Adaptive System Research Group
School of Computer Science
University of Hertfordshire
College Lane, Hatfield, Herts AL10 9AB, UK
{A.J.Blanchard, L.Canamero}@herts.ac.uk

Abstract. We propose here, a new approach of reinforcement learning which does not need discretization, notion of events, or classification. Instead of learning rewards for different possible actions of an agent in all the situations, we propose to only learn the main situations to avoid and the main situations to reach. After describing the algorithm, we present the results of an implementation on a real robot learning which are the sensations it should avoid or reach. We finally conclude with the promises and limitations of this approach.

1 Introduction

Reinforcement learning aims to make an agent learn which actions it should perform in order to maximize the obtention of rewards. This learning system is interesting as it allows to easily "program" agents to make them do whatever we want simply by emitting different signals (reinforcement) according to the relevance of its actions. Moreover, animals and humans are very efficient in this task and it is interesting to know how. That is for example the way dogs can be trained, or that a rat learns to move in a maze in order to find a source of food. They can anticipate the reward associated with their actions. However good models for this kind of behavior are complex to design, the main difficulty is to identify the cues predicting rewards.

In section 1.1 we present the principle of classical approaches of reinforcement learning and in section 1.2 we discuss the problems raised by these approaches, mainly that the environment has to be arbitrarily discretized and that they need a lot of computational resources. In section 2 we propose a new architecture able to handle these problems and we present the results of the experiments on a real robot section 3.

1.1 Classical reinforcement learning

The temporal-difference model (TD learning [1]) is a very common and efficient method for reinforcement learning, the principle is to discretize the inputs (from

the sensors and the internal states) in order to obtain a finite number of possible states (inputs). The expected reinforcement for each state is evaluated using the actual reinforcement of the state more the reinforcement expected in states immediately accessible. Then the agent acts in order to reach the states maximizing the expected reinforcement. Even if the convergence of the algorithm is proved, the learning is very slow because the agent needs to try each state several times, and it strongly depends on the discretization used, which can lead to a huge number of different states. It is therefore also very demanding in memory to store all the expected reinforcement for each possible state.

The Q-learning [2] uses similar principles but it works even if the agent does not know which action to execute in order to reach a given state. The agent learns the expected reinforcement for each possible couple state and action. This increases again the time of learning because there are many more possibilities to explore, the quantity of memory needed is multiplied by the number of different possible actions.

1.2 The problem of discretization

In artificial intelligence, numerous of powerful algorithms have been design to learn, anticipate and decide. However they are often inappropriate when applied on robots in real world without being prepared to detect specific stimuli. For example many models of classical or instrumental conditioning need to predefine the set of possible stimuli to consider. Information theory [3] provides powerful tools to statistically measure the temporal correlation between events and anticipate them. However, the problem is here again to define the set of events through the discretization.

Discretization can be adaptive, by grouping together events which carry the same predictive information. For this we can use classification algorithms like the k-mean, Kohonen's maps, Estimation-Maximization algorithm (see [4] for more). Many of these algorithms need strong assumptions on the distribution of classes and the discretization needs to be arbitrarily or randomly initiated whereas the quality of the learning process depends on this initialization.

When developing the Q-learning algorithm, Watkins was aware of the difficulty to cope with continuity: "To avoid the complications of systems which have continuous state-spaces, continuous action sets, or which operate in continuous time, I will consider only finite, discrete-time Markov decision processes" [2] page 38. Even once the discretization is done, the algorithm converges quite slowly because it needs to try several times the different possible states in order to statistically estimate the reinforcement that can be expected for each one. Once the reinforcement can be reliably anticipated for each state, the agent can act in order to reach the state with the highest expected reinforcement. These approaches are very powerful when they are used in simulation as the environment is often already discretized (e.g. a grid where the agent is moving) and because it is easy to make an agent try different situations a huge number of times. They are very well adapted in robotics when the elements of the envi-

ronment are predefined, when there are obvious salient cues that the robot can consider as classes of events (e.g. salient color or pattern).

In the case of robots in real environments without specific features, the robots have to find by themselves the cues predicting rewards. These cues are not necessarily salient, it can be a specific light's intensity, a range of sound's frequencies or a specific position and not, as it is commonly used, a binary signal as the presence or absence of light, sound, shape, etc. Humans and animals are very efficient to discriminate stimuli a priori similar if they have distinctive predictive values. In this case, using the salience of sensations can be misleading as for example, a light passing on or off does not have any predictive value whereas a small change in intensity of a light at a specific level can be significant. Most algorithms involving discretization are not be able to cope efficiently with this kind of situation because they waste a lot of memory storing the predictions of expected rewards for many different values of the sensory input whereas most of them are not relevant or redundant. Moreover, there is usually no difference between the effect of a small reward obtained immediately and the promise of an important reward later. However in some cases, it is very important to make the difference; if a robot is about to "die" it should go where it is sure to quickly find at least a small reward, whereas it should try to maximize the long term reward when it has more time.

2 Our continuous approach of reinforcement learning

As there is no "free lunch" [5]—in average, we cannot have a better algorithm than another one with the same assumptions—, we need to do assumptions about the world; we assume that the world is continuous: there are continuous variations of rewards with continuous variations of the sensory inputs and the relations between rewards and sensory inputs are consistent. Consequently, if the agent receives a high reward for a specific sensory input (sensation), it can anticipate a good reward for other close sensations. Therefore, instead of estimating the expected reward for all the many possible states and trying to reach the state anticipating the maximum reward, we propose to make the agent only memorize the sensation associated with the best reward called *desired sensation* (see Figure 2).

To illustrate the possibilities, we consider a continuous space (typically, the environment of a robot in a real world) and we use sets of real variables: $S = \{s_1, s_2, \ldots\}$ for sensory input (light intensity, pressure, distance to obstacles, etc.), $A = \{a_1, a_2, \ldots\}$ for actions (velocity, rotation angle, etc.) and r a real variable to represent the immediate reward. To simplify, we focus on one dimension of sensory input ($S = \{s\}$) and we consider the problem presented Figure 1 where a robot moving forward and backward, having the distance to a landmark as sensory input (S) must be able to anticipate the presence of a reward (r) on its side.

In order to make the robot learn the sensation associated with the highest reward, we can simply set the desired sensation ($\hat{S}$) as equal to the current

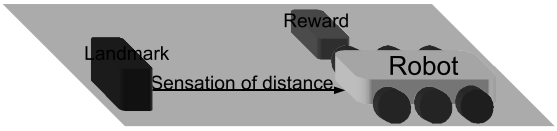

Fig. 1. Using its distance sensor, the robot must be able to anticipate the presence of the reward on its side.

sensation (S) only when the reward (r) is higher than the highest known reward ($\hat{r}$).

$$\text{if } r > \hat{r} \text{ then } \begin{cases} \hat{r} = r \\ \hat{S} = S \end{cases} \tag{1}$$

The problem is that if the reward is very high by chance and is never high again, or if the sensation is very hard to obtain, the desired sensation learned will be useless. Moreover, the agent is not able to learn more than one sensation associated with a reward. Actually, even if it memorizes another desired sensation associated with a slightly smaller reward, the principle of continuity make this desired sensation to be infinitely close to the previous one learned as we can see in Figure 3.

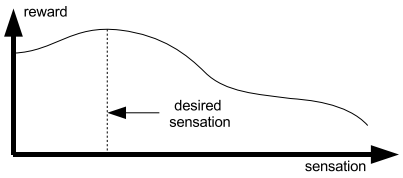

Fig. 2. Desired sensation depending on the reward associated to the sensation

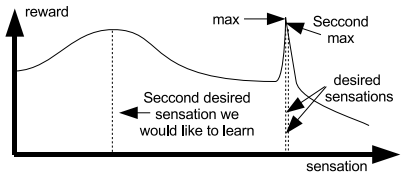

Fig. 3. Impossibility to learn local maximum.

Therefore to be reliable and robust, the agent should not only memorize the sensations associated with the highest reward but the sensations associated with a positive reward at a high probability. We shown in (1) how to memorize the sensation associated with the maximum reward; we present in (2) how the agent can compute the most probable sensation ($\overline{S}$), as the average of all sensations at each time (t).

$$\overline{S}_t = \frac{S_0 + \ldots + S_t}{t+1} \tag{2}$$

To implement this, the agent needs to store all the sensations at all the time which is virtually impossible and moreover it is not biologically plausible. However, we show how it can be equivalent to use an incremental rule (3) similar to

the learning rule of Rescorla and Wagner [6] used for conditioning.

$$\begin{aligned}
\overline{S}_t &= \frac{S_0 + \ldots + S_{t-1} + S_t}{t+1} \\
&= \frac{\frac{S_0+\ldots+S_{t-1}}{t} \times t + S_t}{t+1} \\
&= \frac{\overline{S}_{t-1} \times t + S_t}{t+1} \\
&= \frac{\overline{S}_{t-1} \times (t+1) - \overline{S}_{t-1} + S_t}{t+1} \\
&= \overline{S}_{t-1} + \frac{1}{t+1}\left(S_t - \overline{S}_{t-1}\right) \\
&= \overline{S}_{t-1} + \eta_t . \left(S_t - \overline{S}_{t-1}\right) \qquad (3)
\end{aligned}$$

The learning rate $\eta_t = \frac{1}{\widetilde{T}_t}$ and in this case we only need a variable increasing with the time ($\widetilde{T}_t = \widetilde{T}_{t-1} + 1; \widetilde{T}_0 = 1$) and a variable memorizing the current average sensation ($\overline{S}$). The complexity of the calculus is very low and biologically plausible.

Now the agent can learn two extreme cases: the sensation associated to the best reward ($\hat{S}$), and the average sensation ($\overline{S}$) whatever the reward is. None of these sensations are very useful to be learned because, the first one indicates the sensation associated to the best reward but may not be reliable as it may have happened only once and the second one indicates which are the sensations happening more often but it does not mean it is a good thing. However, all the intermediate cases are very important because in order to maximize the cumulative reward the agent should balance the effect of the reward and the effect of the probability. If an agent urgently needs a reward (for example a resource to avoid to die), it will focus on the sensations promising small rewards with high probabilities (easy to obtain) but if it is not urgent, it will focus on sensations promising higher rewards in order to maximize the cumulative reward and also to get more experiences about these high rewards.

The agent has to be able to memorize a range of desired sensations, from the ones often obtained but predicting small rewards to the ones rarely obtained but predicting high reward. In [7] we shown how an agent can learn the average "best" sensation by weighting each sensation with the associated reward [1] simply by modifying the function of the learning rate η_t with $\eta_t = \frac{r_t}{\widetilde{r}_t}$ with $\widetilde{r}_t = \widetilde{r_{t-1}} + r_t; \widetilde{r}_0 = r_0$. However the agent was not able to balance the importance of the reward with the importance of its probability. Moreover, past experiences with highly positive and negative rewards would have the same consequences than past experiences with an average constant reward.

We propose in (4) a solution to learn different desired sensations ($\overline{S^k}$) where the balance between the importance of the reward and its probability is con-

[1] the desired sensations were called desired perceptions and the comfort corresponded to the reward.

trolled by the parameter k.

$$\begin{aligned}\overline{S_t^k} &= \frac{e^{k.r_0}.S_0 + \ldots + e^{k.r_t}.S_t}{e^{k.r_0} + \ldots + e^{k.r_t}} \\ &= \overline{S_{t-1}^k} + \frac{e^{k.r_t}}{e^{k.r_0} + \ldots + e^{k.r_t}}.\left(S_t - \overline{S_{t-1}^k}\right)\end{aligned} \tag{4}$$

For extreme values of k, 0 and $+\infty$, we obtain respectively the same result as in (2) because $e^0 = 1$ and as in (1) because:

$$\lim_{k \to +\infty} \frac{e^{k.r_0}.S_0 + \ldots + e^{k.r_t}.S_t}{e^{k.r_0} + \ldots + e^{k.r_t}} = S_{argmax(r_0,\ldots,r_t)}$$

Another advantage is that only the variation of the reward has an influence, not its absolute value; we do not need to define a priori which value of reward has to be considered as a good reward. Actually, we can add any constant (c) to the reward and it does not change the learning rate:

$$\begin{aligned}\eta_t^k &= \frac{e^{k.(r_t+c)}}{e^{k.r_0+k.c} + \ldots + e^{k.r_t+k.c}} \\ &= \frac{e^{k.r_t}.e^{k.c}}{e^{k.r_0}.e^{k.c} + \ldots + e^{k.r_t}.e^{k.c}} \\ &= \frac{e^{k.r_t}}{e^{k.r_0} + \ldots + e^{k.r_t}} \\ &= \frac{e^{k.r_t}}{\widetilde{r_t^k}}\end{aligned} \tag{5}$$

with $\widetilde{r_t^k} = \widetilde{r_t^k} + e^{k.r_{t-1}}$. We shown how an agent can learn sensations predicting rewards, but it can also be useful to learn sensations predicting danger or negative reward in order to avoid them. With this model, they are easy to compute as they are equal to the sensations S_t^k for negative values of the parameter k. They are called *avoided sensations*.

The problem of computing the desired sensations is that they can be in between two local maximums and therefore predict a reward where there is no reward, see Figure 4. The solution is to make the agent partly forget the past and consequently have its desired sensations moving from local maximums to local maximums but not staying in between. In [7] we raised the learning rate η_{t^k} to the power of γ, with γ between 0 and 1, in order to make the agent continuously learn and partly forgot the effect of the old experiences.

$$\overline{S_t^{k,\gamma}} = \overline{S_{t-1}^{k,\gamma}} + \left(\frac{e^{k.r_t}}{\widetilde{r_t^k}}\right)^{\gamma}\left(S_t - \overline{S_{t-1}^{k,\gamma}}\right) \tag{6}$$

Smaller is γ, higher is the learning rate and faster the desired sensation changes, therefore the desired sensation oscillates between local maximums depending on the exploration of the agent as depicted in Figure 5.

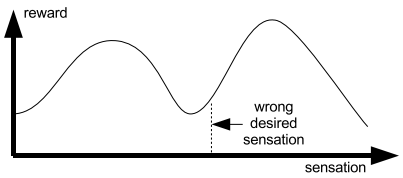

Fig. 4. Wrong desired sensation, average of multiple local maximums

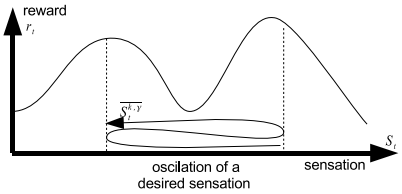

Fig. 5. Oscillation of a desired sensation local maximums ($\gamma < 1$)

The problem of partly forgetting the past is that the agent will not be able to remember a sensation associated with a good reward if it did not experience it recently. However, the desired sensations oscillate from local maximums to local maximums—and avoided sensations oscillates from local minimums to local minimums—therefore, if the agent memorizes the extreme values ($\widehat{S}$) of the successive desired and avoided sensations (see Figure 6 for desired sensations), it can remembers two—the two extremes—sensations anticipating a positive reward and two sensations anticipating a negative reward (punishment). moving forward and backward, having the distance to a landmark as sensory input (S) must be able to anticipate the presence of a reward (r) on its side. In order to

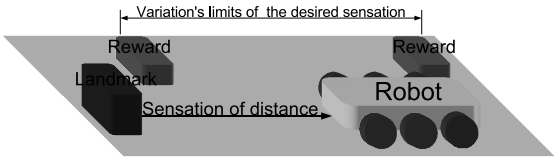

Fig. 6. The desired sensation strictly oscillates between the two rewards.

remember these extreme values, we use a similar equation than in (4) but this time the agent memorizes the desired sensations ($S^{k,\gamma}$) with extreme values of themselves instead of memorizing the sensations associated with extreme rewards ! The weight in the exponential function is therefore the desired sensation itself multiplied by another parameter l defining if the agent memorizes the minimum value of the desired sensation ($l < 0$) or the maximum value ($l > 0$). We can see in (7) how this extreme values are defined with $\widetilde{S_t^{k,\gamma,l}} = e^{l.S_0^{k,\gamma}} + \ldots + e^{l.S_t^{k,\gamma}}$.

$$\widehat{S_t^{k,\gamma,l}} = \widehat{S_{t-1}^{k,\gamma,l}} + \frac{e^{l.\overline{S_t^{k,\gamma}}}}{\widetilde{S_t^{k,\gamma,l}}}.\left(\overline{S_t^{k,\gamma}} - \widehat{S_{t-1}^{k,\gamma,l}}\right) \tag{7}$$

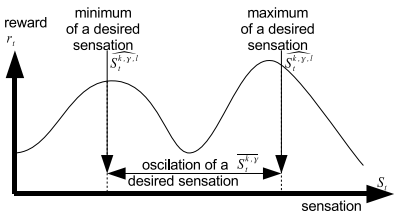

Fig. 7. Extreme values of a desired sensation ($k > 0$). The left extremum is for $l < 0$ and the right extremum is for $l > 0$

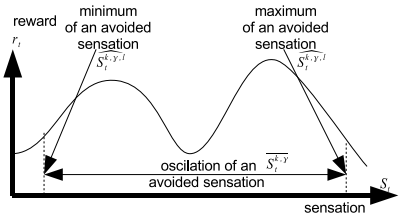

Fig. 8. Extreme values of an avoided sensation ($k < 0$). The left extremum is for $l < 0$ and the right extremum is for $l > 0$

3 Experiments

We test this algorithm, on a real robot (Koala [8]) and want it to memorize sensations associated with reward or punishment. The robot is moving alternatively forward and backward at the front of a box used as a landmark. The sensory input (S) we are using is its frontal distance sensor measuring its distance to the front box. The right distance sensor is used to detect rewards (r), a box on its right represents a positive reward r. We present in Figure 9 the reward obtain by the robot depending on its sensation of distance to the landmark. We observe,

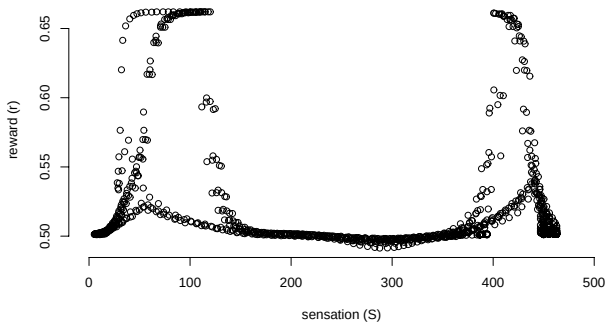

Fig. 9. Value of the reward (r) depending on the sensation (S). We see that the maximum of reward is for a sensations of about 75 and 425 (the units does not matter) which correspond to the presence of a box on the right of the robot.

how the desired sensations of the robot evolve with the time and the experiences of the robot. We compute the desired sensation ($\overline{S^{k,\gamma}}$ with $k = +400; \gamma = 0.9$) and the avoided sensation ($\overline{S^{k,\gamma}}$ with $k = -400; \gamma = 0.9$). If k or γ differ, the

curves are more or less smooth but qualitatively similar. We present the results in Figure 10. The desired sensation oscillate between the sensations 75 and 425 which correspond to the presence of the reward (boxes). The avoided sensation oscillates between the boxes at the beginning and then around them notifying that the robot should avoid to be between the boxes or behind them.

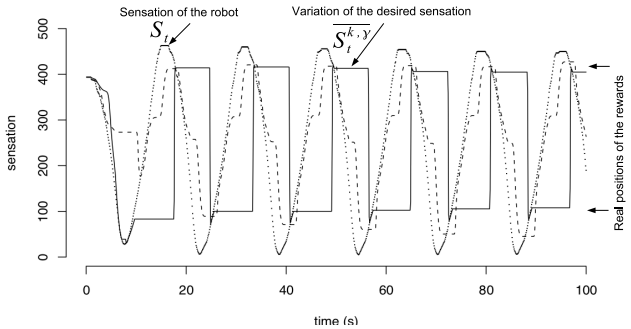

Fig. 10. Evolution of the sensation (S_t) of the robot in dotted line, the desired sensation ($\overline{S^{k,\gamma}}$ with $k = +400; \gamma = 0.9$) in solid line and the avoided sensation in dashed line ($\overline{S^{k,\gamma}}$ with $k = -400; \gamma = 0.9$). The desired sensation oscillate between the sensations 75 and 425 which correspond to the presence of the reward. The avoided sensation oscillates between the boxes at the beginning and then around them notifying that the robot should avoid to be between the boxes or around them.

The desired and avoided sensations are moving all the time therefore, the robot cannot remember anything for a long time. However, the next step for the robot is to memorize the extremums of these desired and avoided sensations. We present in Figure 11 the evolution of these extremums ($\widehat{S^{k,\gamma,l}}$) for the same values of k and γ and -0.1 and 0.1 for l—l is small because the amplitude of the sensation is high but anyway it does not have a strong effect on the qualitative result. The extremums of the avoided sensations quickly converge (almost at the first cycle) to the sensations corresponding to the boxes on the side (the reward 75 and 425). The extremums of the avoided sensations correspond in a first time to the sensation between the boxes and at the end to the sensations behind the boxes which mean the robot should avoid to stay between the boxes (no reward) or behind them (no reward either).

4 Conclusion and perspectives

We have presented the first basic principles and implementation of a new approach of reinforcement learning where the agents can learn to anticipate reward using their sensory inputs. In [9], Doya proposes to approximate the reward function in order to process reinforcement learning in continuous time and space but we argue that it is enough to only memorize where are the rewards even if the

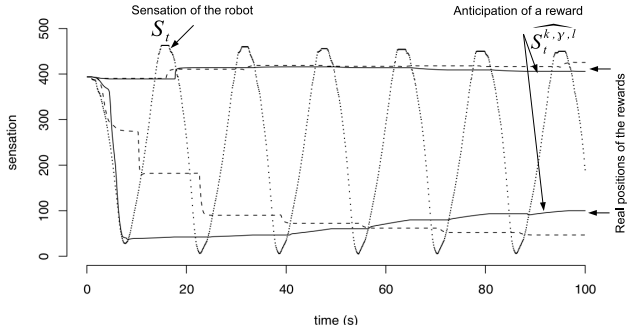

Fig. 11. Evolution of the extreme values ($S^{k,\gamma,l}$) of the desired and avoided sensations with the same parameters as previously for k and γ but l worth 0.1 for the curves on the top, and -0.1 for the curves on the bottom. The extremums of the desired sensations are in solid line and the extremums of the avoided sensations are in dashed line.

robots cannot know what are these rewards. The advantages are that it memorizes only the relevant information and does not need much memory or computer time. It does not use notion of events or discretization which strongly reduces the effects of choices a priori and decreases the learning time. Actually agents can learn with only one presentation of the reward which is very useful in robotics where exploration is "expensive".

Even if the algorithm does not need many a priories on the world, it has a couple of parameters to set. k to balance the importance of the reward's value versus its probability, γ and l to vary the average speed of learning. However, these parameters only have quantitative effects and we have already proposed in [7] ways to modulate these kinds of parameters and there are others ([10], [11]). An agent will also need to decide whether it should explore or exploit its environment in order to use what it learns efficiently an we could adapt several strategies like: [12], [13] or [14].

We shown how a robot can predict the presence of only two rewards, however we can extend it to many more rewards looking for the two extreme desired sensations in between two extreme avoided sensations and so on (see Figure 12). We

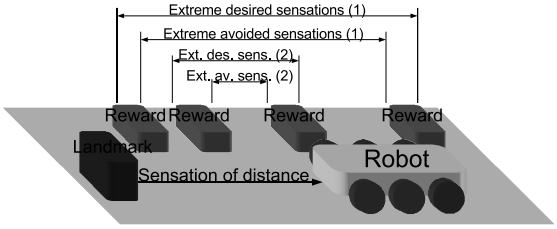

Fig. 12. Using successive detections of desired and avoided rewards, robots can anticipate as many rewards as we want.

are also currently working on expending this algorithm to many more dimensions and it seems promising.

Acknowledgments

Arnaud Blanchard is funded by a research studentship of the University of Hertfordshire. This research is partly supported by the EU Network of Excellence HUMAINE (FP6-IST-2002-507422).

References

1. Sutton, R., Barto, A.: A temporal-difference model of classical conditioning. In: Proceedings of the Ninth Annual Conference of the Cognitive Science Society. (1987) 355–378
2. Watkins, C.: Learning from Delayed Rewards. PhD thesis, King's College (1989)
3. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley-Interscience (1991)
4. Butz, M.V., Sigaud, O., Gérard, P.: Internal models and anticipations in adaptive learning systems. In Butz, M.V., Sigaud, O., Gérard, P., eds.: LNCS 2684 : Anticipatory Behavior in Adaptive Learning Systems. Springer-Verlag (2003)
5. Wolpert, D., Macready, W.: No free lunch theorems for optimisation. In: IEEE Trans. on Evolutionary Computation. Volume 1. (1997) 67–82
6. Rescorla, R., Wagner, A.: A theory of pavlovian conditioning: Variations in effectiveness of reinforcement and nonreinforcement. In Black, A., Prokasy, W., eds.: Classical Conditioning II, New York: Appleton-Century-Crofts (1972) 64–99
7. Blanchard, A., Cañamero, L.: From imprinting to adaptation: Building a history of affective interaction. Proc. of the 5th Intl. Wksp. on Epigenetic Robotics (2005) 23–30
8. K-Team. http://k-team.com/robots/koala (2002)
9. Doya, K.: Reinforcement learning in continuous time and space. Neural Computation **12**(1) (2000) 219–245
10. Arkin, R.: Behavior-Based Robotics. The MIT Press (1998)
11. Avila-Garcia, O., Cañamero, L.: Using hormonal feedback to modulate action selection in a competitive scenario. In Schaal, S., Ijspeert, J., Billard, A., Vijayakumar, S., Hallam, J., Meyer, J.A., eds.: From Animals to Animats 8: Proceedings of the 8th International Conference on Simulation of Adaptive Behavior, Cambridge, MA: The MIT Press. (2004) 243–252
12. Steels, L.: The autotelic principle. In Fumiya, I., Pfeifer, R., Steels, L., Kunyoshi, K., eds.: Embodied Artificial Intelligence. Volume 3139 of Lecture Notes in AI. Springer Verlag, Berlin (2004) 231–242
13. Oudeyer, P.Y., Kaplan, F.: Intelligent adaptive curiosity: a source of self-development. In Berthouze, L., Kozima, H., Prince, C.G., Sandini, G., Stojanov, G., Metta, G., Balkenius, C., eds.: Proc. of the 4th Intl. Wks. on Epigenetic Robotics. Volume 117., Lund University Cognitive Studies (2004) 127–130
14. Blanchard, A., Cañamero, L.: Modulation of exploratory behavior for adaptation to the context. In Kovacs, T., J., M., eds.: Biologically Inspired Robotics (Bironet) in AISB'06: Adaptation in Artificial and Biological Systems. Volume II. (2006) 131–139