Maximizing learning progress: an internal reward system for development

Frédéric Kaplan and Pierre-Yves Oudeyer

Developmental Robotics Group
Sony Computer Science Laboratory Paris
6 rue Amyot, 75005 Paris, France
kaplan@csl.sony.fr, py@csl.sony.fr

Abstract. This chapter presents a generic internal reward system that drives an agent to increase the complexity of its behavior. This reward system does not reinforce a predefined task. Its purpose is to drive the agent to progress in learning given its embodiment and the environment in which it is placed. The dynamics created by such a system are studied first in a simple environment and then in the context of active vision.

1 Introduction

Models of natural or artificial autonomous agents usually imply the existence of primary motivational principles that govern the behavior of the agent. Biologists typically describe most of the processes of metabolical regulations as the results of homeostatic mechanisms maintaining internal "variables" such as hunger, thirst or temperature into desirable limits. Behaviorists have argued that behavior was driven by the search for a set of potential rewards corresponding to primary and conditioned reinforcers [1]. This kind of model has lead to reinforcement learning architectures in which a robot learns to behave in order to receive artificial "rewards"[2]. In return, recent results seem to suggest a convergence between the reinforcement learning theory and neurophysiological data from the midbrain dopamine system in particular [3–5].

Rewards have traditionally been viewed as external stimulations provided by the environment or by an experimenter. This chapter discusses the opportunity of considering the existence of an *internal* reward system that would act as a driving force for learning: a kind of "epistemic hunger". Several authors have argued that human behavior is probably driven by a principle of this sort (e.g. Korand Lorenz's "neophily" [6], Csikszentmihalyi's "flow experiences" [7]). But very few precise models exist to explain the mechanisms underlying such a curiosity principle.

In our vision, curiosity is tightly linked with prediction capabilities and learning experiences [8]. "Learning situations" occur when an agent encounters a situation which is not yet entirely predictable based on its current capabilities but learnable using its algorithms for adaptation. In models which view homeostasis as a major drive for living creatures, such kind of situations may be seen as

perturbations [9]. If the goal of the agent is to constantly try to minimize its error in prediction, learning is simply a way to reach again a form of equilibrium state (e.g. [10]). Taking an opposite point of view, several authors have suggested that in order to learn efficiently agents should focus on "novel", "surprising" or "unexpected" situations. This would mean that a "curious" agent should focus on situations for which it does not yet have adequate prediction capabilities (e.g [11] in field of developmental robotics or [12] in the field of "active learning").

The view presented in this chapter is different, but in the line of Schmidhuber's theoretical machine learning work [13]. Curiosity is defined neither as a pressure to minimize errors in prediction, nor as a tendency to focus on the most "surprising" situations, but on the contrary as a drive that pushes the agent to lose interest in both predictable and unpredictable areas, to concentrate on situations that *maximize learning progress*.

The next section presents an engine that can generate behaviors based on this reward principle. As the reward system is generic, this engine can be associated with any input-output device. However the resulting behavior highly depends on the embodiment of the device, that is on the physical structure of the device and on the particular implementation of the prediction systems used by the engine. In order to understand the learning dynamics created by this reward system, this chapter focuses first on a simple embodiment. Experiments with a more complex active vision system are then described.

2 A generic engine

2.1 Technical description

Input, Output, Internal rewards. An agent can be viewed as a plant consisting of an input-output device and an engine controlling it. At any time t, the engine receives a vector $S(t)$ of input signals (either internal or external to the agent) and can send a vector $M(t)$ of control signals corresponding to its actions on the environment or on internal parameters. The set of internal reward received at time t is contained in a vector $R(t)$. The purpose of the engine is to maximize the amount of rewards received in a given time frame (possibly infinite).[1]

The complete situation (sensory-motor and rewards) is summarized in a vector $SMR(t)$. The behavior of the engine consists in determining $M(t)$ based on $S(t)$ and on previous sensory-motor-reward situations $SMR(t-1), SMR(t-2), ...$ Given the constraints provided by its embodiment and the environment in which it is placed, the engine develops in an unsupervised manner.[2]

[1] In this paper we only consider the case of a reward vector of dimension 1, but we describe the engine in its general form which can deal with more than one reward function.

[2] It is to be understood that, in the present paper, when the expression "sensory-motor" is used the word "motor" does not necessarily entail physical motion. The term "motor" refers more generally to any control signal having a potential effect

An important point is that the engine receives inputs and produces control commands without any a priori information about what they "externally" mean.

Predictors. Predictors form the most important part of the engine. They are responsible to anticipate future sensory-motor evolutions and expected rewards. Their function can be implemented as a single predictor Π that tries to predict future situations.

$$\Pi(SMR(t)) \rightarrow SMR(t+1) \tag{1}$$

However, in practice, it is often more efficient to implement this global predictor through three specialized *prediction devices*: Π_m, Π_s, Π_r. The three devices take the current situation $SMR(t)$ as an input and try to predict the future motor situation $M(t+1)$, the future sensory situation $S(t+1)$ and the future state of the reward vector $R(t+1)$, respectively. At each time step, once $SMR(t)$ is defined, the three devices learn in order to increase their prediction accuracy. Each predictor adapts differently depending on its implementation. The prediction devices can be implemented in different manners, for instance:

- A recurrent Elman neural network with a hidden layer / context layer. Because this network is recurrent it can predict its output based on the value of the sensory-motor vectors several time steps before t [14].
- A prototype-based prediction system that learns prototypic transitions and extrapolates the result for unknown regions.[3]
- A system using Hidden Markov Models [15].
- A mixture of experts like the one described in [16] and [17]

The performances of the prediction devices are crucial for the system, but the architecture of the engine does not assume anything about the kind of devices used. The choice of a particular technique and its implementation are considered to be part of the embodiment of the device.

Reward system. At each time step t, the reward system computes the current values of $R(t)$ based on internal computation on the architecture. The system we describe below is based on "maximizing learning progress". At any time t, the system can evaluate the current error for predicting sensory effect of a given

---

either on the environment of the agent or on the agent itself: control of physical actuators, activation of sensory devices, change on internal parameters.

[3] It takes the form of a set of vectors associating a static sensory-motor context $SMR(t-1)$ with the predicted vector ($M(t)$,$S(t)$ or $R(t)$). New prototypes are regularly learned in order to cover most of the sensory-motor-reward space. Predictions are made by combining the results of the k closest prototypes. k is typically taken as size(SMR(t)) +1 . This prediction system is faster and more adaptive than the Elman network, but may prove less efficient for complex sensory-motor-reward trajectories.

command. It is the distance between the predicted sensory vector and its actual values.

$$\Pi_s(SMR(t-1)) \rightarrow S'(t), e(t) = distance(S'(t), S(t)) \tag{2}$$

We define the "learning progress" $p(t)$ as the reduction of the error $e(t)$. In the case on an increase of e(t), progress is zero.

$$p(t) = \begin{cases} e(t-1) - e(t) & : \quad e(t) < e(t-1) \\ 0 & : \quad e(t) \geq e(t-1) \end{cases} \tag{3}$$

In the case when "learning progress" is the only variable to maximize, the vector $R(t)$ is of dimension 1 :

$$R(t) = \{p(t)\} \tag{4}$$

Maximizing learning progress forces the agent to move away from predictable trajectories in order to receive rewards when returning to predicted ones. This is very different from minimizing or maximizing the error $e(t)$. Minimizing the error involves carrying out the actions whose effects are the most easy to predict. This leads to the specialization of the agent into a very small sensory-motor domain, that it will try to master perfectly. With such a reward system, the diversity of the behavior of the agent tends to be rapidly reduced. Maximizing the error involves carrying out the actions whose effects are the most difficult to predict. This can lead to good results in some cases. But in the case where part of the sensory-motor space is very difficult to predict, this strategy is likely to result in destructive learning as the agent will not go back to predictable trajectories. Experiments with these different kind of reward functions are discussed in [8].

Action selection. The action selection module chooses the output commands that are expected to lead to the maximum rewards between t and t+T. Several techniques taken from the reinforcement learning literature can be used to solve these kind of problems[2]. In our system, the process can be separated into four phases:

1. Generation : The module constructs a set of possible commands $\{mi\}$. For some applications this phase can be trivial, but more elaborate computations may be required when dealing with complex actuators. As an example of a simple case: if the current value of an actuator control signal, m_0, is 0.7 then the controller may randomly shift the current value so as to produce candidate values such as 0.55, 0.67, 0.8, 0.75, for m_0.
2. Anticipation : With the help of the predictors, by using the prediction devices in a recurrent manner, the module simulates the possible evolution $\{SMR_{mi}\}$ over T time steps. The module combines the result of both Π_m and Π_s to predict future sensory-motor situations and uses Π_r to predict the evolution of the reward vector $R(t)$.

3. Evaluation : For each evolution $\{SMR_{mi}\}$ an expected reward r_{mi} is computed as the sum of all the future expected rewards.

$$r_{mi}(t) = \sum_{j=t}^{t+T} ||R(j)|| \tag{5}$$

4. Selection : The motor command $\{mi\}$ corresponding to the highest expected reward is chosen.

The action selection module monitors how well the system is behaving by computing the average reward $< ar(t) >$ over K timesteps. If $< ar(t) >$ is below a given threshold η the system acts randomly instead of using the anticipation process. This allows the discovery of opportunities for learning by chance and then to exploit them.

3 The light switching system : a very simple embodiment

In order to better understand how the architecture works, this section describes first the behavior of the system for a very simple embodiment. Let's assume that the device is equipped with two sensors. The first one, $pos(t)$ is its position between 0 and 1. The second $light(t)$ can only take two values 0 and 1 and corresponds to the presence or absence of a light in the environment. This light is switched on when the agent occupies a position between 0.89 and 0.91, otherwise it is zero. The system is equipped with a single actuator *nextpos* corresponding to the next position the system should go to. It tries to maximizes its "learning progress" $p(t)$. A step by step evolution of the system in this simple situation is described in a detailed manner in the appendix.

Π_s, Π_m, Π_r are three prototype-based predictors with a maximum capacity of 500 prototypes. For these simulations $T = 2, K = 1$. To understand the dynamics produced by the engine more easily, we set $\eta = 0$. This means that the system always attempts to use the action selection mechanism to maximize rewards.

Measures. In order to evaluate the behavioral effect of the action selection mechanism based on maximizing learning progress, we need to systematically compare the simulation results with results obtained for an agent that learns choosing random actions. Despite its simplicity, the "random" action strategy can be efficient for learning about unknown environments and discover sensory-motor contingencies.

The first question is whether the system manages to reach its goal : maximizing learning progress. We define the cumulative progress $P(t)$ as the integration over time of $p(t)$.

$$P(t) = \sum_{j=0}^{t} p(t) \tag{6}$$

To evaluate the performance of the action selection mechanism, we define the comparative progress ratio $C_P(t)$ as :

$$C_P(t) = \frac{P_{MAXPROGRESS}(t)}{P_{RANDOM}(t)} \tag{7}$$

In the context of this environment, the second question is whether the action selection mechanism based on progress leads to a different behavior towards the light. We define the number of times the light has been switch on as :

$$L(t) = \sum_{j=0}^{t} light(t) \tag{8}$$

To evaluate the difference in the behavior, we define the comparative ratio :

$$C_L(t) = \frac{L_{MAXPROGRESS}(t)}{L_{RANDOM}(t)} \tag{9}$$

Simulation results. Figure 1 shows the evolution of $C_P(t)$ and $C_L(t)$ for 10 000 time steps. In the beginning of the evolution, the random strategy outperforms the action selection mechanism. In a situation where very few learning situations are present, choosing actions based on anticipated progress is a worse strategy than choosing actions at random.

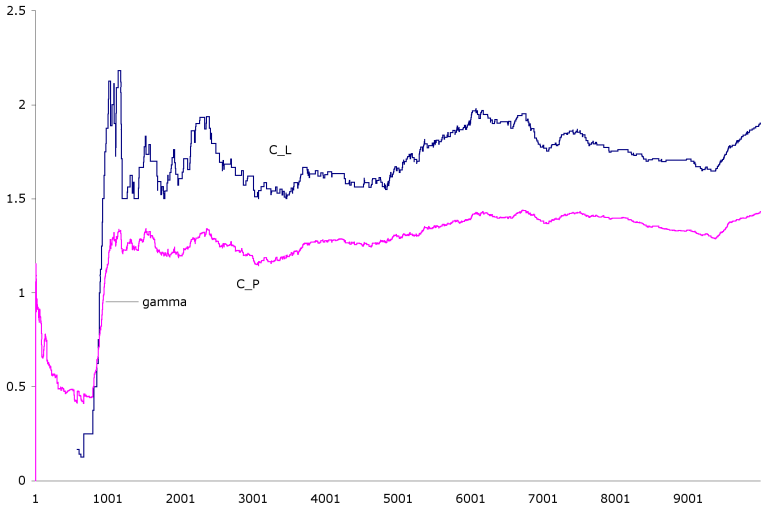

Fig. 1. Evolution of the comparative progress $C_P(t)$ and comparative light ratio $C_L(t)$ for 10 000 time steps

At $t = 579$, the agent using the action selection mechanism switches the light on by chance. From that moment, its starts progressing rapidly. The strategy outperforms the random one around $t = 1000$ and will continue since then.

The $C_L(t)$ curves follows a similar evolution as $C_P(t)$. Random action choices lead first to more situations where the light is on. But as soon as the possibility to switch on the light is "discovered" by chance by the agent, the action selection mechanism exploits that source of learning opportunities by switching the light on much more often than in the random case.

Such behavior is not complex in itself. It would have been easy to define a system to reward the agent when the light is switched on and off. The external behavior of such an agent may have been similar to the one of this experiment. The difference is that our agent had no a priori bias towards that particular stimulus. In this case, the light switching behavior is an "emergent property" resulting of a generic internal reward principle.

We can define γ, the point of transition where $C_P(t)$ becomes superior to 1. Once γ is reached, the action selection strategy outperforms the random strategy. In this simulation γ corresponds also to the transition when the agent starts switching on the light more often than by random moves.

Figure 1 shows a dramatic increase of $C_P(t)$ just after the "discovery of the light". But after γ, $C_P(t)$ reaches a stationary regime where the progress performances are not very different from the one obtained with the random strategy. A similar pattern is observed with the light ratio $C_L(t)$. The light is maximally switched on just after γ. After this initial drive for learning how to switch the light on or off, the system reaches a kind of *habituation* phase. There are no more opportunities for learning. Progress is not zero because there are still some residual errors in prediction to be improved. But the learning progresses as fast as with the random strategy.

4 Embodiment in an active vision system

The embodiment we present in this section shares some similarity with an active vision system described by Marocco and Floreano [18, 19]. But the latter uses an evolutionary robotics paradigm: population of robots are evolved and the best individuals are selected according to a predefined fitness function. We use this embodiment in a developmental perspective.

The system is equipped with a squared retina using RxR perceptual cells. It can move this retina in an image and zoom in and out. Based on the zooming factor, the retina averages the color of the image in order to produce a single value for each cell of the retina. With such a system, it is possible to rapidly scan the patterns present in the overall image and zoom in to perceive some details more accurately. The system has to learn how to "act" on the image by moving and zooming the retina in order to get the higher reward as defined by its reward system.

More precisely, for a given image snapshot $I(t)$, the sensory vector $S(t)$ contains the renormalized grayscale value of the RxR pixels of the retina, its current

position ($X(t)$,$Y(t)$) and zoom factor $Z(t)$. The motor vector $M(t)$ contains the values for the three possible actions the device can perform: changing the X and Y values and the zooming factor Z. [4]

$$S(t) = \begin{vmatrix} Pix_{1,1}(t) \\ Pix_{2,1}(t) \\ ... \\ Pix_{R,R}(t) \\ X(t) \\ Y(t) \\ Z(t) \end{vmatrix}, M(t) = \begin{vmatrix} D_X(t) \\ D_Y(t) \\ D_Z(t) \end{vmatrix} \tag{10}$$

The system is presented with a white image where a grey circle is drawn in the down right corner (Figure 2). This situation shares a lot of similarities with the light switching environment previously studied: the environment is uniform except in a small zone and at a given time t, the agent only perceives a small part of it. But the sensory-motor know-how to be developed to master the retina is much more complex.

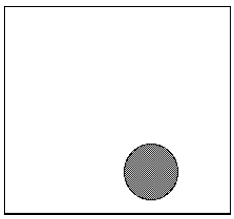

Fig. 2. Image used for the active vision experiment

The system is equipped with a 5x5 retina, so S(t) is of size $5 * 5 + 3 = 28$ and SMR(t) of size $28 + 3 + 1 = 32$. As in the previous experiments, Π_s, Π_m, Π_r are three prototype-based predictors with a maximum capacity of 500 prototypes, $T = 2$ and $K = 1$. Three simulations where conducted : one with $\eta_1 = 0.01$, one with $\eta_2 = 0$ and one with actions chosen randomly. $C_{P1}(t)$ and $C_{P2}(t)$ are defined as previously. By analogy with the light example, we can define $l(t)$ which has value 1 when the center of retina is inside the gray circle and 0 otherwise. $C_{L1}(t)$ and $C_{L2}(t)$ defined as previously measure the relative focus on that part of the image in comparison with a retina governed by random commands.

Figure 3 shows the evolution of $C_{P1}(t)$, $C_{L1}(t)$, $C_{P2}(t)$ and $C_{L2}(t)$. With $\eta_1 = 0.01$, the system discovers rapidly that to focus on the grey circle lead to

[4] It is not sure that the global position information $X(t)$, $Y(t)$, $Z(t)$ are necessary for the system to work as theoretically they can be deduced from the temporal integration of $D_X(t)$, $D_Y(t)$, $D_Z(t)$. However, in practice, it was difficult for the system to discover how the "borders" of the image constrained the retina's movements, without using this positional information.

more learning progress. $C_{P1}(t)$ and $C_{L1}(t)$ show similar features than the ones observed for the light switching problem : (1) the action selection mechanism outperforms clearly the random strategy, (2) once the "interesting" part of the environment is discovered the agent focuses mainly on it for a while, (3) eventually an habituation phase is observed. However with $\eta_2 = 0$, the performances are worse than random. A study of the trajectory shows that the agent focuses on corner instead of focusing on the grey circle. As the sensory-motor-reward space is larger then in the previous system we are confronted with a classical exploration/exploitation trade-off. An optimal strategy may consist in an adaptive system evaluating $C_P(t)$, increasing η when $C_P(t) < 1$, reducing it when $C_P(t) > 1$.

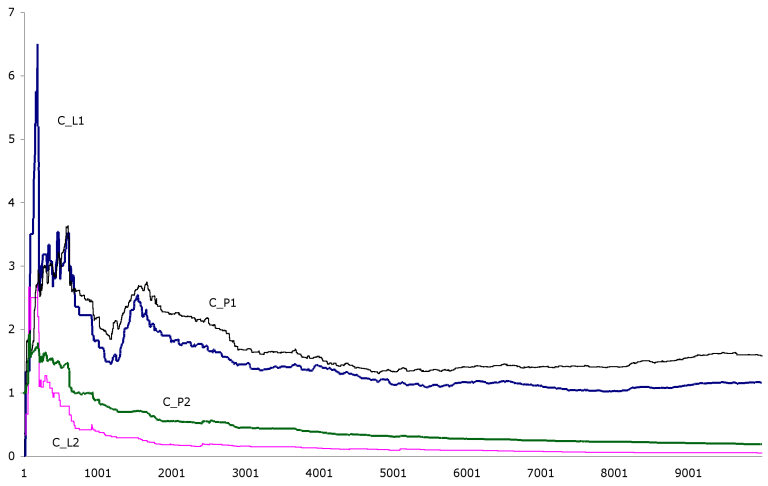

Fig. 3. Evolution of the comparative progress $C_P1(t)(\eta = 0.01)$ and $C_P2(t)(\eta = 0)$and comparative focus on the grey circle $C_L1(t)(\eta = 0.01)$ and $C_L2(t)(\eta = 0.0)$for 10 000 time steps

5 Conclusions

An agent motivated by maximizing learning progress constructs its behavior in order to go from unpredictable situations to predictable ones. Instead of focusing on situations that it predicts well (minimizing prediction error) or on situations it does not predict at all (maximizing prediction error) it focuses on the frontier that separates mastered know-how from unmastered know-how. This strategy enables the development of more complex behaviors in a given environment. In the two experiments described in this chapter, the agent discovers a feature of its environment that is only visible when specific actions are performed. As soon as

the agent discovers this feature, it tries to learn about the sensory effects of its actions in this context. Once its learning progress ceases to increase, the agent stops focusing on this sensory-motor trajectory as it is now part of well predicted situations. This focusing behavior can be seen as a novel behavior mastered by the agent.

One major challenge for developmental robotics is to build a single architecture that would enable a robot to develop novel behaviors of increasing complexity. Can the same mechanisms enable a robot to autonomously learn how to avoid obstacles, discover how to use objects or structure its interactions with other robots? We believe reward systems like the one presented in this paper can play a key role to build robots capable of open-ended development. However, the reward system in itself is not sufficient. The development of complex behaviors results from the interplay between generic motivational principles, particular embodiment (including a particular physical structure, perceptual and motor apparatus and learning techniques) and environmental dynamics (see for instance the experiments discussed in [20] in this volume). For these reasons, further research in that direction should focus on exploring how generic principles such as the one presented in this chapter can be used in experiments with grounded and situated robotic agents.

Acknowledgements

The authors would like to thank Luc Steels and the members of the developmental robotics team (Verena Hafner, Claire d'Este and Andrew Whyte) for precious comments on this work.

References

1. Skinner, B.: The Behavior of Organisms. Appleton Century Crofs, New York, NY. (1938)
2. Sutton, R., Barto, A.: Reinforcement learning: an introduction. MIT Press, Cambridge, MA. (1998)
3. Montague, P., Dayan, P., Sejnowski, T.: A framework for mesencephalic dopamine systems based on predictive hebbian learning. Journal of Neuroscience **16** (1996) 1936–1947
4. Schultz, W., Dayan, P., Montague, P.: A neural substrate of prediction and reward. Science **275** (1997) 1593–1599
5. Doya, K.: Metalearning and neuromodulation. Neural Networks **15** (2002)
6. Lorenz, K.: Vom Weltbild des Verhaltensforschers. dtv, Munchen (1968)
7. Csikszenthmihalyi, M.: Flow-the psychology of optimal experience. Harper Perennial (1991)
8. Kaplan, F., Oudeyer, P.Y.: Motivational principles for visual know-how development. In Prince, C., Berthouze, L., Kozima, H., Bullock, D., Stojanov, G., Balkenius, C., eds.: Proceedings of the 3rd international workshop on Epigenetic Robotics : Modeling cognitive development in robotic systems. Number 101, Lund University Cognitive Studies (2003) 73–80

9. Varela, F., Thompson, E., Rosch, E.: The embodied mind : Cognitive science and human experience. MIT Press, Cambridge, MA (1991)
10. Andry, P., Gaussier, P., Moga, S., Banquet, J., Nadel, J.: Learning and communication in imitation: an autonomous robot perspective. IEEE Transaction on Systems, Man and Cybernetics, Part A : Systems and Humans **31** (2001) 431–444
11. Huang, X., Weng, J.: Novelty and reinforcement learning in the value system of developmental robots. In: Proceedings of the 2nd international workshop on Epigenetic Robotics - Lund University Cognitive Studies 94. (2002) 47–55
12. Thrun, S.: Exploration in active learning. In Arbib, M., ed.: Handbook of Brain Science and Neural Networks. MIT Press (1995)
13. Schmidhuber, J.: Curious model-building control systems. In: Proceeding International Joint Conference on Neural Networks. Volume 2., Singapore, IEEE (1991) 1458–1463
14. Elman, J.: Finding structure in time. Cognitive Science **14** (1990) 179–211
15. Rabiner, L., Juang, B.: An introduction to hidden markov models. IEEE Acoutics, Speech and Signal Processing Magazine **3** (1986) 4–16
16. Tani, J., Nolfi, S.: Learning to perceive the world as articulated : An approach for hiearchical learning in sensory-motor systems. Neural Network **12** (1999) 1131–1141
17. Jordan, M., Jacobs, R.: Hierarchical mixtures of experts and the em algorithm. Neural Computation **6** (1994) 181–214
18. Kato, T., Floreano, D.: An evolutionary active-vision system. In: Proceedings of the congress on evolutionary computation (CEC01), IEEE Press (2001)
19. Marocco, D., Floreano, D.: Active vision and feature selection in evolutionary behavioral systems. In Hallam, B., Floreano, D., Hallam, J., Hayes, G., Meyer, J.A., eds.: From Animals to Animats 7, Cambridge, MA., MIT Press (2002)
20. Lungarella, M., Berthouze, L.: Robot bouncing : On the interaction between body and environmental dynamics. this volume (2004)

Appendix

This appendix describes a step by step evolution of the system for the light switching problem. In this example the parameter η is set to 0.01. Let's assume that the system has been active for t time steps and that during that period the light stayed at value 0. For the sake of simplicity, we will make the unrealistic assumption that the predictors learn rapidly and "perfectly" based on the situations they encountered. So we will assume that the predictors have learned to predict that $pos(t) = nextpos(t-1)$ and that light(t) = 0.

Time t. At time t the situation is the following.

$$e(t-1) = 0, SMR(t-1) = \begin{vmatrix} pos(t) = 0.4 \\ light(t) = 0 \\ nextpos(t) = 0.2 \\ p(t) = 0 \end{vmatrix}, SMR(t) = \begin{vmatrix} pos(t) = 0.2 \\ light(t) = 0 \\ nextpos(t) = ? \\ p(t) = ? \end{vmatrix} \quad (11)$$

The "perfect" predictor for sensory information, Π_s, gave the following predictions:

$$\Pi_s(SMR(t-1)) \rightarrow \begin{vmatrix} position(t) = 0.2 \\ light(t) = 0 \end{vmatrix} \quad (12)$$

So $e(t) = 0$ and $p(t) = e(t-1) - e(t) = 0$.

The role of the action selection module is to determine $nextposition(t)$. The average progress, close to zero, is currently inferior to η. Acting randomly, the system chooses $nextposition(t) = 0.9$. The vector $SMR(t)$ is now completed. The three predictors can learn by comparing $SMR(t)$ with the values predicted based on $SMR(t-1)$.

Time t+1. As the agent moved to 0.9, the light was switched on. Consequently, the situation at time t+1 is the following, and Π_s could not predict that evolution.

$$SMR(t+1) = \begin{vmatrix} pos(t+1) = 0.9 \\ light(t+1) = 1 \\ nextpos(t+1) = ? \\ p(t+1) = ? \end{vmatrix}, \Pi_s(SMR(t)) \rightarrow \begin{vmatrix} pos(t+1) = 0.9 \\ light(t+1) = 0 \end{vmatrix} \tag{13}$$

So $e(t+1) = (0+1)/2 = 0.5$ and as progress is negative $p(t+1) = 0$. Next action is random : $nextpos(t+1) = 0.7$. Predictors learn.

Time t+2. Situation at $t+2$ is the following

$$SMR(t+2) = \begin{vmatrix} pos(t+2) = 0.7 \\ light(t+2) = 0 \\ nextpos(t+2) = ? \\ p(t+2) = ? \end{vmatrix} \tag{14}$$

Let's assume that the predictor has predicted correctly that the light will be switched off after that move.

$$\Pi_s(SMR(t+1)) \rightarrow \begin{vmatrix} pos(t+2) = 0.7 \\ light(t+2) = 0 \end{vmatrix} \tag{15}$$

So $e(t+2) = 0$ and $p(t+2) = 0.5 - 0 = 0.5$. Because $p(t+2) > \eta$, next action will be chosen through anticipation. The system creates a set of possible values for *nextpos* and tries to predict their effects in terms of rewards. If the system only looks one step ahead no reward can be anticipated. But if the system looks at least two steps ahead, it can anticipate that choosing *nextpos* near 0.9 will lead to a situation similar to the one experienced at time $t+1$. Using predictor Π_m to simulate what it would do next in such a situation and Π_r to evaluate the associated expected reward a total anticipation of the situation at t+3 is possible.

$$\Pi(\begin{vmatrix} pos(t+2) = 0.7 \\ light(t+2) = 0 \\ nextpos(t+2) = 0.91(tried) \\ p(t+2) = 0.5 \end{vmatrix}) \rightarrow \begin{vmatrix} pos(t+3) = 0.91(\Pi_s) \\ light(t+3) = 1(\Pi_s) \\ nextpos(t+3) = 0.7(\Pi_m) \\ p(t+3) = 0(\Pi_r) \end{vmatrix} \tag{16}$$

The system can then use Π_r to anticipate the expected reward at $t+4$ (e.g. $p(t+4) = 0.5$). Based on this anticipation the system will decide to move to 0.91 in order to experience again the transition that lead to an increase of $p(t)$. This example illustrates one possible way for the system to optimize the learning progress $p(t)$. The variable $p(t)$ is positive when the system moves from an unpredicted situation to a predictable one. This means that the system is rewarded when it "returns" to known situations. But to be "returned", the system must first leave the situations it anticipates well.