Combining self-organizing maps with mixtures of experts: Application to an Actor-critic model of reinforcement learning in the Basal Ganglia

Mehdi Khamassi[1,2], Louis-Emmanuel Martinet[1], and Agnès Guillot[1]

[1] Université Pierre et Marie Curie - Paris 6, UMR7606, AnimatLab - LIP6, F-75005 Paris, France ; CNRS, UMR7606, F-75005 Paris, France
[2] Laboratoire de Physiologie de la Perception et de l'Action, UMR7152 CNRS, Collège de France, F-75005 Paris, France
{mehdi.khamassi, louis-emmanuel.martinet, agnes.guillot}@lip6.fr
http://animatlab.lip6.fr

Preprint accepted for publication in SAB06, Roma, Italy, 25-29 sept. 2006.

Abstract. In a reward-seeking task performed in a continuous environment, our previous work compared several Actor-Critic (AC) architectures implementing dopamine-like reinforcement learning mechanisms in the rat's basal ganglia. The task complexity imposes the coordination of several AC submodules, each module being an expert trained in a particular subset of the task. We showed that the classical method where the choice of the expert to train at a given time depends on each expert's performance suffered from strong limitations. We rather proposed to cluster the continuous state space by an *ad hoc* method that lacked autonomy and generalization abilities. In the present work we have combined the mixture of experts with self-organizing maps in order to cluster autonomously the experts' responsibility space. On the one hand, we find that classical *Kohonen maps* give very variable results: some task decompositions provide very good and stable reinforcement learning performances, whereas some others are unadapted to the task. Moreover, they require the number of experts to be set a priori. On the other hand, algorithms like *Growing Neural Gas* or *Growing When Required* have the property to choose autonomously and incrementally the number of experts to train. They lead to good performances, even if they are still weaker than our hand-tuned task decomposition and than the best Kohonen maps that we got. We finally discuss on propositions about what information to add to these algorithms, such as knowledge of current behavior, in order to make the task decomposition appropriate to the reinforcement learning process.

1 Introduction

In the frame of the Psikharpax project, which aims at building an artificial rat having to survive in complex and changing environments, and having to satisfy different needs and motivations [5][14], our work consists in providing a simulated robot with habit learning capabilities, in order to make it able to associate efficient behaviors to relevant stimuli located in an unknown environment.

The control architecture of Psikharpax is expected to be as close as possible to known anatomy and physiology of the rat brain, in order to unable comparison between functioning of the model with electrophysiological and behavioral recordings. As a consequence, our model of reinforcement learning is based on an Actor-Critic architecture inspired from basal ganglia circuits, following well established hypotheses asserting that this structure of the mammalian brain is responsible for driving action selection [16] and reinforcement learning of behaviors to select via substantia nigra dopaminergic neurons [17].

At this stage of the work, our model runs in 2D-simulation with a single need and a single motivation. However the issue at stake already has a certain complexity: it corresponds to a continuous state-space environment; the perceptions have non monotonic changes; an obstacle-avoidance reflex can interfere with actions selected by the model; the reward location provides a non instantaneous reward. In a previous paper [11], we demonstrated that this task complexity requires the use of multiple Actor-Critic modules, where each module is an expert trained in a particular subset of the environment. We compared different hypotheses concerning the management of such modules, concerning there more or less autonomously determined coordination, and found that the classical mixture of experts method - where the choice of the expert to train at a given time depends on each expert's performance [3][4] – cannot train more than one single expert in our reinforcement learning task. We rather proposed to cluster the continuous state space and to link each expert to a cluster by an ad hoc method that could indeed solve the task, but that lacked autonomy and generalization abilities.

The objective of the present work is to provide an autonomous categorization of the state space by combining the mixture of experts with self-organizing maps (SOM). This combination has already been implemented by Tang et al. [20] - these authors having criticized the undesirable effects of classical mixture of experts on boundaries of non disjoint regions. However, they did not test the method in a reinforcement learning task. When they were used in such tasks [18][13] - yet without mixture of experts –, SOM were applied to the discretization of the input space to the reinforcement learning model, which method suffers from generalization abilities. Moreover, the method has limited performance in high-dimensional spaces and remains to be tested robustly on delayed reward tasks.

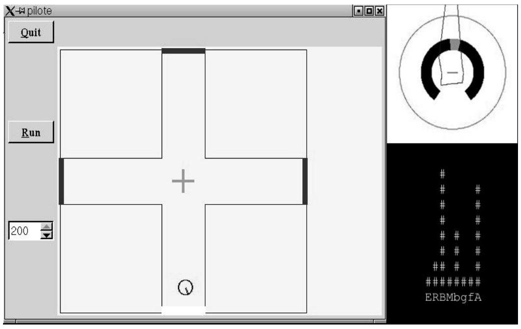
In our case, we propose that the SOM algorithms have to produce a clustering of the responsibility space of the experts, in order to decide which Actor-Critic expert has to work in a given zone of the perceptual state space. In addition, the selected Actor-Critic expert of our model will receive the entire state space, in order to produce a non constant reward prediction inside the given zone.

After describing the task in the following section, we will report the test of three self-organizing maps combined with the mixture of Actor-Critic experts, for the comparison of their usefulness for a complex reinforcement learning task. It concerns the classical *Kohonen* algorithm [12], which requires the number of experts to be a priori set; the *Growing Neural Gas* algorithm [6], improved by [9], which adds a new expert when an existing expert has a important error of classification; and the *Growing When Required* algorithm [15], which creates a new expert when habituation of the map to visual inputs produces a too weak output signal when facing new visual data.

In the last section of the paper, we will discuss the possible modifications that could improve the performance of the model.

2 The task

Figure 1 shows the simulated experimental setup, a simple 2D plus-maze. The dimensions are equivalent to a 5m * 5m environment with 1m large corridors. In this environment, walls are made of segments colored on a 256 grayscale. The effects of lighting conditions are not simulated. Every wall of the maze is colored in black (luminance = 0), except walls at the end of each arm and at the center of the maze, which are represented by specific colors: the cross at the center is gray (191), three of the arm ends are dark gray (127) and the fourth is white (255), indicating the reward location equivalent to a water trough delivering two drops (non instantaneous reward) – not a priori known by the animat.



```
  #
  #    #
  #    #
  #    #
  # #  #
  # #  #
 ## #  #
########
ERBMbgfA
```

Fig. 1. Left: the robot in the plus-maze environment. Upper right: the robot’s visual perceptions. Lower right: activation level of different channels in the model.

The plus-maze task reproduces the neurobiological and behavioral experiments that will serve as future validation for the model [1]. At the beginning of each trial, one arm end is randomly chosen to deliver reward. The associated wall becomes white whereas the other arm ends become dark gray. The animat has to learn that selecting the action *drinking* when it is near the white wall (distance < 30 cm) and faces it (angle < 45°) gives it two drops of water. Here we assume that reward = 1 for n iterations (n = 2) during which the action *drinking* is being executed. However, the robot's vision does not change between these two moments, since the robot is then facing the white wall. As visual information is the only sensory modality that will constitute the input space of the Actor-Critic model, this makes the problem to solve a Partially Observable Markov Decision Process [19]. This characteristic was set in order to fit the multiple consecutive rewards that are given to rats in the neurobiological plus-maze, enabling comparison between our algorithm with the learning process that takes place in the rat brain during the experiments.

We expect the animat to learn a sequence of context-specific behaviors, so that it can reach the reward site from any starting point in the maze:

- When not seeing the white wall, face the center of the maze and move forward
- As soon as arriving at the center (the animat can see the white wall), turn to the white stimulus
- Move forward until being close enough to reward location

- Drink

The trial ends when reward is consumed: the color of the wall at reward location is changed to dark gray, and a new arm end is randomly chosen to deliver reward. The animat has then to perform another trial from the current location. The criterion chosen to validate the model is the time – number of iterations of the algorithm - to goal, plotted along the experiment as the learning curve of the model.

3 The animat

The animat is represented by a circle (30 cm diameter). Its translation and rotation speeds are 40 cm.s^{-1} and 10°.s^{-1}.

Its simulated sensors are:

- Eight sonars with a 5m range, an incertitude of ±5 degrees concerning the pointed direction and an additional ±10 cm measurement error. The sonars are used by a low level obstacle avoidance reflex which overrides any decision taken by the Actor-Critic model when the animat comes too close to obstacles.
- An omnidirectional linear camera providing every 10° the color of the nearest perceived segment. This results in a 36 colors table that constitute the animat's visual perception (see figure 1).

The animat is provided with a visual system that computes 12 input variables and a constant equal to 1 $(\forall i \in [1;13], 0 \leq \mathrm{var}_i \leq 1)$ out of the 36 colors table at each time step. These sensory variables constitute the state space of the Actor-Critic and so will be taken as input to both the Actor and the Critic parts of the model (figure 3). Variables are computed as following:

- *seeWhite* (resp. *seeGray*, *seeDarkGray*) = 1 if the color table contains the value 255 (resp. 191, 127), else 0.
- *angleWhite*, *angleGray*, *angleDarkGray* = (number of boxes in the color table between the animat's head direction and the desired color) / 18.
- *distanceWhite*, *distanceGray*, *distanceDarkGray* = (maximum number of consecutive boxes in the color table containing the desired color) / 18.
- *nearWhite* (resp. *nearGray, nearDarkGray*) = 1 – *distanceWhite* (resp. *distanceGray, distanceDarkGray*).

The model permanently receives a flow of sensory information and has to learn autonomously the sensory contexts that can be relevant for the task resolution.

The animat has a repertoire of 6 actions: *drinking*, *moving forward*, *turning to white perception, turning to gray perception, turning to dark gray perception*, and *waiting*. These actions constitute the output of the Actor model (described below) and the input to a low-level model that translates it into appropriate orders to the animat's engines.

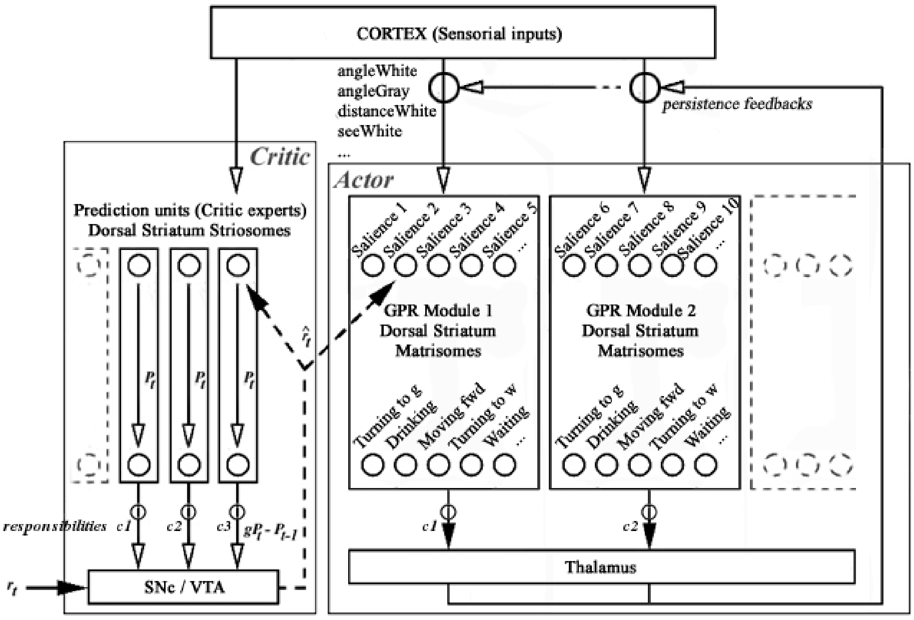


Fig. 2. General scheme of the model tested in this work. The Actor is a group of “GPR” modules [8] with saliences as inputs and actions as outputs. The Critic (involving striosomes in the dorsal striatum, and the substantia nigra compacta (SNc)) propagates towards the Actor an estimate $\check{r}$ of the instantaneous reinforcement triggered by the selected action. The particularity of this scheme is to combine several modules for both Actor and Critic, and to gate the Critic experts’ predictions and the Actor modules’ decisions with responsibility signals. These responsibilities can be either computed by a Kohonen, a GWR or a GNG map.

4 The Model

4.1 The multi-module Actor-Critic

The model tested in this work has the same general scheme than described in [11]. It has two main components, an Actor which selects an action depending on the visual perceptions described above; and a Critic, having to compute predictions of reward based on these same perceptions (figure 2). Each of these two components is composed of N submodules or *experts*. At a given time, each submodule k $(k \in [1\ ;N])$ has a responsibility $c_k(t)$ that determines its weight in the output of the overall model. In the context of this work, we restrict to the case where only one expert *k* has its responsibility equal to 1 at a given moment, and $\forall\, j \neq k,\ \mathrm{c}_j(t) = 0$.

Inside the Critic component, each submodule is a single linear neuron that computes its own prediction of reward:

$$p_k(t) = \sum_{j=1}^{13} w'_{k,j}(t) \cdot \text{var}_j(t) \qquad (1)$$

where $w'_{k,j}(t)$ is the synaptic weight of expert *k* representing the association strength with input variable j. Then the global prediction of the Critic is a weighted sum of experts' predictions:

$$P(t) = \sum_{k=1}^{N} c_k(t) \cdot p_k(t) \qquad (2)$$

Concerning the learning rule, derived from the Temporal-Difference Learning algorithm [19], each expert has a specific reinforcement signal based on its own prediction error:

$$\hat{r}_k(t) = r(t) + gP(t) - p_k(t-1) \qquad (3)$$

The synaptic weights of each expert *k* are updated according to the following formula:

$$w'_{k,j}(t) \leftarrow w'_{k,j}(t-1) + \eta \cdot \hat{r}_k(t) \cdot \text{var}_j(t-1) \cdot c_k(t) \qquad (4)$$

Actor submodules also have synaptic weights $w_{i,j}(t)$ that determine, inside each submodule *k*, the salience – i.e. the strength – of each action *i* according to the following equation:

$$sal_i(t) = \left[\sum_{j-1}^{13} \text{var}_j(t) \cdot w_{i,j}(t) \right] + persist_i(t) \cdot w_{i,14}(t) \qquad (5)$$

The action selected by the Actor to be performed by the animat corresponds to the strongest output of the submodule with responsibility 1. If a reinforcement signal occurs, the synaptic weights of the latter submodule are updated following equation (4).

An exploration function is added that would allow the animat to try an action in a given context even if the weights of the Actor do not give a sufficient tendency to perform this action in the considered context.

To do so, we introduce a clock that triggers exploration in two different cases:

- When the animat has been stuck for a large number of timesteps (*time* superior to a fixed threshold α) in a situation that is evaluated negative by the model (when the prediction *P*(t) of reward computed by the Critic is inferior to a fixed threshold).
- When the animat has remained for a long time in a situation where *P*(t) is high but this prediction doesn't increase that much ($|P(t+n) - P(t)| < \epsilon$) and no reward occurs.

If one of these two conditions is true, exploration is triggered: one of the 6 actions is chosen randomly. Its salience is being set to 1 (Note that: when exploration = false,

$sal_i(t) < 1, \forall i, t, w_{i,j}(t)$) and is being maintained to 1 for a duration of 15 timesteps (time necessary for the animat to make a 180° turn or to run from the center of the maze until the end of one arm).

4.2 The self-organizing maps

In our previous work [11], we showed that the classical method used to determine the experts' responsibilities – a gating network, giving the highest responsibility to the expert that approximates the best the future reward value [3][4] – was not appropriate for the resolution of our reinforcement learning task. Indeed, we found that the method could only train one expert which would remain the more responsible in the entire state space without having a good performance. As our task is complex, we rather need the region of the state space where a given expert is the most responsible to be restricted, in order to have only limited information to learn there. As a consequence, we propose that the state space should be clustered independently from the performance of the model in learning the reward value function.

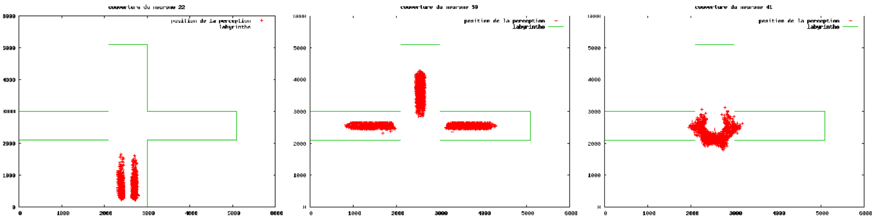


Fig. 3. Examples of clusterings found by the GWR self-organizing map. The pictures show, for three different AC experts, the positions of the robot for which the expert has the highest responsibility – thus, positions where the Actor-Critic expert is involved in the learning process.

In this work, the responsibility space of the Actor-Critic experts is determined by one of the following self-organizing maps (SOMs): the Kohonen Algorithm, the Growing Neural Gas, or the Growing When Required. We will describe here only essential aspects necessary for the comprehension of the method maps. Each map has a certain number of nodes, receives as an input the state space constituted of the same perception variables than the Actor-Critic model, and will autonomously try to categorize this state space. Training of the SOMs is processed as following:

```
Begin
  Initialize a fixed number of nodes (for the Kohonen
  Map) or 2 nodes for GNG and GWR algorithms;
  While (iteration < 50000)
    Move the robot randomly; //Actor-Critic disabled
    Try to categorize the current robot's perception;
    If (GNG or GWR) and (classification-error > threshold)
        Add a new node to the map;
    End if;
    Adapt the map;
  End;
  // After that, the SOM won’t be adapted anymore
  While (trial < 600)
    Move the robot with the Actor-Critic (AC) model;
    Get the current robot’s perception;
    Find the SOM closest node (k) to this perception;
    Set expert k responsibility to 1 and others to 0;
    Compute the learning rule and adapt synaptic weights of the AC;
  End;
End;
```

Parameters used for the three SOM algorithms are given in the appendix table. Figure 3 shows some examples of categorization of the state space obtained with a GWR algorithm. Each category corresponds to a small region in the plus-maze, where its associated Actor-Critic expert will have to learn. Notice that we set the parameters so that regions are small enough to train at least several experts, and large enough to require that some experts learn to select different actions successively inside the region.

5 Results

The results correspond to several experiments of 600 trials for each of the three different methods (11 with GWR, 11 with GNG, and 11 with Kohonen maps). Each experiment is run following the algorithmic procedure described in the previous section.

Table 1. Summarized performances of the methods applied to reinforcement learning.

Method	Average performance during second half of the experiment (nb iterations per trials)	Standard error	Best map's average performance
Hand-tuned map	93.71	N/A	N/A
KOH (n=11)	548.30	307.11	87.87
GWR (n=11)	459.72	189.07	301.76
GNG (n=11)	403.73	162.92	193.39

Figure 4 shows the evolution with time of the learning process of each method. In each case, the smallest number of iterations occurs around the 250th trial and remains stabilized. Table 1 summarizes the global performances averaged over the second half of the experiment – e.g. after trial #300. Performances of the three methods are comparable (Kruskall-Wallis test reveals no significant differences: $p > 0.10$). When looking at the maps' categorizations precisely and independently from the reinforcement learning process, measure of the maps' errors of categorization highlights that Kohonen maps provide a slightly worst result in general, even while using more neurons than the GWR and GNG algorithms. However, this doesn't seem to have consequences on the reinforcement learning process, since performances are similar. So, the Kohonen algorithm, whose number of experts is a priori set, is not better than the two others which recruit new experts autonomously.

Performances with GNG and GWR algorithms are not very different either. In their study, Marsland et al. [15] conclude that GWR is slightly better than the GNG algorithm in its original version. Here, we used a modified version of GNG [9]. In our simulations, the GNG recruited on average less experts than the GWR but had a classification error a little bigger. However, when applied to reinforcement learning, the categorizations provided by the two algorithms did not show major differences.

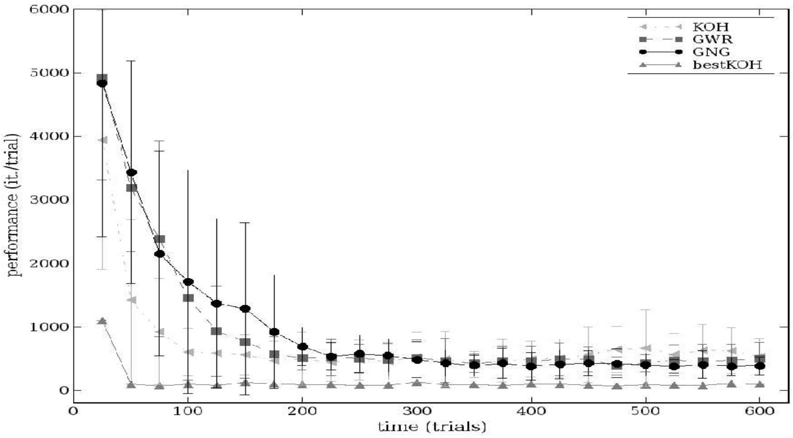


Fig. 4. Learning curves of the reinforcement learning experiments tested with different self-organizing maps.

Qualitatively, the three algorithms have provided the multi-module Actor-Critic with quite good experts' responsibility space clustering, and the animat managed to learn an appropriate sequence of actions to the reward location. However, performances are still not as good as a version of the model with hand-tuned synaptic weights. The latter has an average performance of 93.71 iterations per trial, which is characterized by a nearly “optimal” behavior where the robot goes systematically straight to the reward location, without loosing any time (except the regular trajectory deviation produced by the exploration function of the algorithm). Some of the best Kohonen maps and GNG maps reached similar nearly optimal behavior. As shown in table 1, the best Kohonen map got an average performance of 87.87 iterations per trial. Indeed, it seems that the categorization process can produce very variable reinforcement learning depending on the map built during the first part of the experiment.

6 Discussion

In this work, we have combined three different self-organizing maps with a mixture of Actor-Critic experts. The method was designed to provide an Actor-Critic model with autonomous abilities to recruit new expert modules for the learning of a reward-seeking task in continuous state space. Provided with such a control architecture, the simulated robot can learn to perform a sequence of actions in order to reach the reward. Moreover, gating Actor-Critic experts with our method strongly ressembles neural activity observed in the striatum – e.g. the input structure of the basal ganglia – in rat performing habit learning tasks in an experimental maze [10]. Indeed, the latter study

shows striatal neurons' responses that are restricted to localized chunks of the trajectory performed by the rat in the maze. This is comparable with the clusters of experts' responsibilities shown in figure 3.
However, the performance of the model presented here remains weaker than a hand-tuned behavior. Indeed, the method produces very variables results, from maps with nearly optimal performance to maps providing unsatisfying robotics behavior.
Analysis of the maps created with our method shows that some of them are more appropriate to the task than others, particularly when the boundaries between two experts' receptive fields corresponds to a region of the maze where the robot should switch from one action to another in order to get the reward. As an example, we noticed that the majority of the maps obtained in this work had their expert closer to the reward location with a too large field of responsibility. As a consequence, the trunk of the global value function that this expert has to approximate is more complex, and the behavior to learn is more variable. This results in selecting inappropriate behavior in the field of this expert – for example, the robot selects the action “drinking” too far from reward location to get a reward. Notice however that this is not a problem with selecting several different actions in the same region of the maze, since some experts managed to learn to alternate between two actions in their responsibility zone, for example in the area close to the center of the plus-maze. A given expert having limited computational capacities, its limitations occur when its region of responsibility is too large.

To improve the performance, one could suggest setting parameters of the SOM in order to increase the number of experts in the model. However, this would result in smaller experts' receptive field than those presented in figure 3. As a consequence, each expert would receive a nearly constant input signal inside its respective zone, and would need only to select one action. This would be computationally equivalent to the use of small fields place cells for the clustering of the state space of an Actor-Critic, which has been criticized by several authors [2], and would not be different than other algorithms where the winning node of a self-organizing map produces a discretization of the input space to a reinforcement learning process [18].

One could also propose to increase each expert-module's computational capacity. For instance, one could use a more complex neural network than the single linear neuron that we implemented for each expert. However, one cannot a priori know the task complexity, and no matter the number of neurons an expert possesses, there could still exist too complex situations. Moreover, “smart” experts having a small responsibility region could overlearn the data with poor generalization ability [7].

7 Perspective

In future work, we rather propose to enable the experts' gating to adapt slightly to the behavior of the robot. The management of experts should not be mainly dependent on the experts' performances in controlling behavior and estimating the reward value, as we have shown in previous work [11]. However, considering the categorization of the visual space as the main source of experts’ specialization, it could be useful to add information about the behavior in order for boundaries between two experts' responsibility regions to flexibly adapt to areas where the animat needs to switch its behavior.

In [21], the robot's behavior is a priori set and stabilized, and constitutes one of the inputs to a mixture of experts having to categorize the sensory-motor flow perceived by a robot. In our case, at the beginning of the reinforcement learning process, when behavior is not yet stable, visual information could be the main source of experts' specialization. Then, when the model starts to learn an appropriate sequence of actions, behavioral information could help adjusting the specialization. This would be similar to electrophysiological recordings of the striatum showing that, after extensive training of the rats, striatal neurons' responses tend to translate to particular "meaningful" portions of the behavioral sequences, such as the starting point and the goal location [10].

Acknowledgments

This research has been granted by the LIP6 and the European Project Integrating Cognition, Emotion and Autonomy (ICEA). The authors wish to thank Thomas Degris and Jean-Arcady Meyer for useful discussions.

References

1. Albertin, S. V., Mulder, A. B., Tabuchi, E., Zugaro, M. B., Wiener, S. I.: Lesions of the medial shell of the nucleus accumbens impair rats in finding larger rewards, but spare reward-seeking behavior. *Behavioral Brain Research.* 117(1-2) (2000) 173-83
2. Arleo, A., W. Gerstner,W.: Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity. *Biological Cybernetics*, 83(3) (2000) 287-99
3. Baldassarre, G.: A modular neural-network model of the basal ganglia's role in learning and selecting motor behaviors. *Journal of Cognitive Systems Research,* 3 (1) (2002) 5-13
4. Doya, K., Samejima., K., Katagiri, K., Kawato, M.: Multiple model-based reinforcement learning. *Neural Computation,* 14(6) (2002) 1347-69
5. Filliat, D., Girard, B., Guillot, A., Khamassi, M., Lachèze, L., Meyer, J.-A. : State of the artificial rat Psikharpax. In: Schaal, S., Ijspeert, A., Billard, A., Vijayakumar, S., Hallam, J., Meyer, J.-A. (eds): *From Animals to Animats 8: Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior,* Cambridge, MA. MIT Press (2004) 3-12
6. Fritzke, B.: A growing neural gas network learns topologies. In: Tesauro, G, Touretzkys, D.S., Leen, K.(eds): *Advances in Neural Information Processing Systems,* MIT Press, (1995) 625-32
7. Geman, S., Bienenstock, E., Doursat, R.: Neural networks and the bias/variance dilemma. *Neural Computation* 4 (1992) 1-58
8. Gurney, K.,Prescott, T.J., Redgrave, P.: A computational model of action selection in the basal ganglia. I. A new functional anatomy. *Biological Cybernetics,* 84 (2001) 401-10

9. Holmström, J.: Growing neural gas : Experiments with GNG, GNG with utility and supervised GNG. Master's thesis, Uppsala University (2002)
10. Jog, M.S., Kubota, Y., Connolly, C.I., Hillegaart, V., Graybiel, A.M.: Building neural representations of habits. *Science*, 286(5445) (1999) 1745-9
11. Khamassi, M., Lachèze, L., Girard, B., Berthoz, A., Guillot, A: Actor-critic models of reinforcement learning in the basal ganglia: From natural to artificial rats. *Adaptive Behavior, Special Issue Towards Artificial Rodents* 13(2) (2005) 131-48
12. Kohonen, T.: Self-organizing maps. Springer-Verlag, Berlin (1995)
13. Lee, J. K., Kim, I. H.: Reinforcement learning control using self-organizing map and multi-layer feed-forward neural network. In: *International Conference on Control Automation and Systems*, ICCAS 2003 (2003)
14. Meyer, J.-A., Guillot, A., Girard, B., Khamassi, M., Pirim, P., Berthoz, A.: The Psikharpax project: Towards building an artificial rat. *Robotics and Autonomous Systems* 50(4) (2005) 211-23
15. Marsland, S., Shapiro, J., Nehmzow, U.: A self-organising network that grows when required. *Neural Networks*, 15 (2002) 1041-58
16. Prescott, T.J., Redgrave, P., Gurney, K.: Layered control architectures in robots and vertebrates. *Adaptive Behavior, 7* (1999) 99-127
17. Schultz, W., Dayan, P., Montague, P. R. : A neural substrate of prediction and reward. *Science*, 275 (1997) 1593-9
18. Smith, A. J.: Applications of the self-organizing map to reinforcement learning. *Neural Networks* 15(8-9) (2002)1107-24
19. Sutton, R. S., Barto, A. G.: Reinforcement learning: An introduction. The MIT Press Cambridge, MA. (1998)
20. Tang, B., Heywood, M. I.: Shepherd, M.: Input Partitioning to Mixture of Experts. In: *IEEE/INNS International Joint Conference on Neural Networks*, Honolulu, Hawaii (2002) 227-32
21. Tani, J., Nolfi, S.: Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems.*Neural Networks*12(1999)1131-41

Appendix: Parameters table

Symbol	Value	Description
Δt	1 sec.	Time between two successive iterations of the model.
α	[50;100]	Time threshold to trigger the exploration function.
g	0.98	Discount factor of the Temporal Difference learning rule.
η	0.05 / 0.01	Learning rate of the Critic and the Actor respectively.
N	36	Number of nodes in Kohonen Maps.
η-*koh*	0.05	Learning rate in Kohonen Maps.
σ	3	Neighborhood radius in Kohonen Maps.
Ew, En	0.5, 0.005 / 0.1, 0.001	Learning rates in the GNG and GWR respectively.
a-max	100	Max. age in the GNG and GWR.
S		Threshold for nodes recruitment in the GNG.
α-*gng, ß-gng*	0.5, 0.0005	Error reduction factors in the GNG.
λ	1	Window size for nodes incrementation in the GNG.
a-T	0.8	Activity threshold in the GWR
h-T	0.05	Habituation threshold in the GWR.