

A model of neuronal specialization using Hebbian policy-gradient with “slow” noise

Emmanuel Dac  ^{1,2}

¹ INRIA Lille Nord-Europe, Villeneuve d’Ascq, France

² Institute of Movement Sciences, University of the Mediterranean, Marseille, France

Abstract. We study a model of neuronal specialization using a policy gradient reinforcement approach. (1) The neurons stochastically fire according to their synaptic input plus a noise term; (2) The environment is a closed-loop system composed of a rotating eye and a visual punctual target; (3) The network is composed of a foveated retina, a primary layer and a motoneuron layer; (4) The reward depends on the distance between the subjective target position and the fovea and (5) the weight update depends on a Hebbian trace defined according to a policy gradient principle. In order to take into account the mismatch between neuronal and environmental integration times, we distort the firing probability with a “pink noise” term whose autocorrelation is of the order of 100 ms, so that the firing probability is overestimated (or underestimated) for about 100 ms periods. The rewards occurring meanwhile assess the “value” of those elementary shifts, and modify the firing probability accordingly. Every motoneuron being associated to a particular angular direction, we test at the end of the learning process the preferred output of the visual cells. We find that accordingly with the observed final behavior, the visual cells preferentially excite the motoneurons heading in the opposite angular direction.

1 Introduction

The issue of plasticity and neuronal specialization is important in computational neuroscience, in order to explain more realistically the adaptation processes taking place in the brain. When looking at the first steps of sensory processing, for instance, the topology of the captors appears prominent in the shaping of the cortical maps (“somatotopic” layers, “retinotopic” layers, “tonotopic” layers, etc...). This topography is expected to be the result of a progressive plasticity process which shapes the input/output response curve of a population of neurons according to their location, making them more reactive to a particular class of signals. Similar adaptation processes are expected to take place at all the stages of the brain processing, continuously modifying the way the brain will take its decisions and control its movements.

Taking now this problem under a reinforcement learning (RL) framework, a brain is seen as a neural controller facing a complex decision problem, receiving non-stationary perceptions and continuously sending motor commands which immediately take effect in the sensory scene. The relevance of the motor commands is not known, and some signals from the environment (or from the body) indicate to which point the choices made are “good” or “bad”. The controller being composed of a huge population of

neurons, the global decision is the combination of many local infinitesimal decisions (i.e. emitting a spike or not at time t). A local process that would appropriately shape the neuronal decision according to the reward is expected to favour better collective responses, thus improving the performances of the controller.

2 Principles

Classical RL approaches consider value functions which map some (state, action) couples to a value estimating the expectation of future rewards [1]. In the case of a parametrized stochastic policy implemented by binary units, a policy gradient approach [2] avoids such explicit value function and directly shapes the parameters according a local estimation of the gradient of the reward expectation. It ends up as an expression where the weights update depends on the Hebb-like product $[S(t) - f(I(t))] \times \varepsilon_j(t)$ where S is the actual neuron output, f is the firing probability according to the input and ε_j is the pre-synaptic activity. The global process is a gradient ascent algorithm. Bartlett and Baxter [3] show that it can be extended to any network of stochastic neurons where the reward signal is spread to every neuron. In that case, the policy improvements realized locally participate to the improvement of the global response.

Some extensions to more realistic spiking neurons have been proposed : Seung [4] treated the case of stochastic synapses with poisson firing. Baras [5] considers a Spike Response Model (SRM) [6] with a stochastic spike emission mechanism (proportional to the weighted sum of pre-synaptic activities). Florian [7] considers the more classical case of stochastic SRM neurons with escape noise, applied to a closed-loop reinforcement task. In all of those models the characteristics of the noise is mixed with the model of spike emission.

The purpose of our model is to prove that such neuro-realistic implementations of very general gradient ascent principles can reproduce the emergence of global regularities, as observed in biological neural networks. We use in the following a model which shares some similarities with [7], but we put a stronger emphasis on the characteristics of the noise, in order to improve the learning process.

Several temporal scales are indeed to be considered when modeling such neuromimetic controllers. First, the typical integration time of the neurons is of the order of few milliseconds. Second, the motor commands have a duration of the order of 100 ms. In the design of an adaptive controller, this temporal mismatch must be taken into account, so that the process will be guided by signals containing some random “events” whose typical duration will be of the same order than the events taking place in the environment to be controlled. For that, we consider that the firing probability is driven by a “pink noise” term whose autocorrelation is of the order of 100 ms, so that the firing probability is overestimated (or underestimated) for periods of about 100 ms. The rewards occurring meanwhile will assess the “value” of those elementary shifts, and modify the firing probability accordingly.

3 Model

3.1 Neuron model

We consider a simple model of spiking neurons. A neuron is defined by a set of N synaptic weights $W = \{w_1, \dots, w_N\}$. The neuron synaptic input is

$$I_{syn}(t) = \sum_j w_j \varepsilon_j(t - d_j)$$

where $\varepsilon_j(t)$ is the post-synaptic potential (PSP) of the j -th neuron and d_j is the axonal delay. We use a discretized Spike Response Model approach [6] for modelling the PSP arrivals and membrane refractoriness. We take here a classical exponential PSP kernel, i.e.

$$\varepsilon_j(t) = \sum_{s < t, S_j(s)=1} \exp\left(-\frac{t-s}{\tau_m}\right)$$

where τ_m is the membrane time constant.

According to the “escape noise” principle [6], the noise source is decoupled from the spike emission process, i.e. is modelled as an independent input $I_{noise}(t)$. It may be considered as an additional signal coming from an external source (a supplementary layer of neurons for instance). The total neuronal input is then $I(t) = I_{syn}(t) + I_{noise}(t)$. The membrane potential is $V(t) = I(t) - \eta(t)$ with refractoriness

$$\eta(t) = \sum_{s < t, S(s)=1} V(s) \exp\left(-\frac{t-s}{\tau_m}\right)$$

There is no stochasticity in the spike emission mechanism: a spike is emitted at t if $V(t) \geq \theta$ and if the previous spike was emitted before $t - \tau_r$, where τ_r is the refractory period. The output of the neuron $S(t)$ is 0 (no spike) or 1 (spike).

3.2 Firing probability

The firing probability is $f = P(S(t) = 1 | I_{syn}(t))$, and is expected to depend on the noise and δt . This hypothesis holds if we estimate the firing rate on large time interval $T \gg \tau_m$, and ignore some fluctuations of the probability related to the refractoriness that follows the spike emission. The neuron firing rate is $\nu = 1/\langle \text{ISI} \rangle$ where $\langle \text{ISI} \rangle$ is the mean inter-spike interval. Then, we define the firing probability in a small temporal interval $[t, t + \delta t[$ as $f = \nu \delta t$ (where $\delta t < \tau_r = \frac{1}{\nu_{max}}$). We suppose in the following that a relationship can be established between the total synaptic input I_{syn} and the firing probability. The firing probability f can be given explicitly on the basis of an estimation of mean inter-spike interval, i.e. $f(I_{syn}) = \frac{\delta t}{\langle \text{ISI}(I_{syn}) \rangle}$ (see for instance [6, 8]). The explicit calculation being rather intricate, we provisionally use an estimation of f on the basis of a set of empirical measures of $(I_{syn}(t), S(t))$ couples.

3.3 Gradient ascent principles

A reward r can occur at time t , depending on the neuron input/output $(\mathcal{E}(t), S(t))$ where \mathcal{E} is the pre-synaptic activity. The expectation of the reward is thus supposed to rely on the parameters of the input/output mapping, i.e. on the neuron’s weights \mathcal{W} . We note $J(\mathcal{W}) = \mathbb{E}(r(t))$. A local

estimator of the gradient of $J(\mathcal{W})$ can be given on the basis of the neuron input $I_{syn}(t)$, using a likelihood ratio approach [3], namely

$$\nabla_{\mathcal{W}} J = E \left[r(t) \times \left(S(t) \frac{\nabla_{\mathcal{W}} f(I_{syn}(t))}{f(I_{syn}(t))} + (1 - S(t)) \frac{\nabla_{\mathcal{W}} (1 - f(I_{syn}(t)))}{1 - f(I_{syn}(t))} \right) \right] \quad (1)$$

This estimator can be used to modify the weights at each time step in order to increase the expectation of the reward. In the case of a temporal credit assignment problem with a delayed reward occurring at time T , a local estimate of this gradient is $\nabla_{\mathcal{W}} J \simeq r(T)Z(T)$ [2] with

$$Z(t) = \lambda Z(t - \delta t) + (1 - \lambda) \left(S(t) \frac{\nabla_{\mathcal{W}} f(I_{syn}(t))}{f(I_{syn}(t))} + (1 - S(t)) \frac{\nabla_{\mathcal{W}} (1 - f(I_{syn}(t)))}{1 - f(I_{syn}(t))} \right) \quad (2)$$

with $\lambda \in [0, 1]$. By analogy with classical reinforcement learning techniques [9], $Z(t)$ is called the eligibility trace.

3.4 Weights update

Two cases can be considered.

Dense rewards When the reward signal is present at each time step of the process, the weight update is

$$w_j(t) = w_j(t - \delta t) + \alpha r(t) \Phi(t) \times (S(t) - f(I_{syn}(t))) \times \varepsilon_j(t) \quad (3)$$

which is deduced from (1), using the learning parameter α , with factor

$$\Phi(t) = \frac{f'(I_{syn}(t))}{f(I_{syn}(t))(1 - f(I_{syn}(t)))}$$

This update rule only relies on local quantities available at the vicinity of the synapse, which makes it applicable for the modelling of biologically inspired learning processes. Its main advantage is its applicability to large networks of neurons where the same reward is sent to every neuron.

So the weight change is based on an estimate of the difference between the actual firing $S(t)$ and the firing probability according to the synaptic input $f(I_{syn}(t))$. When no noise modifies the actual firing, the response may not be improved in time since the two quantities are expected to cancel each other. The noise introduces a distortion in the firing rate, so that the expectation of the difference can be positive (resp. negative) if the random signal locally increases (resp. decreases) the actual firing rate. The reward taking place meanwhile will validate (or invalidate) this distortion and modify the weights accordingly, favouring higher (resp. lower) firing rates in case of positive reward.

This rule shares common characteristics with the classical Hebbian rule [10], and the first term of the product is typically Hebbian. The subtracted term corresponds to a slow decay occurring when no spike happen. When the noise is centered, the rule is "balanced" i.e. the expectation of weight change is zero and no weight drift is expected to happen, either for positive or negative rewards. This point is important since it simplifies the design of the reward, as no particular balance between positive and negative rewards needs to be considered.

Sparse rewards In the case of occasional rewards, the weights update rule expression is:

$$\begin{aligned} w_j(t) &= w_j(t - \delta t) + \alpha r(t) z_j(t) \\ z_j(t) &= \left(1 - \frac{\delta t}{\tau_z}\right) z_j(t - \delta t) + \frac{\delta t}{\tau_z} Z_j(t) \\ Z_j(t) &= \Phi(t) \times (S(t) - f(I_{syn}(t))) \times \varepsilon_j(t) \end{aligned} \quad (4)$$

where τ_z is the trace time constant. The larger τ_z , the larger the time interval taken into account for the gradient estimate. If $\tau_z = \delta t$, we go back to equation (3). When the rewards are rare events, the trace constants needs to be large for taking into account the full sequence of actions that have led to them. Occasional reward correspond to a situation frequently encountered in real control problems. The existence of a physiological equivalent of this trace is plausible since it only relies on quantities which are locally available at the synapse.

3.5 Self-correlated noise

The capacity to avoid local minima crucially depends on the quality of the exploration process and thus on the characteristics of the noise injected in the system.

In order to help the learning process to catch the effect of a particular shift from the expected firing rate, we use a self-correlated Gaussian noise whose leak is τ_{noise} :

$$I_{noise}(t) \sim \left(1 - \frac{\delta t}{\tau_{noise}}\right) I_{noise}(t - \delta t) + \sqrt{2 \frac{\delta t}{\tau_{noise}} - \left(\frac{\delta t}{\tau_{noise}}\right)^2} \times \mathcal{N}(0, \sigma_{noise}^2)$$

This noise model is centered, gaussian with standard deviation σ_{noise} for long durations, but remains correlated on a $\pm \tau_{noise}$ ms window. In that case, on the contrary to simple escape noise or diffusive noise models [6], the deviations from the expected firing rate lasts far longer than the membrane time constant.

3.6 Neuronal controller

A neuronal adaptive controller is a set of interconnected neurons where a subpopulation of neurons is driven by the sensory signal, and the activity of another subpopulation is used to define the motor command(s). The neurons which receive the sensory signal are primary sensory neurons and the ones that drive the motor command velocity are premotor neurons. If we consider a system composed of P populations, every population owns $N^{(p)}$ neurons, and the input of one particular neuron is now :

$$I_i^{(p)}(t) = I_{i,syn}^{(p)}(t) + I_{i,noise}^{(p)}(t)$$

with

$$I_{i,syn}^{(p)}(t) = I_{i,ext}^{(p)}(t) + \sum_{q=1}^P \sum_{j=1}^{N^{(q)}} w_{ij}^{(pq)} \varepsilon_j^{(q)}(t - d_{ij}^{(q)})$$

A reward r is sent (densely or occasionally) to every neuron. The neurons locally modify their synaptic weights in order to maximize the expectation of reward. Bartlett and Baxter have shown [3] that a combination of local optimizations drives the system toward a global improvement of the actions.

4 Application

We consider a neural network controller embedded in a virtual environment. The example taken here is a model of a visual tracking system. It has no direct biological counterpart, but owns some characteristics (foveated retina, continuous moves, simple reward) that makes him suitable in a perspective of biological modelling. The achievement of the task (target tracking) is not a challenge in itself. Our aim is to study how the system learns some regularities from the environment on the basis of simple global constraints.

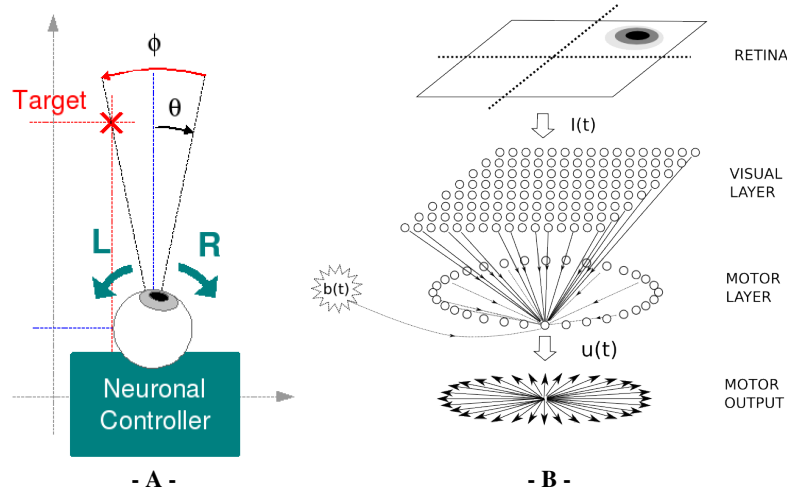


Fig. 1. - A - Environment : A rotating eyeball is tracking a punctual target (cross) : θ is the actual eye direction, ϕ is the subjective target direction (relative to the pupil's direction). **- B - Controller** : The visual layer is composed of 256 neurons sending excitatory axons toward a motor layer composed of 32 neurons. The premotor neurons inhibit each other. A slow noise $b(t)$ is added to every premotor neuron. $I(t)$ is the visual signal, and $u(t)$ is the command (see text for more details).

4.1 Environment

Consider a fixed eyeball with 2 degrees of freedom in rotation. The current direction of the pupil is given by $\vec{\theta}(t) = (\theta_x(t), \theta_y(t))$. A target is moving in the visual field and the controller must track the target by making it as close as possible to the fovea. Its direction is $\vec{\psi}(t) = (\psi_x(t), \psi_y(t))$. Its subjective direction (i.e. direction relative to the pupil) is $\vec{\phi}(t) = \vec{\psi}(t) - \vec{\theta}(t) \in]-\pi, \pi]^2$. The movement of the eye is controlled by $\vec{u}(t) = (\dot{\theta}_x(t), \dot{\theta}_y(t))$. In the examples that follow, the target is moving circularly in front of the eyeball. Its objective direction is $\vec{\psi}(t) = (\cos(\frac{2\pi t}{320}), \sin(\frac{2\pi t}{320}))$ whose period is 320 ms.

4.2 Controller

The controller is composed of a retina and 2 layers of neurons. We use a 2D foveated retina composed of $N^{(1)} = 16 \times 16 = 256$ pixels. The spatial resolution is stronger at the center of the retina than at the periphery (like in vertebrate eyes). The perceived direction is proportional to the log of the real distance to the center:

$$\vec{d}(t) = \text{sign}(\vec{\phi}(t)) \times \log_{10}\left(1 + 9 \frac{|\vec{\phi}(t)|}{\pi}\right) \in]-1, 1]^2$$

The first layer consists of 256 neurons which are directly stimulated by the external signal. Consider that each cell $i \in \{1, \dots, N^{(1)}\}$ of the first layer responds to a preferred input direction \vec{d}_i . Taking $M = \sqrt{N^{(1)}}$, $k = \lfloor \frac{i-1}{M} \rfloor + 1$, $\ell = i - (k-1)M$, so that $(k, \ell) \in \{1, \dots, M\}^2$, we fix $\vec{d}_i = (\frac{2k-(M+1)}{M}, \frac{2\ell-(M+1)}{M})$. Then

$$I_{i,ext}^{(1)}(t) = 2 \left\langle \cos((\vec{d}(t) - \vec{d}_i) \times \pi) \mid H(\cos((\vec{d}(t) - \vec{d}_i) \times \pi)) \right\rangle$$

where $\langle \cdot | \cdot \rangle$ is the inner product and H is the Heaviside function.

The second layer consists of $N^{(2)} = 32$ neurons. It sends a velocity command $\vec{u}(t) = (\dot{\theta}_x, \dot{\theta}_y)$ to the eye effectors. Every neuron of the second layer is associated with a particular angular direction, i.e. $\theta_i = i \times \frac{\pi}{N^{(2)}}$ rad. The output of the controller is defined according to the spiking activity of the population, i.e.

$$\vec{u}_i(t) = \varepsilon_i^{(2)}(t) \times (\sin(\theta_i), \cos(\theta_i))$$

and

$$\vec{u}(t) = \frac{1}{\sqrt{N^{(2)}}} \sum_{i=1}^{N^{(2)}} \vec{u}_i(t)$$

which is normalized in $\sqrt{N^{(2)}}$ in order to maintain the variance of the command independent of $N^{(2)}$.

The neurons of the first layer send excitatory links to the second layer. The neurons of the second layer send lateral inhibitory links so that the global input of the premotor neurons is balanced. The connectivity is initially random. Considering the synaptic sums $W_i^{(pq)} = \sum_{j=1}^{N^{(q)}} w_{ij}^{(pq)}$, the weights are randomly set so that $E(W_i^{(pq)}) = \mu^{(pq)}$ and $\text{var}(W_i^{(pq)}) = \sigma^{(pq)^2}$. Here we fix $\mu^{(21)} = 2.5$, $\sigma^{(21)} = 0.1$, $\mu^{(22)} = -2.5$, $\sigma^{(22)} = 0.25$.

A slow noise is added to the input of the premotor neurons. If not precised, we use in the following $\tau_{noise} = 100$ ms and $\sigma_{noise} = 0.35$.

4.3 Reward

We expect the system to learn to compensate the visual error by making a movement in the observed direction of the target, i.e. to learn the appropriate feedback homeostatic controller. The reward being dimensionless, the system is only informed how good or bad its current command is at following the target, but is not informed of the precise direction to choose to improve it. Its improvement will rely on "trials and errors".

Considering the simpler the better, the reward is based on the subjective direction from the fovea $\vec{d}(t)$:

$$r(t) = \frac{1}{2} - \|\vec{d}(t)\|$$

The rewards are expected to be negative most of the time at the beginning of the learning process, since the probability to have the target near the center of the fovea is initially low.

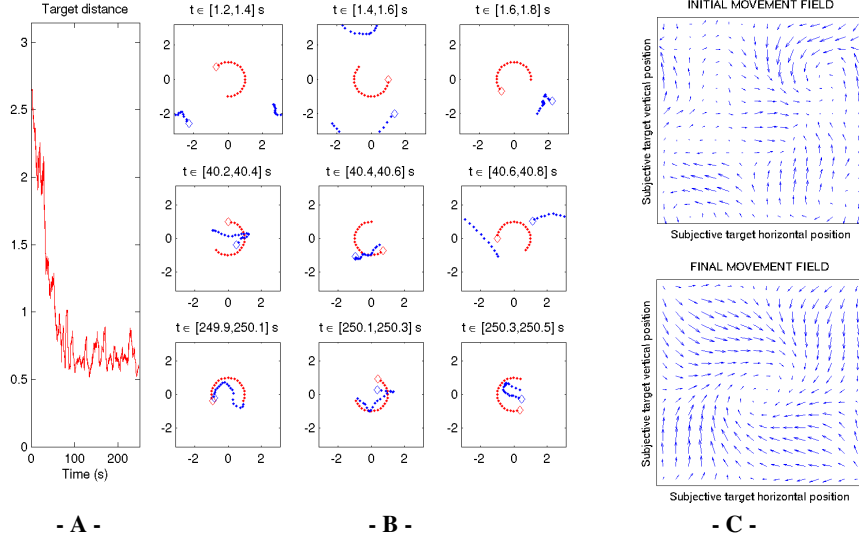


Fig. 2. - A - Evolution of the mean distance ($\|\psi - \theta\|$) on a 1 s interval, during the 250 s of the learning process - **B -** Samples from the controller and target dynamics, on 200 ms intervals, with 10 ms resolution. The red dots give the target direction (the diamond gives the final direction in the interval). The blue dots give the eye direction. - **C -** (top) Initial average motor output for a target appearing at the considered subjective direction, (bottom) final average motor output (see text for more details).

4.4 Learning

The learning process is applied for 250 s on both excitatory and inhibitory synapses, a reward is sent every τ_z ms (in figure 2, $\tau_z = 100$ ms). The learning parameter is $\alpha = 0.03/N^{(1)}$. The neural controller initially sends erratic commands to the effectors, which contributes to an approximately uniform exploration of various eye positions, while the target rapidly moves on the retina. The rewards being sent every 100 ms, a progressive improvement of the command is observable after 20-40 s where the target appears more frequently near the center of the retina (as the expectation of the reward increases). Then the behavior of the controller progressively stabilizes around 50-100 s and the mean distance remains stable (0.5-0.7 rad) for the rest of the simulation. The eye is now tracking the target, approximately reproducing the circular movement observed.

If we look at the premotor neurons responses individually at the end of the learning process, no specialization is clearly visible. Individual premotor neurons do not systematically respond to targets appearing on the opposite direction (not shown). If we now look at the output command (i.e. the mean of the 32 premotor neurons response), then a clear specialization is visible at the end of learning. In figure 2-C-, we give the mean motor response according to 256 typical target directions, at the beginning and at the end of the learning process. The initial movement field displays no particular organization, as it only relies on the randomness of the initial connections. On the contrary, the movements produced at the end of learning are systematically oriented toward the center of the retina, the command being weaker when the target is near the center, almost proportionally to the observed “error”.

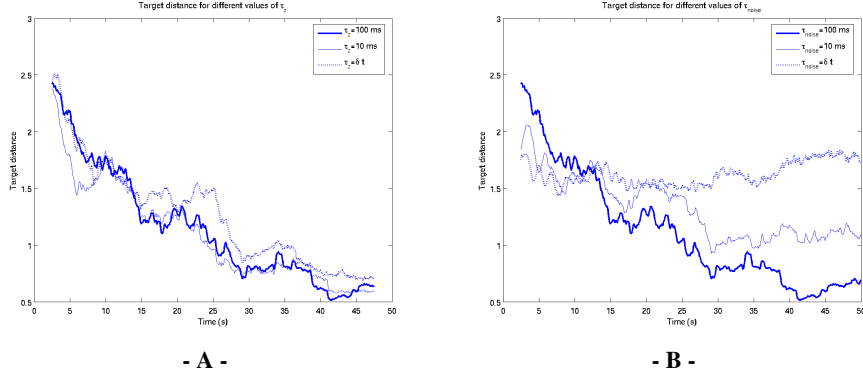


Fig. 3. - A - Mean distance during the first 50 s of the learning process, for $\tau_z = 100$ ms (bold line), $\tau_z = 10$ ms (thin line) and $\tau_z = \delta t$ (dotted line), with $\tau_{noise} = 100$ ms - **B -** Idem, for $\tau_{noise} = 100$ ms (bold line), $\tau_{noise} = 10$ ms (thin line) and $\tau_{noise} = \delta t$, with $\tau_z = 100$ ms.

We consider now the two parameters : τ_z and τ_{noise} . The value of τ_z controls both the length of the trace and the frequency of the rewards, the rewards appearing every τ_z ms and the trace taking into account the most recent firings in the same interval. We consider different values of τ_z from continuous rewards ($\tau_z = \delta t$) to sparse rewards ($\tau_z = 100$ ms). We observe in figure 3-A- that changing τ_z has no particular effect on the learning performance. The frequency of rewards is not found to significantly influence the learning process: the performance is about the same for rewards frequency within a [0.5...100] ms range.

The value of τ_{noise} controls the “slowness” of the noise (its autocorrelation) without modifying its variance. The case of $\tau_{noise} = \tau_m = 10$ ms roughly corresponds to a diffusive noise filtered by the neuron membrane, while the value $\tau_{noise} = \delta t$ corresponds to the simple gaussian escape noise model [6]. Changing the characteristics of the noise appears on the contrary to strongly impair the learning performance. With a simple uncorrelated gaussian noise ($\tau_{noise} = \delta t$), the controller is simply unable to learn the task. The “slowness” of the noise thus appears to be an essential feature for the achievement of learning. This effect can be explained by the closer correspondence between the noise and environment time constants. In particular, for the environment we consider, the autocorrelation window of the current error signal $\|\vec{\psi}(t) - \vec{\theta}(t)\|$ is of the order of 100 ms (not shown), which corresponds to the value of τ_{noise} giving the best learning performance.

5 Conclusions

We have presented a new learning setup that illustrates the relevance of a policy-gradient based RL approach for modelling realistic and convergent synaptic plasticity rules. We show here in particular that the shape of the noise is of crucial importance for the algorithm convergence. The time constants shared by the noise and the environment help the controller to capture the relevant shifts from the expected firing rates, and modify the weights accordingly, even when the reward information is sparse. During the learning process, the premotor neurons modify their response in order to favour better rewards, without getting finely specialized. It is only by looking at the population that a coherent response pattern appears, as a combination of individual outputs.

The realism of our approach in modelling natural learning depends on the existence of internal random signals having comparable self-correlation characteristics. Our modelling hypothesis should be considered regarding the known characteristics of spiking activity [11]. This study more generally points out the interest of temporally (and spatially) structured noises for orienting the “exploration” of the environment in artificial control systems.

Acknowledgements

The author thanks the INRIA Lille-Nord europe for 1-year delegation in the SEQUEL team. This work is supported by the french ANR MAPS (ANR-07-BLAN-0335-02).

References

1. C.J. Watkins and P. Dayan. Q-learning. *Machine learning*, 8:279–292, 1992.
2. R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.
3. Peter L. Bartlett and Jonathan Baxter. Hebbian synaptic modifications in spiking neurons that learn. Technical report, Research School of Information Sciences and Engineering, Australian National University, 1999.
4. H. Sebastian Seung. Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. *Neuron*, 40:1063–1073, december 2003.
5. Dorit Baras and Ron Meir. Reinforcement learning, spike time dependent plasticity and the bcm rule. *Neural Computation*, 19(8):2245–2279, 2007.
6. W. Gerstner and W. Kistler. *Spiking Neuron Models. Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002.
7. Razvan V. Florian. A reinforcement learning algorithm for spiking neural networks. In *Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC’05)*, pages 299,306, 2005.
8. Hédi Soula, Guillaume Beslon, and Olivier Mazet. Spontaneous dynamics of asymmetric random recurrent spiking neural networks. *Neural Computation*, 18:60–79, 2006.
9. R.S. Sutton. Learning to predict by the method of temporal differences. *Machine learning*, 3:9–44, 1988.
10. D. Hebb. *The Organization of behavior*. Wiley, New York, 1949.
11. W. Softky and C. Koch. The highly irregular firing of cortical cells is inconsistent with temporal integration of random epsps. *J. of Neuroscience*, 13(1):334–450, 1993.