Eur. Phys. J. Special Topics **142**, 123–161 (2007)
© EDP Sciences, Springer-Verlag 2007
DOI: 10.1140/epjst/e2007-00060-8

THE EUROPEAN
PHYSICAL JOURNAL
SPECIAL TOPICS

Learning and control with large dynamic neural networks

E. Daucé[12]

[1] Movement and Perception, UMR 6152,
[2] École Centrale de Marseille, France
E-mail: edauce@ec-marseille.fr

Abstract. This paper is a presentation of neuronal control systems in the terms of the dynamical systems theory, where (1) the controller and its surrounding environment are seen as two co-dependent controlled dynamical systems (2) the behavioral transitions that take place under adaptation processes are analyzed in terms of phase-transitions. We present in the second section a generic method for the construction of multi-population random recurrent neural networks. The third section gives an overview of the various phase transitions that take place under an external forcing signal, or under internal parametric changes. The section 4 presents some applications in the domain of sequence identification and active perception modeling. The section 5 presents some applications in the domain of closed-loop control systems and reinforcement learning.

Introduction

The previous papers have mainly described the properties of the intrinsic interactions of big sets of neurons. Those neuronal sets have been described in the terms of the dynamical systems theory, following the pioneer work of Grossberg [1], Amari [2] and Hopfield [3].

Most of the recurrent neural networks models fall in the category of dissipative systems[1]. The qualitative properties of dissipative dynamical systems can be summarized the following way:

- The state space of a dynamical system is divided into several *basins of attraction* into which every trajectory converges toward a unique *attractor*.
- The nature and structure of these attractors can vary strongly from one system to another, and from one attraction basin to another. The most simple attractors are fixed points. The most complex attractors that we shall consider are called “strange attractors”. They are associated with chaotic dynamics.
- The shape and nature of the attraction basins can vary under parametric changes, which are often supposed adiabatic (i.e. slow according to the state update). Phase transitions occur when the topological characteristics of trajectories undergo a sudden change.

A dynamical system can model phenomena taking place at various temporal scales. First, it determines a short term causality between the successive states of the system. Second, the evolving topology of its basins of attraction under parametric changes determines the long term changes in the organization of the dynamics.

So, the dynamical description gives in first place a set of tools in order to analyze the versatile properties of real neuronal assemblies. It has long been suggested, for instance, that an attractor basin encodes a *memory* of a particular item [3], where the “recall” (or recognition)

[1] See the appendix of paper 1 [79] for the definition and properties of dissipative systems.

corresponds to a “resonance” between a particular sensory or sensori-motor configuration and a recurrent neural network [4,5] (see also paper 1 [79], part 6.6).

In the field of computer sciences and applied mathematics, recurrent neural networks are proved to be powerful *auto-associators*. In other terms, they have been proved to robustly reconstruct some given prototypical patterns out of piecewise or noisy input patterns. The counterpart is a rather low capacity which is found to linearly increase with the size N (i.e. to increase as the square root of the number of parameters). Recurrent memories are however much more powerful than mere fixed-point auto-associators. They are indeed found to display non-stationary and possibly complex self-sustained dynamics as soon as the classical constraint of weights symmetry [6] is released. This allows in particular to store and retrieve multiple spatio-temporal patterns in the form of limit cycles of the dynamics [7–9], which is, to our opinion, the prominent property of recurrent memories.

In a biological context, this property is ubiquous throughout the whole nervous system. It can be noticed for instance that primitive and vital controls such as mastication and locomotion rely on small and versatile recurrent sets of neurons called Central Pattern Generators (CPG) [10,11]. The recurrent structure of hippocampal CA3 network is suggested to take part in the recall of sequences of events [12,13]. More broadly, the connections between cortical areas are most of the time found to be reciprocal, and large scale mechanisms of synchrony locking have been suggested to take part in perception processes [14].

So, if recurrent dynamical neural networks clearly appear as prototypical models of short and long term memory, they are also suggested to give clues in some aspects of perception and action production. As they tend to produce multiple self-sustained spatio-temporal patterns of activation, they seem indeed particularly suitable for the production and stabilization of sensori-motor patterns [15]. This point is the one we will try to enhance in the present paper.

The first section gives an overview of the available dynamical systems tools and methods in motor control. In particular, we show how the neuronal models of agents (or robotic devices) take advantage in being embedded in more generic models of interaction processes. The second section gives some insights into neuronal modeling and network design. We show how to build modular structures with random weights distributions out of minimal parameter sets. The choice of the spatial and temporal resolution thus fixes the effective network realization. The third section presents some aspects of the bifurcations and transitions taking place in various models of recurrent neural networks. We distinguish in particular two families of transitions, the first being input-driven transitions, the second being parameter-driven transitions (in the particular case of Hebbian learning). The fourth section gives an example of on-line spatio-temporal sequence retrieval. The retrieval property relies on the resonance between a sensory layer and a recurrent memory. Another example taking place in the framework of robotic control is given. The fifth section gives a prototypical example of motor control achievement with the use of simple Hebbian reinforcements in a recurrent network of binary neurons.

1 Interacting systems and learning

Before going further in neuronal modeling, we draw in this section some basic formal settings for the definition of a global model of interaction.

1.1 Modeling the two sides of an interaction process

A control system is a *model* of interactions between a model of controller (or agent) and a model of the environment. An agent (man, animal or robot) owns a nervous system or any kind of fast internal process which eventually sends some command signals toward effectors (muscles, arms, wheels . . .). The body of the agent is at the same time immersed in an environment with its own constraints and dynamics. The agent’s body owns sensors which translate some of the external state variables into various signals which take part in the (fast) internal process. The variety of the sensors implies that the external world is perceived through different sensory

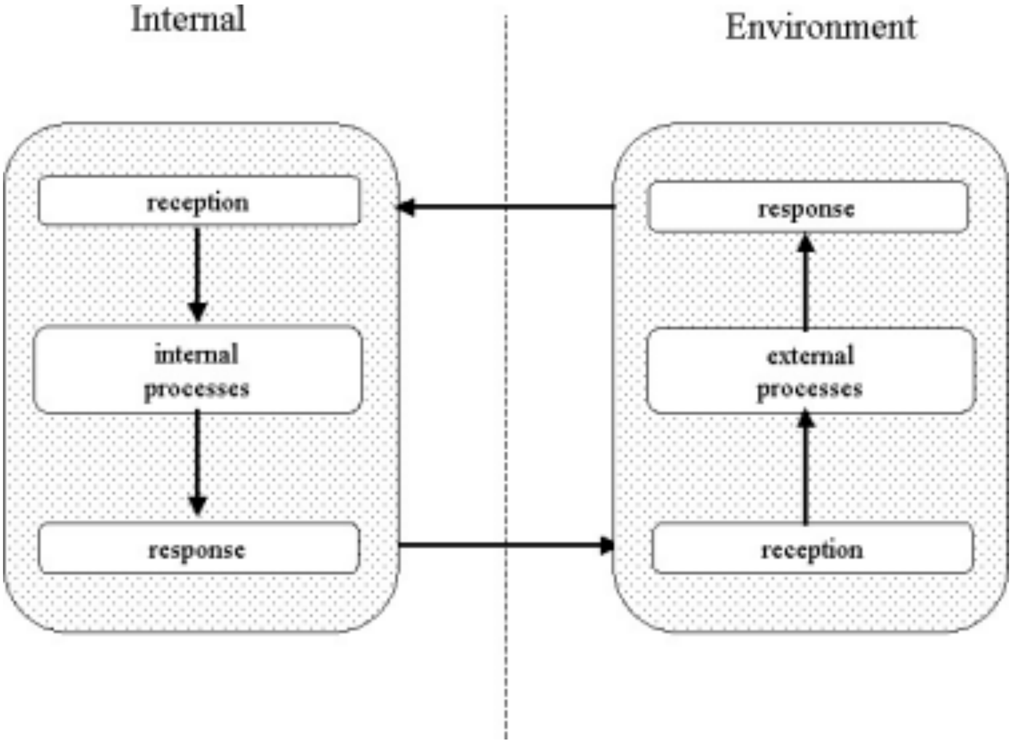

Fig. 1. Agent/Environment interaction.

modes (touch, smell, vision . . .). The environment evolves under the actions of the agent, and those actions are updated according to the sensory flow.

In the most general framework (see figure 1), we have to consider at the same time two domains of description. Those domains are separated by the agent's body, where the skin and sensory organs draw the "frontier" between the "inside" and the "outside".

- The first domain of description corresponds to the physical world, including the agent's body/actuators and everything surrounding the agent.
- The second domain of description is the agent internal space, i.e. the agent's body seen "from the inside". It can be possibly modeled, for instance, as a dynamic neural network, or a set of expert systems, or even an electric circuit. This system is under the influence of various sensory inputs.

The agent's and environment evolutions are co-dependent, i.e. belong to the same process, whose evolution originates from each side influence. A specific dynamical system can be associated to the two domains, each system giving a partial picture of the global ongoing interaction[2]. The outer system includes a series of processes taking place in the physical world, *including* the agent actions and movements. The inner system includes a series of processes taking place inside the agent's "brain".

The outer system does not include the agent's "purposes". Reciprocally, the inner system does not include what is not perceivable by the agent. Those unpredictable parts, taking place on one side or the other, will be called "hidden variables" or "hidden processes".

Such systems can be described either in a determinisic or, more commonly, in a stochastic fashion.

[2] In that framework, we thus have "two systems in one", and those two subsystems are in a mirror relationship, so that the "perceptions" of one system are the "actions" of the other one, and reciprocally, even if it is of course a bit unusual to think of the environment as "perceiving" the agent, and "acting on" or controlling the agent.

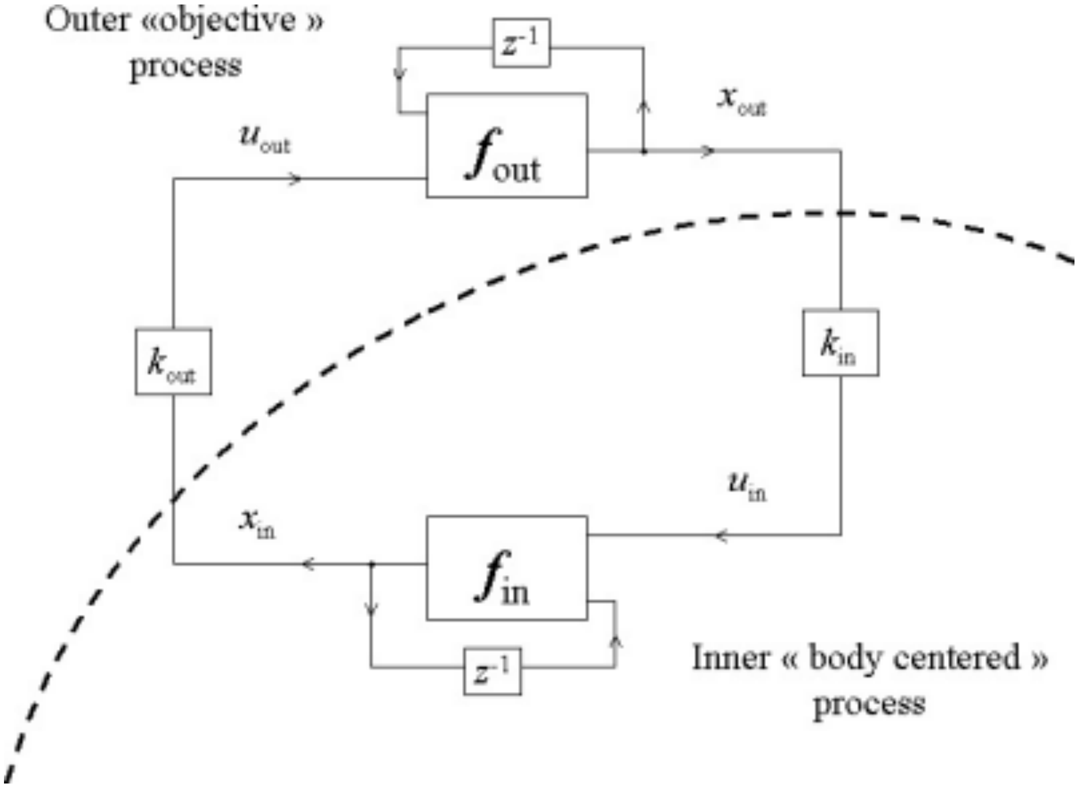

Fig. 2. Functional diagram of an interaction system (the dashed line represents the frontier between the internal processes and the external processes, i.e. the "skin").

Deterministic interaction models

For the seek of simplicity, we start with a deterministic presentation of the considered interaction system. Our formal presentation remains at a schematic level: we implicitly suppose that our interaction system is fully determined. The interaction system is thus a single dynamical system which has been split, for clarity reasons, into an environment and an agent, for which we suppose we have a precise state description. Without loss of generality, we use here a discrete time description. An *interaction system* can thus be described by the set of deterministic equations:

$$\begin{cases} u_{\text{out}}(t) = k_{\text{out}}(x_{\text{in}}(t)) \\ x_{\text{in}}(t) = f_{\text{in}}\left(x_{\text{in}}(t-1), u_{\text{in}}(t-1)\right) \\ u_{\text{in}}(t) = k_{\text{in}}(x_{\text{out}}(t)) \\ x_{\text{out}}(t) = f_{\text{out}}\left(x_{\text{out}}(t-1), u_{\text{out}}(t-1)\right). \end{cases} \tag{1}$$

Where X_{in} is the internal state space, U_{in} is the internal input state space, $f_{\text{in}} : X_{\text{in}} \times U_{\text{in}} \to X_{\text{in}}$ is the internal transition function, $x_{\text{in}} \in X_{\text{in}}$ is the internal state, $u_{\text{in}} \in U_{\text{in}}$ is the internal command (the observation vector), X_{out} is the external state space, U_{out} is the external input state space, $f_{\text{out}} : X_{\text{out}} \times U_{\text{out}} \to X_{\text{out}}$ is the external transition function, $x_{\text{out}} \in X_{\text{out}}$ is the external state, $u_{\text{out}} \in U_{\text{out}}$ is the external command,

- The mapping $k_{\text{out}} : X_{\text{in}} \to U_{\text{out}}$ represents the transformation of the agent's state space to the commands space, i.e. the various forces which activate the agent's body. The outer process is thus dependent on the internal state, through the agent's *movements* $u_{\text{out}}(t)$.
- Conversely, the mapping $k_{\text{in}} : X_{\text{out}} \to U_{\text{in}}$ represents a transformation that maps the external space to the agent body-centered space, basically corresponding to the signal sent to the agent by its various sensors. The agent is dependent on the external state, through its *observations* $u_{\text{in}}(t)$.

A simple example

We illustrate the framework of interacting systems (1) with an idealized situation of a Khepera-type robotic agent interacting with a flat environment. This example is inspired by [16]. This agent is represented on figure 3.

The interaction model we take in consideration uses rough simplifications of both robot and environment models. Let us considerate an agent A living in a 2 dimensional space, which also contains a target S.

- The agent is externally described by 3 degrees of freedom (DOF), with two cartesian coordinates x_A and y_A, and its current orientation ϕ. Its input is here composed by a wheel command v acting on the agent angular velocity. The target has 2 DOF, and can be described by its two coordinates x_S and y_S. The outer state space X_out is consequently composed of 5 state variables so that $\mathbf{x}_\mathrm{out} = (x_\mathrm{A}, y_\mathrm{A}, \phi, x_\mathrm{S}, y_\mathrm{S})$, and the command state space is $\mathbf{u}_\mathrm{out} = v$.
- The inner space is composed by a single variable ψ (which corresponds to the current inner estimate of the target position). The current observation is ψ^* (visual signal/target perceived position). The inner state space X_in is composed of 1 state variable so that $\mathbf{x}_\mathrm{in} = \psi$, and the observation space is mono-dimensional so that $\mathbf{u}_\mathrm{in} = \psi^*$.

What we usually consider as the “objective” or physical space constitutes the external space. It corresponds to the Cartesian space, and needs an arbitrary reference to be defined, which is the origin of x and y axes. On the contrary, the inner space is body-centered, so that its “origin” is defined through the configuration of the agent’s body. In this case, the origin of the visual field is simply the center of the field, according to the visual sensors position.

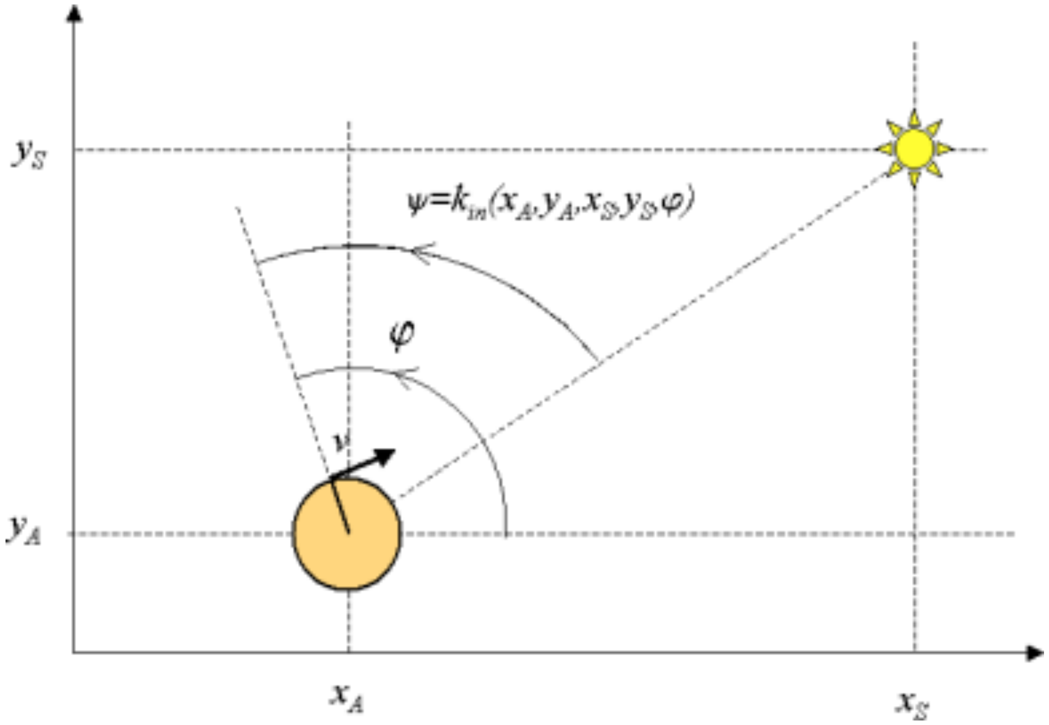

Fig. 3. Model of a Khepera-type robotic agent in a simplified environment composed of one target towards which the agent tends to orientate. This situation is externally defined by 5 state variables: $x_\mathrm{A}, y_\mathrm{A}, x_\mathrm{S}, y_\mathrm{S}, \phi$. Variable ψ represents the relative target position (i.e. “observation”), according to the five previous variables. Variable v is a wheel command allowing the agent to rotate.

Two mappings can now be defined. They constitute the interfaces from one space to the other. The first mapping is $k_{\mathrm{in}} : X_{\mathrm{out}} \to U_{\mathrm{in}}$ with

$$\psi^*(t) = k_{\mathrm{in}}(\mathbf{x}_{\mathrm{out}}) = \phi - \arctan\left(\frac{y_{\mathrm{S}} - y_{\mathrm{A}}}{x_{\mathrm{S}} - x_{\mathrm{A}}}\right).$$

This mapping is an idealization of the transformation done by the sensors. The second mapping $k_{\mathrm{out}} : X_{\mathrm{in}} \to U_{\mathrm{out}}$ is

$$s(t) = k_{\mathrm{out}}(\mathbf{x}_{\mathrm{in}}) = -\sin(\psi(t)) \tag{2}$$

which translates the inner state in a certain command operating in the direction of the current estimate of the target position, according to an homeostatic regulation principle.

At last, we can describe the two dynamics:

- The external dynamics is here very simple. This example only describes the process by which an agent orientates toward a target. The environment dynamics is here reduced to the agent's body which has only one degree of freedom: ϕ , the absolute angular position. The environment dynamics is thus updated this way:

  $$\tau_{\mathrm{out}}\frac{d\phi}{dt} = s(t)$$

  where τ_{out} is the external time constant.
- The inner dynamics gives the current estimate of the target position. Once again, we choose for this particular example a simple update:

  $$\tau_{\mathrm{in}}\frac{d\psi}{dt} = \psi^*(t) - \psi(t)$$

  whose fixed point is $\psi(t)$ and parameter τ_{in} is the internal time constant. In presence of several targets, say ψ_1 and ψ_2, a more elaborate transition function could be imagined, in order to allow a selection of one target among the others:

  $$\tau_{\mathrm{in}}\frac{d\psi}{dt} = (\psi_1^* - \psi)\exp\left(-\frac{(\psi_1^* - \psi)^2}{2}\right) + (\psi_2^* - \psi)\exp\left(-\frac{(\psi_2^* - \psi)^2}{2}\right)$$

  see [16]. In a more realistic stochastic model (see next paragraph), a Kalman filter could also be used, giving estimates of ψ_1^* and ψ_2^* instead of the straightforward target perception.

The external relaxation parameter τ_{out} should be adapted in a realistic model to the response characteristics/transfer function of the robot's wheels. The internal relaxation parameter τ_{in} is supposed to allow for a fast adaptation to the input change, for the response of the system to reliably represent the target position. It is thus implicitly supposed that $\tau_{\mathrm{in}} \ll \tau_{\mathrm{out}}$, which corresponds to the idea that the internal (neuronal) process is a fast process, and the external (body/environment) process is a slow one.

Stochastic interaction models

In a more realistic framework, one has to consider various random factors which represent the non-deterministic/unknown part of our model. In our formal setting, this means that our system should be updated according to several random processes: $v_{\mathrm{in}}(t)$, $w_{\mathrm{in}}(t)$, $v_{\mathrm{out}}(t)$, and $w_{\mathrm{out}}(t)$, for which the probability law may be known, so that

$$\begin{cases} u_{\mathrm{out}}(t) = k_{\mathrm{out}}(x_{\mathrm{in}}(t), v_{\mathrm{out}}(t)) \\ x_{\mathrm{in}}(t) = f_{\mathrm{in}}(x_{\mathrm{in}}(t-1), u_{\mathrm{in}}(t-1), w_{\mathrm{in}}(t)) \\ u_{\mathrm{in}}(t) = k_{\mathrm{in}}(x_{\mathrm{out}}(t), v_{\mathrm{in}}(t)) \\ x_{\mathrm{out}}(t) = f_{\mathrm{out}}(x_{\mathrm{out}}(t-1), u_{\mathrm{out}}(t-1), w_{\mathrm{out}}(t)) \end{cases}$$

where v_{in} and v_{out} correspond to *observation noises* and w_{in} and w_{out} correspond to *process noises.*

In real world problems, it is often difficult to obtain the global model, so that the complexity of the domain description is often reduced to one side or the other. In order to design proper controllers, assumptions are often made on the stationarity of the environment for instance (Partially Observed Markov Decision Processes framework). In the case of non-stationnary environments, an estimation of the external states can be processed out of adaptive filters, under the assumption of a full description of the measure process (Kalman filters and extended Kalman filters). For more details, the reader will refer to [17,18].

1.2 Knowledge acquisition and movement production

The mixed process of equation (1) is moreover supposed to evolve: assuming that (1) is dissipative, the basins of attraction may undergo structural transformations. Those structural changes correspond to a slow construction process through which the agent gets new skills (and new "knowledge") (the environment may also undergo structural changes), and are generally called *learning processes.*

In the vocabulary of dynamical systems, skill acquisition is assimilable to *uncertainty reduction*, i.e. to an increase of the predictability of the system's trajectories[3]. Up to this point, two main approaches can be distinguished:

- For the representationalist (or realistic) school [19], skill acquisition means to reduce the uncertainty on "what is *really* going on out there", i.e. to manage to produce motor patterns which are the most relevant in given sensory situations. This approach preaches for the use of *internal models* of the environment.
- For the non-representationalist (or constructivist) school [20–22], skill acquisition means to reduce the uncertainty on "what's coming next", i.e to globally produce more predictable interaction patterns. Under this approach, the sensory and internal processes are subordinate to the production of regular and persistent couplings between the body and some affordant partners from the environment.

The balance between the predictable part and the unpredictable part of action production is also found within the learning process itself. In the process of procedural knowledge and skill acquisition, a subtle balance between *exploration* (progressism) and *exploitation* (conservatism) has to be maintained. This question relates to the classical stability/plasticity trade-off [4]. The process has to switch opportunistically between exploration, when the action mainly comes from the agent's internal dynamics, and exploitation, when the actions are mainly driven by the environment.

1.2.1 Supervised methods

Using a supervised method means to give the agent a model for its actions. The aim is thus to reduce the error between the spontaneous motor response and the desired one. This question arises, of course, when one knows the suitable control, but doesn't know the method or parameters for designing the desired controller. We must thus have a "prescriptive" model of this kind:

$$\mathbf{u}^*(t+1) = f^*(\mathbf{u}(t))$$

which represents the desired future perception $\mathbf{u}^*(t+1)$ following the current perception $\mathbf{u}(t)$.

In this framework, we have to consider an internal identification process (i.e. internal model) which is supposed to anticipate the future perception:

$$\tilde{\mathbf{u}}(t) = f_{\mathrm{I}}(\mathbf{x}(t-1), \quad \mathbf{u}(t-1))$$

where $\mathbf{x}$ represents the last "action". Such an identifier f_{I} can be determined using classical non-linear regression methods, for instance backpropagation methods.

[3] The process of variability reduction possibly takes place together with the reduction of a cost function, which is out of the scope of the dynamical systems theory.

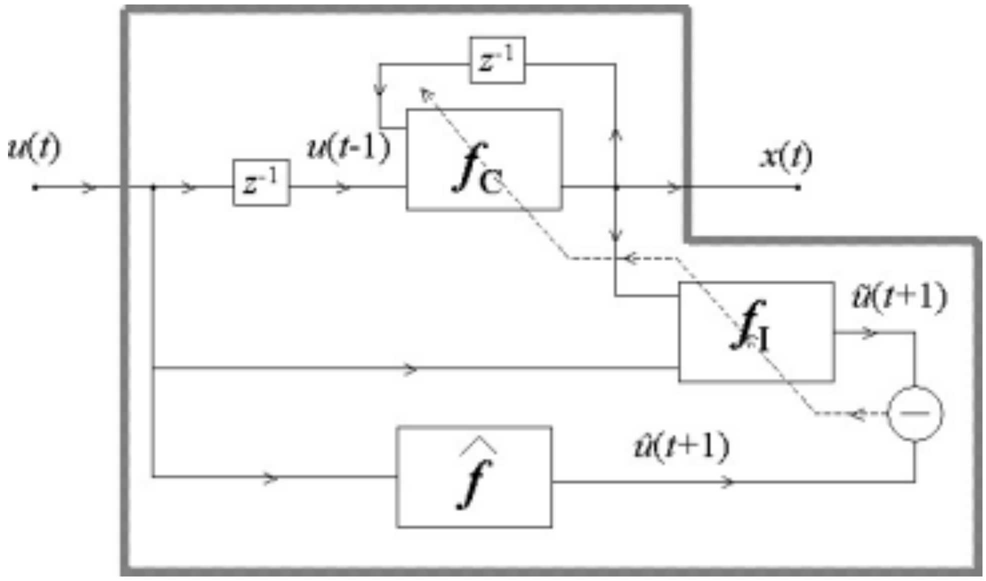

Fig. 4. Functional diagram (solid lines) and backpropagation path (dashed line) for the supervised motor learning framework.

Two methods can be distinguished for parametric learning: *on-line* learning and *batch* learning. On-line learning means a continuous change of the system parameters according to every incoming signals. Batch learning means to dissociate perception-action processes from adaptation processes, so that parameter changes do not take effect immediately, but arise by steps, at discrete moments.

The main problem is to determine a certain controller f_{C}:

$$\mathbf{x}(t) = f_{\mathrm{C}}(\mathbf{x}(t-1), \quad \mathbf{u}(t-1))$$

such that $\mathbf{x}$ will correspond to the proper suitable action.

The global update equation of the controller is:

$$\begin{cases} \mathbf{x}(t) = f_{\mathrm{C}}(\mathbf{x}(t-1), \quad \mathbf{u}(t-1)) \\ \tilde{\mathbf{u}}(t) = f_{\mathrm{I}}(\mathbf{u}(t-1)) \\ \mathbf{u}^*(t) = f^*(\mathbf{u}(t-1)). \end{cases}$$

Its functional diagram is given on figure 4.

Determining f_{C} typically corresponds to an "inverse modeling" problem: knowing the suitable perception $\mathbf{u}^*(t)$, one need to state the suitable action $\mathbf{x}^*(t-1)$ which could be the cause of this perception. This estimate needs f_{I} to be invertible. Even if f_{I} is invertible, there is a great risk that it may be badly conditioned, so that a strong error on action selection may result from a small identification error. Backpropagation offers in this case a suitable method for the design of such supervised controller (see figure 4). Knowing f^*, *after* f_{I} has been identified, the error term $\mathbf{e}(t+1) = \mathbf{u}^*(t+1) - \tilde{\mathbf{u}}(t+1)$ is backpropagated in order to obtain an error term $\mathbf{e}_x(t)$, *without changing the weights inside identifier* f_I. This first backpropagation corresponds to the use of the inverse model, i.e. $\mathbf{x}^*(t) \simeq \mathbf{x}(t) + \mathbf{e}_x(t)$. This error term is then backpropagated into module f_{C} for the regression to converge on the desired controller.

The error $\mathbf{e}_x(t)$ represents the control correction to be done. It is actually determined at time $t+1$, after a measure on $\mathbf{u}(t)$. If the correction is made on-line, f_{C} may be modified so that $\mathbf{x}(t+1)$ will include a correction of the previous error (but the system is not aware of the *current* error).

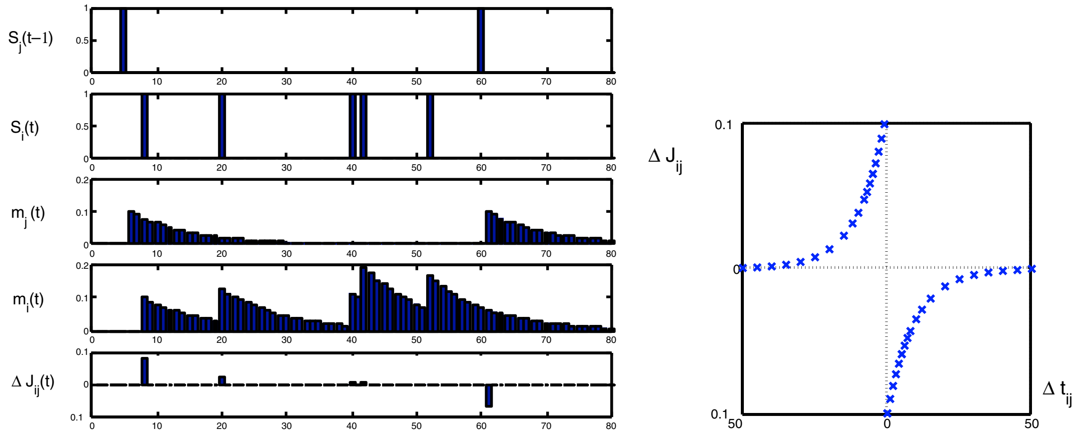

Fig. 5. The left figure gives the time evolution of indicators m_i and m_j according to a random pre-synaptic and post-synaptic spike train. The resulting weight reinforcement ΔJ_{ij} is on the lower plot. The right figure gives the value of the weight reinforcement according to the temporal difference between pre-synaptic arrival and post-synaptic firing, i.e. $\Delta t_{ij} = \sum_{t=1}^{100} t(\delta(S_j(t-\tau_{ij}) - \delta(S_i(t))))$ in the particular case $\tau_{ij} = 1$.

1.2.2 Reinforcement methods

Reinforcement methods must be used when the agent designer doesn't even know exactly what his agent is supposed to do in the interaction process. The reinforcement process thus relies on a series of rewards and punishments, that occasionally occur during the interaction process. This method explicitly needs:

- an *exploration* process, for the agent to generate its own actions;
- a *selection process*, for the agent to maximize the rewards.

Given a certain agent

$$\mathbf{x}_{\text{in}}(t) = f(\mathbf{x}_{\text{in}}(t-1), \quad \mathbf{u}_{\text{in}}(t-1))$$

the problems is thus to find a function f^* which optimizes the rewards.

The classical reinforcement learning techniques are derivatives of the dynamics programming approach, and divide into TD-learning methods [23,24] and Q-learning methods [25].

A reinforcement learning problem is classically defined by a space with its states $\mathbf{x}_{\text{out}}$, with the observation $\mathbf{u}_{\text{in}} = \mathbf{x}_{\text{out}}$ in the fully observable case. The external states are valuated according to a value function $x_{\text{in}}(t) = V(\mathbf{u}_{\text{in}}(t))$. This value function is an estimate of the sum of the future rewards $E(\sum_{k=t}^{\infty} r(k))$. In the linear case[4]: $V(\mathbf{u}_{\text{in}}) = \mathbf{J}\mathbf{u}_{\text{in}}$, the learning of the value function relies on the temporal differences rule TD(0) [12], i.e.

$$\Delta\mathbf{J}(t) = \alpha(r(t) + x_{\text{in}}(t) - x_{\text{in}}(t-1)) \times \mathbf{u}_{\text{in}}(t-1) \tag{3}$$

where $r(t)$ is the current reward and $\mathbf{u}_{\text{in}}(t-1)$ is the *last* input. After several experiments, this rule converges toward a prediction of the sum of the next expected rewards according to the current perception $\mathbf{u}_{\text{in}}$. In the TD(λ) scheme, a *trace* of the previous inputs is memorized, i.e

$$\mathcal{T}_{\text{in}}(t) = \mathbf{u}_{\text{in}}(t) + \lambda\mathcal{T}_{\text{in}}(t-1) \tag{4}$$

and the update rule is now

$$\Delta\mathbf{J}(t) = \alpha(r(t) + x_{\text{in}}(t) - x_{\text{in}}(t-1)) \times \mathcal{T}_{\text{in}}(t-1) \tag{5}$$

[4] In a more general case, $V(.)$ can be considered as a differentiable mapping, for instance a multi-layer perceptron, with weight parameters $\mathbf{J}$. In the case a discount factor γ takes place, the weight update is of the form $\Delta\mathbf{J}(t) = \alpha(r(t) + x_{\text{in}}(t) - \gamma x_{\text{in}}(t-1))\nabla_J(x_{\text{in}}(t)) \times \mathbf{u}_{\text{in}}(t-1)$. The linear formula has been given for simplicity, but the reader can easily generalize to the non-linear case and to the use of classical backpropagation learning techniques.

Under the classical TD approach [23], an "actor" process is responsible for the choice of a relevant action. The tuning of the action is thus under the control of the "critic" process, which owns the estimation of the value function.

In the Q-learning scheme [25], the external states are valuated according to a transition value function $x_{\text{in}}(t) = Q(\mathbf{u}_{\text{in}}(t-1), \mathbf{u}_{\text{out}}(t))$. If we define $\mathbf{u}(t) = (\mathbf{u}_{\text{in}}(t-1), \mathbf{u}_{\text{out}}(t))$, and consider the simple linear case $Q(\mathbf{u}) = \mathbf{J}\mathbf{u}$,

$$\Delta\mathbf{J}(t) = \alpha(r(t) + x_{\text{in}}(t) - x_{\text{in}}(t-1)) \times \mathbf{u}(t-1) \tag{6}$$

and the *politics* f elects the current action $\mathbf{u}_{\text{out}}$ according to the best expectation on the future valuations

$$\begin{aligned} \mathbf{u}_{\text{out}}(t) &= f_{\text{in}}(\mathbf{u}_{\text{in}}(t-1)) \\ &= \text{argmax}_{\mathbf{v}} Q(\mathbf{u}_{\text{in}}, \mathbf{v}) \end{aligned} \tag{7}$$

The convergence toward a relevant Q function often need an exhaustive exploration of the environment. This exploration is often rooted on a random action generation process. After a while, the system can be more confident on its own estimation and choose its action according to equation (7).

In this approach, the estimation of the value function (seen as an inner model) and the action production are independent processes.

In the reinforcement learning framework, devoting a module for the simulation of an inner model is not always necessary. In that case, the state $\mathbf{x}$ may not fit with an estimate of the current perception, and the inner process f may not mimic any external process. Such "simplified" versions of the reinforcement learning paradigm fall in the category of direct "policy learning" methods [26,27]. The question is not to obtain the best model, but to obtain the most proper action. If the dynamics developed by f is rich enough, the problem is to select the proper action among a series of self generated arbitrary actions. In this kind of system, without any knowledge on the environmental processes, the selection of transition function f^* is obtained by "trial and errors".

This approach is of course highly dependent on:

- the choice of the action generation process. Most of the known methods rely on a random generator, so that reinforcement fall into the field of stochastic optimization methods. We will try to illustrate at the end of this paper that one could also use chaotic dynamics as a generative process, for its versatility and adaptivity may allow a large scope of exploration, with simple and biologically founded methods for stabilizing the suitable interactions;
- the nature, dimension and complexity of the agent's environment.

2 Neuronal modeling

We now present in this section a series of methodological clues for the construction and simulation of large sets of interacting neurons. In continuity with the previous section, we suppose that the considered networks are embedded in a control framework, where $\mathbf{u}$, represent the current observation and $\mathbf{x}$ represents the current state.

2.1 Neuron models

In this paper, we only consider simple neuronal models in order to focus on the collective behaviors rather than on the individuals.

In the following, we will suppose that the spike emissions are bounded to take place on a discrete temporal scale (whose unit is of the order of the temporal resolution T). We will show how such model may realize for instance an implementation of more classical integrate and fire models [28]. For simplicity, we use a virtual temporal unit so that the elementary temporal step T is noted 1.

Let us suppose we have a population of N neurons in our system. The state of neuron i is given by its membrane potential V_i. When V_i reaches the threshold θ, a spike is emitted (the threshold is supposed positive). That spike emission is stored in variable S_i whose value is 1 when a spike is emitted, and 0 elsewhere. The activation dynamics of neuron i is thus formally

$$S_i(t) = H(V_i(t) - \theta) \tag{8}$$

where H is the Heaviside function, which is equal to 1 when $V_i > \theta$, and 0 elsewhere (see also paper 2 [80], equation 1).

The calculation of V_i gives the main features of the neuronal model:

- The neuron may own a memory of its own past potential (leaky integrator).
- It may own a refractory period, i.e. a resting period after spike emission.
- The transmission times may be explicitly encoded between every neuron, with a discrete positive value (those transmission times may correspond to the sum of an axonal delay and PSP transmission delay through the dendrites).

2.1.1 McCulloch and Pitts model

Let us look at a simple model, the McCulloch and Pitts model. It relies on several sets of parameters. First of all, the interaction matrix $\mathbf{J}$, of size $N \times N$, defines the interactions between neurons (see paper 1 [79]). This matrix can be sparse or full. More generally, that matrix defines the pattern of connectivity, giving the global organization of the system (layers, full recurrence, columns, symmetric or non-symmetric connections, etc...). The second series of parameters is the thresholds vector θ, of size N, giving the local sensitivity of the neurons. At last, every neuron is submitted to an external signal $\mathbf{u}$ which is added to the neuron potential. The update of such a system is then

$$\begin{cases} \mathbf{V}(t) = \mathbf{J}\mathbf{S}(t-1) + \mathbf{u}(t-1) \\ \mathbf{S}(t) = H\left(\mathbf{V}(t) - \theta\right). \end{cases} \tag{9}$$

2.1.2 Integrate and fire model

Integrate and fire models allow to precisely model the timing of the spikes and the refractory periods. It is based on a leaky integrator model which maintains a memory of its last potentials, according to an exponential decay scheme (with continuous time t)

$$\frac{dV}{dt} = -\frac{V(t)}{\kappa} + I(t) \tag{10}$$

where $I(t)$ is the global input and κ is the time constant of the neuron. The classical expression of the Integrate and Fire neuron update is

$$\frac{dV_i}{dt} = -\frac{V_i(t)}{\kappa} + \sum_{j=1}^{N} J_{ij} \sum_f \delta(t - T_j^{(f)} - \tau) + u(t) \tag{11}$$

where δ is the indicator function and $T_j^{(f)}$ is the time of the f^{th} spike of the pre-synaptic neuron j and τ is the axonal transmission delay.

With discrete-time Euler numerical integration scheme, one can define a decay parameter $\gamma = 1 - T/\kappa$, where T is the sampling resolution. Note that $\gamma = 0$ means that the sampling is of the order of κ, so that the local memory effects are not modelled. We now have (in matricial notation):

$$\begin{cases} \mathbf{V}(t) = \gamma\mathbf{V}(t-1) + \mathbf{J}\mathbf{S}(t-\tilde{\tau}) + \mathbf{u}(t-1) \\ \mathbf{S}(t) = H\left(\mathbf{V}(t) - \theta\right) \end{cases} \tag{12}$$

where $\tilde{\tau} = \lfloor \frac{\tau}{T} \rfloor$.

At last, several refinement should be added to obtain the classical integrate and fire model, in particular the instant resetting of V after spike emission and the refractory period r with $\tilde{r} = \lfloor \frac{r}{T} \rfloor$, i.e.:

$$\begin{cases} \mathbf{V}(t) = \gamma \mathbf{V}(t-1)\delta(\mathbf{S}(t-1)) + \mathbf{J}\mathbf{S}(t-\tilde{\tau}) + \mathbf{u}(t-1) \\ \mathbf{S}(t) = H\left(\mathbf{V}(t) - \theta\right)\delta\left(\sum_{k=1}^{\tilde{r}} \mathbf{S}(t-k)\right). \end{cases} \tag{13}$$

Remark: if we set the typical neuronal time constant to $\kappa = 10\,\text{ms}$, the axonal delay $\tau = 10\,\text{ms}$ and the refractory period $r = 2\,\text{ms}$, the choice of the sampling resolution $T = 10\,\text{ms}$ implements a McCulloch and Pitts model!

2.1.3 Firing rate model

The firing rate models (i.e. models with continuous activation) are at the opposite side in terms of time precision. The output $x_i(t)$ of a neuron represents the spike firing rate within a certain time window. For the seek of clarity, the firing rate is set to take place within interval $[0, 1]$. A typical activation function is $f(\mathbf{V}, \theta, g) = \frac{1+\tanh(g \times (\mathbf{V}-\theta))}{2}$ where $\mathbf{V}$ is the vector of membrane potential, θ is the threshold and g is the "gain" of the activation function. This gives the network update,

$$\mathbf{x}(t) = f(\mathbf{J}\mathbf{x}(t-1) + \mathbf{u}(t-1), \theta, g). \tag{14}$$

In biological modeling each of the given models may correspond to a particular resolution, i.e.

Model	firing rate	McCulloch and Pitts	Integrate and Fire
T	100 ms	10 ms	1 ms

2.1.4 Various delays

In the two following subsections, we use $\gamma = 0$, and thus use a McCulloch and Pitts model, but the translation to the integrate and fire model is straightforward according to the previous settings (in other words, the resolution T is set to 10 ms).

The use of transmission delays in neuronal processing is traditionally associated to the problem of temporal sequences learning within recurrent dynamical networks. It has been shown for instance that Hebbian learning of temporal patterns remains efficient within a system with a broad range of delays [29]. The important point is that the non-specificity of the delay scattering is not a drawback for learning sequences with long term dependencies. On the contrary, a simple Hebbian mechanism allows to select the appropriate delayed line for the learning of a specific temporal dependency.

Introducing non-homogeneous delays means to introduce a new series of parameters, namely the matrix of positive integer delays $\mathbf{T} = (\tau_{ij})_{i,j=1\ldots N}$. The update of the system is now given by:

$$\forall t \geq 1, \forall i \in \{1, \ldots, N\}$$
$$\begin{cases} V_i(t) = \sum_{j=1}^{N} J_{ij} S_j(t - \tau_{ij}) + u_i(t-1) \\ S_i(t) = H\left(V_i(t) - \theta_i\right). \end{cases} \tag{15}$$

The delay repartition may rely on physiological studies. If we suppose for instance that the mean delay is of the order of 10 ms, the expectancy of the τ_{ij}'s should be of the order of $10/T$.

2.1.5 Several populations

A network can be defined as a pool of P interacting populations of neurons, of respective sizes $N^{(1)}, \ldots, N^{(P)}$, where the global number of neurons is $N = \sum_{p=1}^{P} N^{(p)}$. The synaptic weights from population q towards population p are stored in a matrix $\mathbf{J}^{(pq)}$ of size $N^{(p)} \times N^{(q)}$. The state vector of population p at time t is $\mathbf{S}^{(p)}(t)$, of size $N^{(p)}$. The initial distribution of spikes $S_i^{(p)}(0)$ is set according to a random draw in $\{0,1\}$.

At each time step $t \geq 1$, $\forall (p,q) \in \{1,..,P\}^2$, $\forall i \in \{1,\ldots,N^{(p)}\}$,

$$h_i^{(pq)}(t) = \sum_{j=1}^{N^{(q)}} J_{ij}^{(pq)} S_j^{(q)}(t - \tau_{ij}^{(pq)}) \tag{16}$$

is the *local field* of population q towards neuron i of population p, and $\tau_{ij}^{(pq)}$ is the transmission delay from neuron j to neuron i.

We also consider input signals $\mathbf{u}^{(p)} = \{\mathbf{u}^{(p)}(t)\}_{t=1..+\infty}$, where $\mathbf{u}^{(p)}(t)$ is a $N^{(p)}$ dimensional input vector at time t on population p. The input $\mathbf{u}^{(p)}(t)$ acts like a bias on each neuron[5]. Then, the global equation of the dynamics is :

$$\begin{aligned} &\forall t \geq 1, \quad \forall p \in \{1,\ldots,P\}, \quad \forall i \in \{1,\ldots,N^{(p)}\} \\ &S_i^{(p)}(t) = H\left(u_i^{(p)}(t-1) + \sum_{q=1}^{P} h_i^{(pq)}(t) - \theta^{(p)}\right). \end{aligned} \tag{17}$$

2.2 Learning rules

In neuronal modeling, learning means “weight adaptation”. The neuron synapses are modulated by the characteristics of the signals arriving at the synaptic interface, according to Hebb's principle [30]. The basic Hebbian rule states that a synapse is reinforced when the arrival of a pre-synaptic signal repeatedly coincides with post-synaptic spike emission. The emission of a post-synaptic spike *after* pre-synaptic firing is thus facilitated.

The Hebb's definition being rather vague, numerous realizations of the Hebb's rule have been proposed.

A good Hebbian learning rule is in first place a rule which is:

- local: the rule must rely on signals that are available in the vicinity of the neurons.
- sufficiently plausible in biological terms, i.e. still measured and/or realizable at low cost at the level of the neurons.

The first and most widely admitted implementation of the Hebb's rule is the direct product of pre-synaptic and post-synaptic activity [1] (see also paper 1 [79], part 6.5):

$$\varDelta J_{ij}(t+1) = \frac{\alpha}{N} S_i(t) \times S_j(t - \tau_{ij}) \tag{18}$$

where α is the learning parameter, scaled with the number of afferent links N. We call it “order 0” Hebb rule since it only takes into account the instantaneous values of the neurons action potentials. In the particular case where the S_i's belong to $\{-1,1\}$ (bipolar neurons), which is of course unrealistic, the Hebb's rule is said to be balanced as the probability of synaptic potentiation is equal to the probability of depression. In that case, the rule can been interpreted as an instantaneous measure of correlation [3]. In the standard binary case ($S_i \in \{0,1\}$), the rule is notably unbalanced since only synaptic potentiation is allowed. The weights thus tend to diverge for long time, even if some bounding factor may be added to the learning mechanism in order to avoid too strong weight drift.

The weights may for instance be normalized at each learning step with the following synaptic scaling mechanism [1]:

$$J_{ij}(t+1) = J_{ij}(t) + \varDelta J_{ij}(t+1) - \frac{\sum_{k=1}^{N} \varDelta J_{ik}(t+1)}{N} \tag{19}$$

[5] On the contrary to Hopfield system [3], the input is not supposed to correspond to the initial state $x_i^{(p)}(0)$ of the network.

under which the simple Hebb's rule performs contrast enhancement, i.e. favors the current subset of active neurons and leaves the inactive synapses disappear. In general, the order 0 rules only take into account the current neuron activity and thus tend to reinforce the neurons whose activity is strong and to weaken the neurons whose activity is weak, giving rise to a contrast enhancement effect[6]. In that case, the learning process will reinforce the order 0 characteristics of the neurons dynamics (their mean activation).

It can be noticed that the synaptic scaling mechanism is considered as biologically plausible [31].

In the framework of dynamical neural networks, it can be interesting however to take into account some differential aspects of the neuronal activity. It is known, actually, that the *contrast* within the individual activity may be more significant than its mere absolute value. Under that hypothesis, the *difference* in neuronal activity $S_i(t) - S_i(t-1)$ may be taken into account in some part of the learning rule. The most straightforward proposal is the following (that we call the conjugate difference rule):

$$\Delta J_{ij}(t+1) = \frac{\alpha}{N}(S_i(t) - S_i(t-1)) \times (S_j(t-\tau_{ij}) - S_j(t-\tau_{ij}-1)) \tag{20}$$

under which the learning rule implements a rough approximation of the neurons delayed covariance. In a context of binary neurons, this rule is found to be balanced, on the contrary to the simple Hebbian product. This rule operates on the differences of activity and thus enhances the transmission between neurons whose activity rapidly switches, and weakens the ones that maintain a stable level of activity.

A derivative of this rule is the classical "temporal difference" rule [23] (see equation (3) and further), where the post-synaptic switch enhancement is under the control of the pre-synaptic neuron.

$$\Delta J_{ij}(t) = \frac{\alpha}{N}(S_i(t) - S_i(t-1)) \times S_j(t-\tau_{ij}). \tag{21}$$

With the TD rule, the weight is reinforced when $S_i(t) > S_i(t-1)$ (which means that $S_i(t) = 1$ and $S_i(t-1) = 0$) and when the PSP[7] arrives at time t (according to the transmission delay τ_{ij}). The weight is thus enhanced when the PSP and the post-synaptic action potential exactly coincidate in time. The weight decreases when $S_i(t) < S_i(t-1)$, i.e. when the PSP arrives at time t, shortly *after* a spike has been emitted at time $t-1$. The TD rule can thus been interpreted as a refined coincidence detection mechanism.

Another parameter that should be taken into account in order to balance the simple Hebbian rule is the mean neuronal activity (or frequency) that we will note m_i. It may correspond to the mean firing measure $m_i(t) = \frac{1}{t}\sum_{k=0}^{t} S_i(k)$, or, more practically, to an instantaneous estimate of the mean firing $m_i(t) = (1-\beta)S_i(t) + \beta m_i(t-1)$ with β close to 1. It has been shown in [27] that the use of the difference $(S_i(t) - m_i(t))$ may help to implement direct optimization algorithms with the use of stochastic neurons. The expression of the rule is the following:

$$\Delta J_{ij}(t+1) = \frac{\alpha}{N}(S_i(t) - m_i(t)) \times (S_j(t-\tau_{ij})). \tag{22}$$

This rule can be seen as a regulation of the simple Hebbian rule as it only potentiates (or depresses) the neurons whose activity is susceptible to undergo significant change in their response. On the contrary to the direct difference rules, this rule tends to favor the transitions taking place at the level of the global organization (to identify the changes in the repartion between active and inactive neurons for instance). The synaptic enhancement is supposed to activate when significant transitions take place in the relationship between the neurons.

At last, we must take in consideration the covariance rule [32] which roots on a fine estimation of the covariance between the pre-synaptic and the post-synaptic neuron:

$$J_{ij}(t) = J_{ij}(t-1) + \frac{\alpha}{N}(S_i(t) - m_i(t)) \times (S_j(t-1) - m_j(t-1)). \tag{23}$$

[6] Also known as the "Matthew effect: "*For unto every one that hath shall be given, and he shall have abundance: but from him that hath not shall be taken away even that which he hath*" - Mat 25:29 -.

[7] Post Synaptic Potential.

Some qualitative effects of the Hebb rules have been presented in paper 1 [79], part 6.5, and some other will be presented further.

The biological relevance of the Hebbian rule has long been conjectural since the first observation of a potentiation mechanism based on the co-activation of pre-synaptic and post-synaptic neurons [33]. The lasting potentiation of the synapse is commonly called “Long Term Potentiation” (LTP), and the reverse “Long Term Depression” (LTD). More recent observations have shown that the timing of spike arrivals may be of critical importance in the mechanism of synaptic potentiation [34,35]. This mechanism has been called Spike-Time Dependent Plasticity (STDP). The STDP can be seen as a coincidence detection mechanism whose precision may be of the order of few milliseconds. The main effect of a STDP rule is to potentiate the sequential co-activation: the EPSP that anticipates the arrival of a spike on the post-synaptic neurons lead to a synaptic potentiation. The EPSP taking place several milliseconds *after* spike emission leads to a synaptic depression.

A classical expression of the STDP rule is the following [36]:

$$\Delta J_{ij}(t+1) = \begin{cases} A_+ \exp(\Delta t/\tau_+) & \text{if} \Delta t < 0 \\ -A_- \exp(-\Delta t/\tau_-) & \text{if} \Delta t > 0 \end{cases} \tag{24}$$

where Δt is the temporal difference between the time of PSP arrival and the time of spike emission, A_+ and τ_+ calibrate the LTP, A_- and τ_- calibrate the LTD.

In accordance with the integrate and fire model (equation (13)), we show here that the STDP rule can be implemented using local indicators, i.e. without explicitly storing the temporal difference Δt between pre-synaptic and post-synaptic spike. For that, we locally store for each neuron a vector $\mathbf{m}$ which memorizes the history of recent spike emissions, with the unified decay parameter $\gamma = 1 - T/\kappa$, where T is the sampling resolution and κ is the time constant of the neuron. In the framework of integrate and fire neurons, the STDP rule may be implemented in the following way :

$$\begin{cases} m_i(t) = (1-\gamma)S_i(t-1) + \gamma m_i(t-1) \\ \Delta J_{ij}(t) = \frac{\alpha}{N}\left(m_j(t-\tau_{ij}+1) \times S_i(t) - m_i(t) \times S_j(t-\tau_{ij})\right). \end{cases} \tag{25}$$

Now, taking $\gamma = 0$, the STDP rule simplifies as follows:

$$\Delta J_{ij}(t) = \frac{\alpha}{N}\left(S_i(t) - S_i(t-1)\right) \times S_j(t-\tau_{ij}). \tag{26}$$

It can be noticed that the close functional equivalence between the TD rule and the STDP rule has been noticed in [12]. Under the hypothesis of a sampling $T = 10\,\text{ms}$, the TD rule is thus seen as a low-resolution approximation of the STDP rule, *but it should be functionally equivalent.* The TD rule may thus be applied in replacement of the STDP rule in a McCulloch and Pitts model under the hypothesis that the sampling is of the order of the neuron potential relaxation ($\kappa \simeq T \simeq 10\,\text{ms}$).

2.3 Networks construction

Every particular synaptic weight, every activation threshold and every delay is a parameter for the system. In the design of large networks, those parameters can not be fixed by hand, and some global parameters have to be set in order to describe the average strength and shape of the couplings between groups of neurons. Those global parameters (also called macroscopic parameters - see paper 1 [79]) may for instance describe the distributions of weights, thresholds and delays.

In the design process of elaborate networks, it is interesting to minimize the set of global parameters in order to keep the global description as generic as possible, allowing several concrete implementations of the same scheme (with different spatial and temporal resolutions).

It seems for instance interesting *not* to allow the network size N to belong to the global parameters. With a proper definition of the weights distribution, the dynamical behavior should be *independent of the size.*

A relevant macroscopic parameter for weights definition is for instance the distribution of the sum of the afferent weights $J_i = \sum_{j=1}^{N} J_{ij}$, which is independent of N (a Gaussian distribution is taken most of the time). Knowing the distribution of the J_i's, the distribution of the J_{ij}'s can then be fixed with respect to the J_i's mean and standard deviation.

In the most simple case, if we suppose that the distribution of the J_i's is $\mathcal{N}(\bar{J}, \sigma_J^2)$, we can take for individual weights setting any distribution whose mean is $\frac{\bar{J}}{N}$ and whose standard deviation is $\frac{\sigma_J}{\sqrt{N}}$ (see also paper 1 [79], part 6.1). Those settings have been justified in the previous papers in terms of local field scaling in order to allow the calculation of their large size limit. It can also be considered as an off-line realization of the synaptic scaling principle [31].

With respect to the global law $\mathcal{N}(\bar{J}, \sigma_J^2)$, some refinements can be introduced in order to get closer to biological plausibility: weights sparsity, weights sign specification and non-homogeneous delays.

2.3.1 Weights sparsity

In order to build the more generic models, we wish the weights sparsity ρ *not* to belong to the macroscopic parameters! This means that the qualitative behavior of a particular network realization should be the same whatever the connectivity pattern is sparse or not. The following individual weights settings are designed in order to maintain the distribution of the J_i's in accordance with $\mathcal{N}(\bar{J}, \sigma_J^2)$. We propose to define the sparsity with the help of a binomial law $\mathcal{B}(\rho)$: the probability of a connection between neuron i and neuron j is ρ. In that case, the mean number of afferent weights is ρN, so that the mean of the nonzero weights is $\frac{\bar{J}}{\rho N}$ and the standard deviation of the nonzero weights is $\frac{\sigma^*}{\sqrt{\rho N}}$, where σ^* is defined such that $\text{var}(J_{ij}) = \frac{\sigma_J^2}{N} = \rho(1-\rho)\left(\frac{\bar{J}}{\rho N}\right)^2 + \rho\left(\frac{\sigma^*}{\sqrt{\rho N}}\right)^2$, i.e. ${\sigma^*}^2 = \sigma_J^2 - \frac{1-\rho}{\rho}\frac{\bar{J}^2}{N}$. In that case, N has a lower bound which is $\frac{1-\rho}{\rho}\left(\frac{\bar{J}}{\sigma_J}\right)^2$. The value $d = \frac{\bar{J}}{\sigma_J}$ is the *eccentricity* of the J_i's distribution (the relative shift toward positive or negative values). When the eccentricity is not zero, there is a minimum number of neurons in the network for the sum of the afferent weights to attain the mean $\bar{J}$. When $N = \frac{1-\rho}{\rho}d^2$, the nonzero weights are constant and the variance between the J_i's comes from the difference in the number of afferent links from neuron to neuron.

2.3.2 The design of excitatory (vs. inhibitory) populations of neurons

In order to design strictly excitatory or inhibitory populations of neurons, we need to use bounded distributions and certify that the individual links distribution lower bound is 0. In particular, the individual weights can not be defined according to a Normal law. We nevertheless persist here in trying to maintain the distribution of the J_i's in accordance with $\mathcal{N}(\bar{J}, \sigma_J^2)$! The most simple bounded distributions are thus uniform distributions, which are regular enough for the sum of several random variable to rapidly approach a gaussian distribution (according to the law of large numbers). We note $\mathcal{U}(m, \sigma^2)$ the uniform distribution within interval $[m - \sqrt{3}\sigma, m + \sqrt{3}\sigma]$. We make the hypothesis that the individual weights are drawn according to $\mathcal{U}\left(\frac{\bar{J}}{\rho^* N}, \frac{{\sigma^*}^2}{\rho^* N}\right) \times \mathcal{B}(\rho^*)$ where σ^* and ρ^* are such that the J_{ij}'s lower bound is 0. In case $\bar{J} > 0$ for instance, we have to certify that $\frac{\bar{J}}{\rho^* N} \geq \sqrt{\frac{3}{\rho^* N}}\sigma^*$, i.e. ${\sigma^*}^2 \leq \frac{\bar{J}^2}{3\rho^* N}$. If we fix 0 as the lower bound, then ${\sigma^*}^2 = \frac{\bar{J}^2}{3\rho^* N}$. Knowing that ${\sigma^*}^2 = \frac{\rho^* N \sigma_J^2 - (1-\rho^*)\bar{J}^2}{\rho^* N}$ (see previous paragraph), we find

$$\rho^* = \frac{4\bar{J}^2}{3(N\sigma_J^2 + \bar{J}^2)}$$

i.e. ρ^* is $\mathcal{O}\left(\frac{1}{N}\right)$. We find that the number of afferent weights

$$\rho^* N = \frac{4\bar{J}^2}{3\left(\sigma_J^2 + \frac{\bar{J}^2}{N}\right)} \overset{N\to\infty}{\longrightarrow} \frac{4}{3}d^2$$

i.e. the network falls in the category of strongly diluted networks. It can be noticed that a network of that category can be defined according to only two macroscopic parameters, for instance $\bar{J}$ and d. In that model, the eccentricity also gives the sparsity of the weights. Reversely, the weights sparsity gives the eccentricity, and thus the variability of the repartition of the J_i's according to the law of large numbers.

2.3.3 Transmission delays parameterization

When the delays are not homogeneous, one has to define a new global parameter in order to characterize their distribution. In the following, every delay τ_{ij} is set according to $\tau_0 + \mathcal{P}(\lambda)$ where τ_0 is the minimal delay and $\mathcal{P}$ is a Poisson law of parameter λ. The mean transmission delay is thus $\tau_0 + \lambda$. The definition of delays thus needs two more macroscopic parameters: τ_0 and λ. For simple unitary delays, $\tau_0 = 1$ and $\lambda = 0$.

If we consider that a typical transmission delay is 10 ms, we can define the delays distribution according to the temporal resolution T in the following way: $\tau_0 = 1$ and $\lambda = \frac{10}{T-1}$, with $T \leq 10\,\text{ms}$.

2.3.4 Remarks

Random networks

With the previous settings, the networks belong to the category of Random Recurrent Neural Networks (RRNNs - see paper 2 [80]). The connectivity pattern is non-symmetric, so that one can not ensure the convergence of the dynamics towards a fixed point (see paper 1 [79], part 5.2). Random recurrent neural networks (RRNN) have been introduced by Amari [2] in a study of their large size properties. Predictions on the *mean field* of such systems can be obtained in the limit of large size under an hypothesis of independence of the individual signals [37,38], and under a condition of homogeneity of the law of the weights in a given population p (i.e the mean field equations are valid in a multi-population model, see paper 2 [80]). Autonomous RRNN's (i.e. $\forall t, \mathbf{u}(t) = 0$) are discrete time dynamical systems, that can for instance display a generic quasi-periodicity route to chaos while progressively increasing the gain of a continuous transfer function [39]. All those regimes and their conditions of appearance are reliably predicted by the mean field equations (see paper 2 [80]). To our knowledge, no mean field rigorous result yet exists in the case of strongly diluted networks.

Network parameters

In the most general framework, the macroscopic parameters of a class of networks can be described by several matrices : $\bar{J} = \begin{pmatrix} \bar{J}^{(11)} & \dots & \bar{J}^{(1P)} \\ & \cdots & \\ \bar{J}^{(P1)} & \dots & \bar{J}^{(PP)} \end{pmatrix}$, $\sigma_J = \begin{pmatrix} \sigma_J^{(11)} & \cdots & \sigma_J^{(1P)} \\ & \cdots & \\ \sigma_J^{(P1)} & \cdots & \sigma_J^{(PP)} \end{pmatrix}$, $\bar{\theta} = \begin{pmatrix} \bar{\theta}^{(1)} \\ \cdots \\ \bar{\theta}^{(P)} \end{pmatrix}$, $\sigma_\theta = \begin{pmatrix} \sigma_\theta^{(1)} \\ \cdots \\ \sigma_\theta^{(P)} \end{pmatrix}$, τ_0, λ. Those matrices own the most general description of a family a random neural networks which can be implemented under various spatial and temporal scales. The general mean-field equations of such models are given in paper 2 [80] (equation 40).

3 Internal dynamics and bifurcations

The *versatility* of an agent is the propensity by which it may undergo transitions/bifurcations in its behavior. Every cyclic or chaotic attractor the global system may attain can be seen as a particular *functional regime* of the interaction. For an agent, the versatility is a sign of fast adaptivity and variety in its behaviors.

A versatile agent may undergo two kinds of constraints:

- there is a need for a particular regime to remain stable under various distractions;
- there is a need for the different regimes to alternate in order to fit with environmental constraints.

There is thus an trade-off between the necessity to stabilize some behaviors, and the necessity to switch behavior when the environmental contexts requires it (Stability/Plasticity dilemma, see [40]).

We are here interested in the way some bifurcations may be obtained inside the dynamics of a network under the effect of some "control parameter". There are two ways by which we can identify the control parameter that may cause such bifurcations.

- The first way is to identify the control parameter with the incoming command $\mathbf{u}_{in}(t)$ in equation (1).
- The second way is to define a specific adaptation process by which the transition function f is submitted to a parametric change.

3.1 Networks under external influence

In the most general framework, a network which is submitted to an external influence is a non-autonomous dynamical system.

$$\frac{d\mathbf{x}_{in}}{dt} = f(\mathbf{x}_{in}, \mathbf{u}_{in})$$

where $\mathbf{x}_{in}$ is the internal state and $\mathbf{u}_{in}$ is the external signal (also called "stimulus"), having its own evolution.

We suppose that the external signal $\mathbf{u}_{in}$ is *continuously* influencing the internal dynamics. We assume in particular that:

- No hypothesis is made on the stationarity or repeatability of the input signal. The dynamics of the system is not supposed to properly converge toward a particular attractor.
- The initial conditions of the dynamic system have no long term effect on the dynamics, which can be disrupted and re-routed under the influence of the stimuli.
- Under some conditions of non-reversibility (i.e. "breaking of symmetry"), significant environmental fluctuation are needed for the system to go back to a previous configuration, or to reach a new configuration.

Under these conditions, the system may undergo various transitions and explore various attraction basins in its input/state space. The transitions between those attraction basins may be expressed in terms of changes in the shape and structure of the trajectories.

We present here some of the models whose dynamics is interesting in terms of pattern formation under external influence.

3.1.2 Systems with dynamic multistability

Switching behaviors can be obtained in various models of neural networks. Switching can be obtained classically through bifurcation under the effect of a slow change of some parameters of the system (control parameters). Another way to obtain switching behavior, without explicit parameter drive, is to define high dimensional systems where a slow dynamics is coupled with a fast dynamics. The slow dynamics thus plays the role of a parametric drive over the fast dynamics. The next paragraphs give some examples.

Route to chaos in recurrent models

In large recurrent neural networks with random connectivity and continuous activation (equation (14)), a generic route to chaos by quasi-periodicity can be observed (see also

[8] One can note that Mexican hat kernels have a wide range of application in Self organizing Maps [42], but self organized classification systems are not in the scope of this paper.

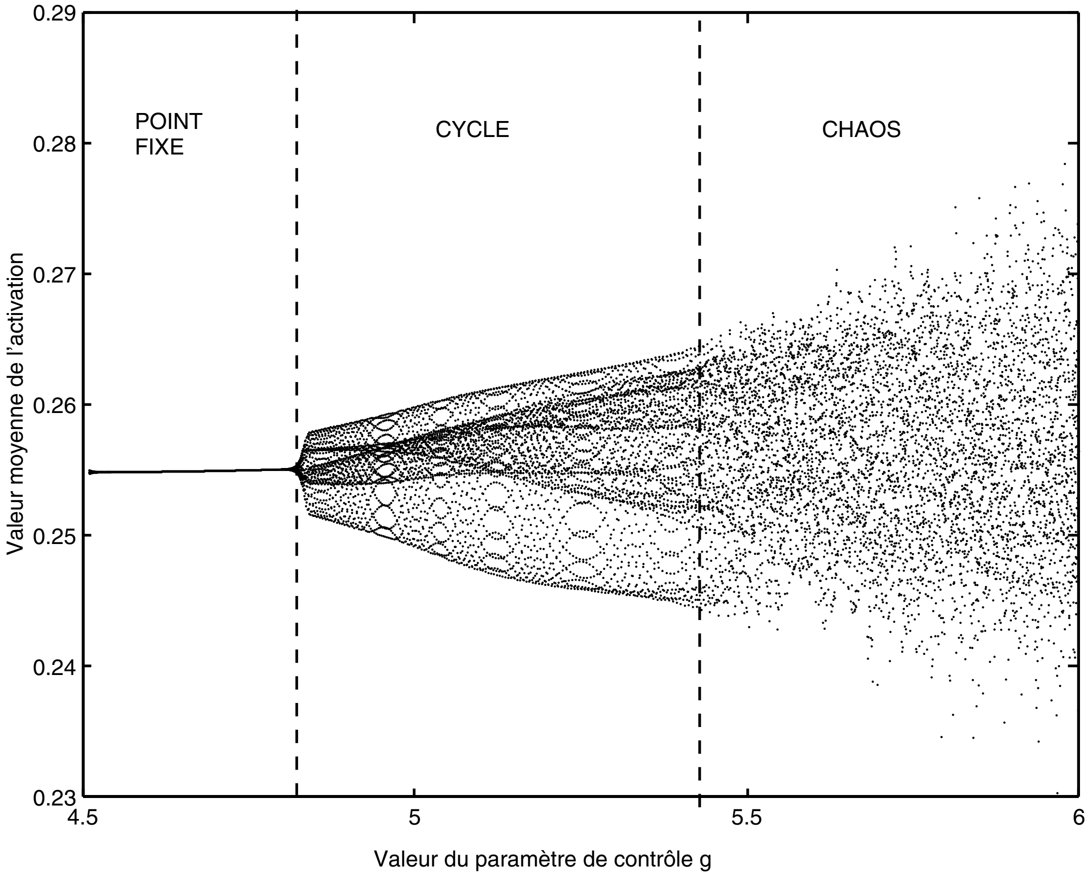

Fig. 6. Generic quasi periodicity route to chaos in continuous random network. The gain parameter g slowly increases from left to right.

Every transition, from fixed point to cycle, T2 torus, frequency locking and chaos modifies the behavior of the system by steps, from order (fixed point) to strong disorder (deep chaos).

Switching behavior with quasi-stationary inputs

A recurrent model under external drive can be modeled in the following way:

$$\mathbf{x}(t) = f(\mathbf{J}\mathbf{x}(t-1) + \mathbf{u}(t-1), \theta, g) \tag{28}$$

where g is given and the drive $\mathbf{u}(t)$ is almost stationary. This drive can be for instance a random pattern whose values slowly and linearly evolve from random pattern $\mathbf{P}^{(1)}$ to random pattern $\mathbf{P}^{(2)}$ as time goes on. Figure 7 presents some aspects of the evolution of such a network under the effect of a slow external drive during 10000 time steps. It clearly appears that such dynamics also evolves by steps, despite the input evolution is continuous. Some temporary attractors get stabilized for a while, then undergo some deformations, and finally vanish and get replaced by other attractors with different shape and intrinsic periodicity.

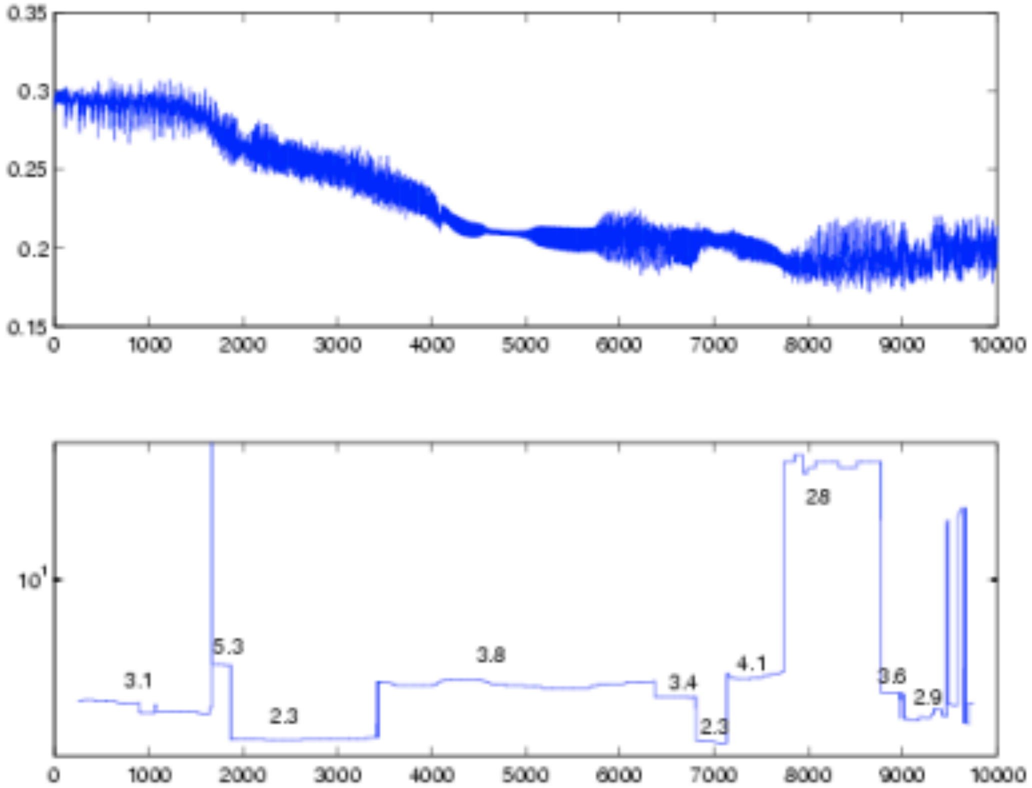

Fig. 7. System multistability under quasi-stationnary external drive. The upper figure gives the evolution of the mean activity. The lower figure gives the intrinsic period of the system over 500 time steps sliding windows. This figure is taken from [52].

Synchronous behaviors can be observed in networks owning at least two populations of neurons, one population being excitatory, the other one being inhibitory. This basic network structure is a first step toward biological plausibility, where structures owning cooperating populations of excitators and inhibitors are broadly found. As stated in section 2.3.2, such a network can be defined with very few parameters, namely d (eccentricity) and k (coupling parameter), i.e. $\bar{J} = \frac{1}{2}\left(\begin{smallmatrix} 1 & -k \\ k & -1 \end{smallmatrix}\right)$ and $\sigma_J = \frac{1}{2d}\left(\begin{smallmatrix} 1 & \sqrt{k} \\ \sqrt{k} & 1 \end{smallmatrix}\right)$. In such networks, population 1 is the excitatory one and population 2 is the inhibitory one. Those settings are inspired by the "neural oscillator" scheme (see paper 1 [79], figure 25), except that the self feeding of the inhibitory population is inhibitory. See also the settings of paper 2 [80], equation 41 for another example of excitatory/inhibitory interaction scheme. In the present case, the interactions between populations are stronger than the interactions within populations, by a factor k (coupling factor), and the mean number of links between populations is k times the mean number of links within a population.

Figure 8 presents a transition occurring in this kind of network where the control parameter is a macroscopic one: the eccentricity d. For $k = 5.5$, and d varying between 1.2 and 1.8, a transition from unsynchronized chaos to synchronized chaos can be observed. The property of synchronization is not a specificity of that particular network. Those behaviors are theoretically tractable at the limit of large sizes [62,63]. The period of oscillation linearly depends on the range of the delays [60], which is also the case in our model. In deterministic systems, one can also obtain under different parameter sets chaotic regimes (without synchronization) [64] or either synchronized chaos [48] or cyclostationary chaos [65] (see also paper 2 [80]). More generally, synchronizing behaviors in unitary delays networks depend on the coupling factor k, i.e. inhibition has to dominate excitation on the excitatory layer for the network to produce synchrony. This point has been overlooked in other simulation works, see for instance [66,67].

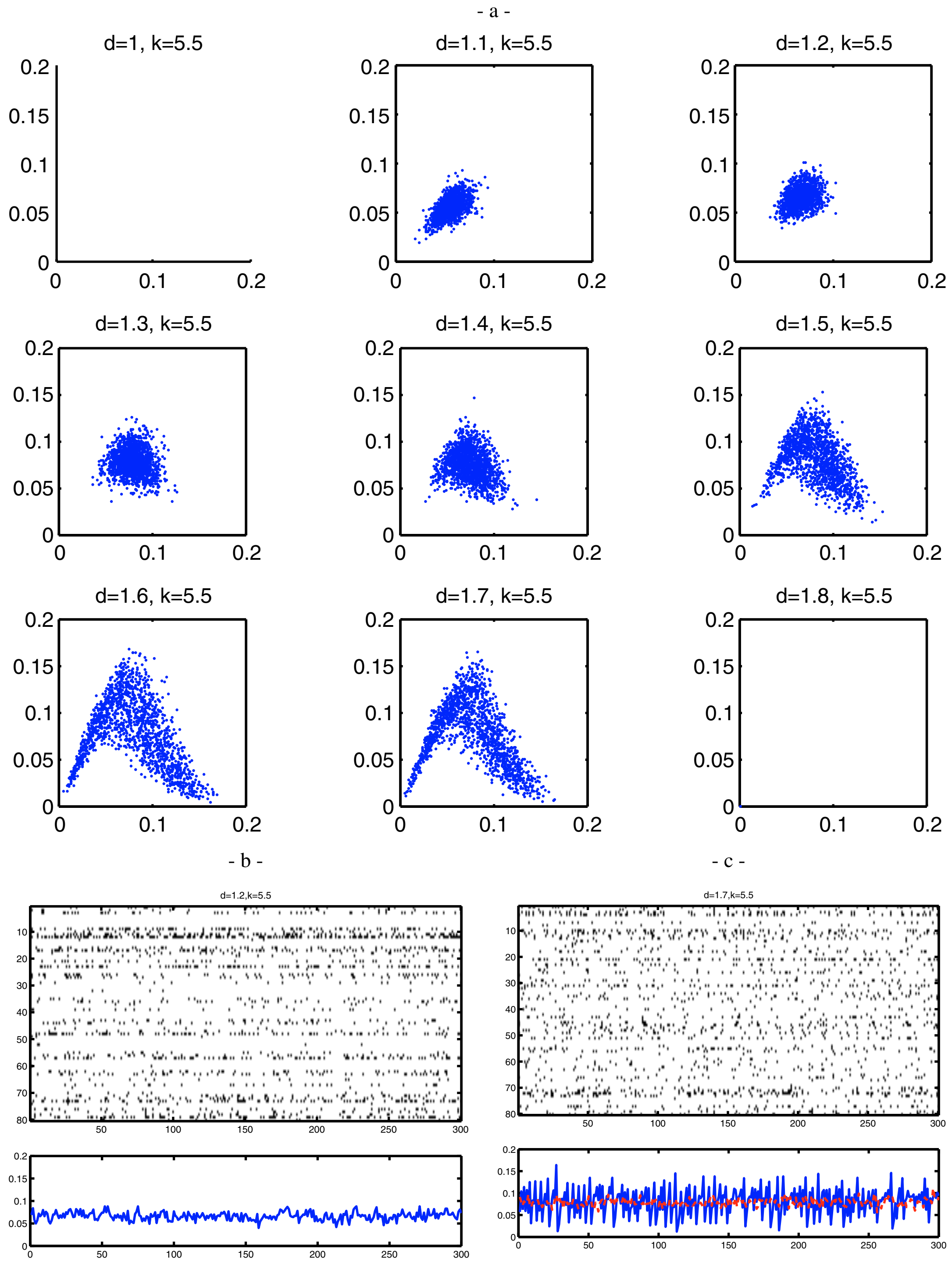

Fig. 8. - a - Return maps of the mean activity of the excitatory layer, for networks whose eccentricity d vary from 1.2 to 1.8, with coupling $k = 5.5$ **- b -** Activity of the excitatory layer, with $d = 1.2$. The mean activity is on the lower part of the figure. **- c -** Activity of the excitatory layer, with $d = 1.8$. The mean activity is on the lower part of the figure, with a red line showing for comparison the amplitude of the sum of independent activities.

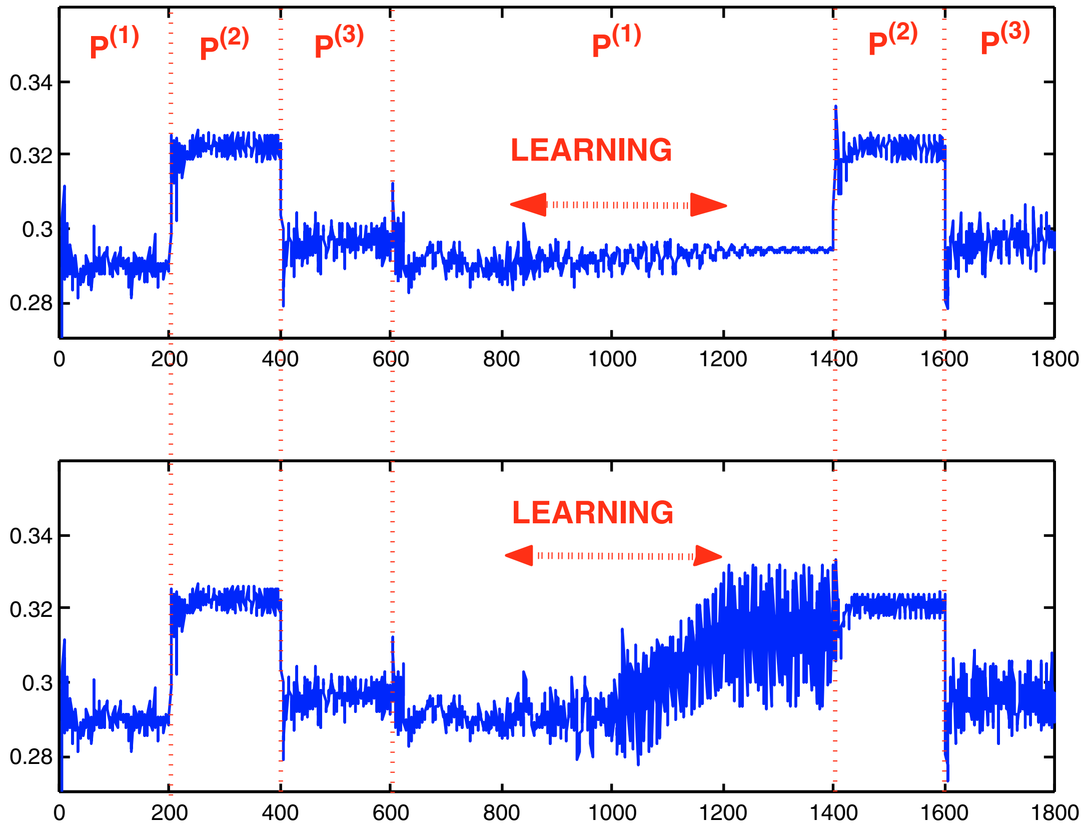

Fig. 9. Mean activity in a random network of 500 neurons, under the influence of 3 static gaussian random patterns $\mathbf{P}^{(1)}$,$\mathbf{P}^{(2)}$ and $\mathbf{P}^{(3)}$. A learning process is activated between $t = 800$ and $t = 1200$. Upper figure: Simple Hebbian learning process ($\alpha = 0.001$). Lower figure: TD learning process ($\alpha = 0.05$).

presented for periods of 200 time steps. Then, pattern $\mathbf{P}^{(1)}$ is learned for 400 time steps. At last, The three patterns are presented anew.

The two rules induce changes in the state/input dynamics, but they operate in an opposite manner. The simple Hebbian rule tends to extinct the dynamics, giving rise to small limit cycles or fixed points, while the TD rule (equation (21)) tends to amplify the variations taking place in the dynamics, giving rise to large amplitude limit cycles. In the two cases however, the dynamics tends to simplify, from disorder to order.

The long term application of those rules has a strong effect on the reactivity of the system. In the first case, the dynamics converges toward a fixed point, in the second case to a cycle with a high level of activity (of the order of 0.5), but most of all in the two cases, the response of the system becomes stereotypic, i.e. any input pattern will lead the system to the same response.

3.2.3 Transitions from asynchrony to synchrony

Starting from asynchronous chaos, like in figure 8 - b -, a specific learning process can also produce an increase of the synchrony. This learning process has been designed in order to favor the *first spike* of a neuron after significant decrease of the neuron potential. Namely, given the coupling factor k, a given neuron i is allowed to learn if its potential has reached the negative value $-0.2k$. Its “token” T_i is thus set to 1. The neuron is allowed to learn as soon as it has emitted a spike. Then, the token is set to 0 until the potential reaches the negative

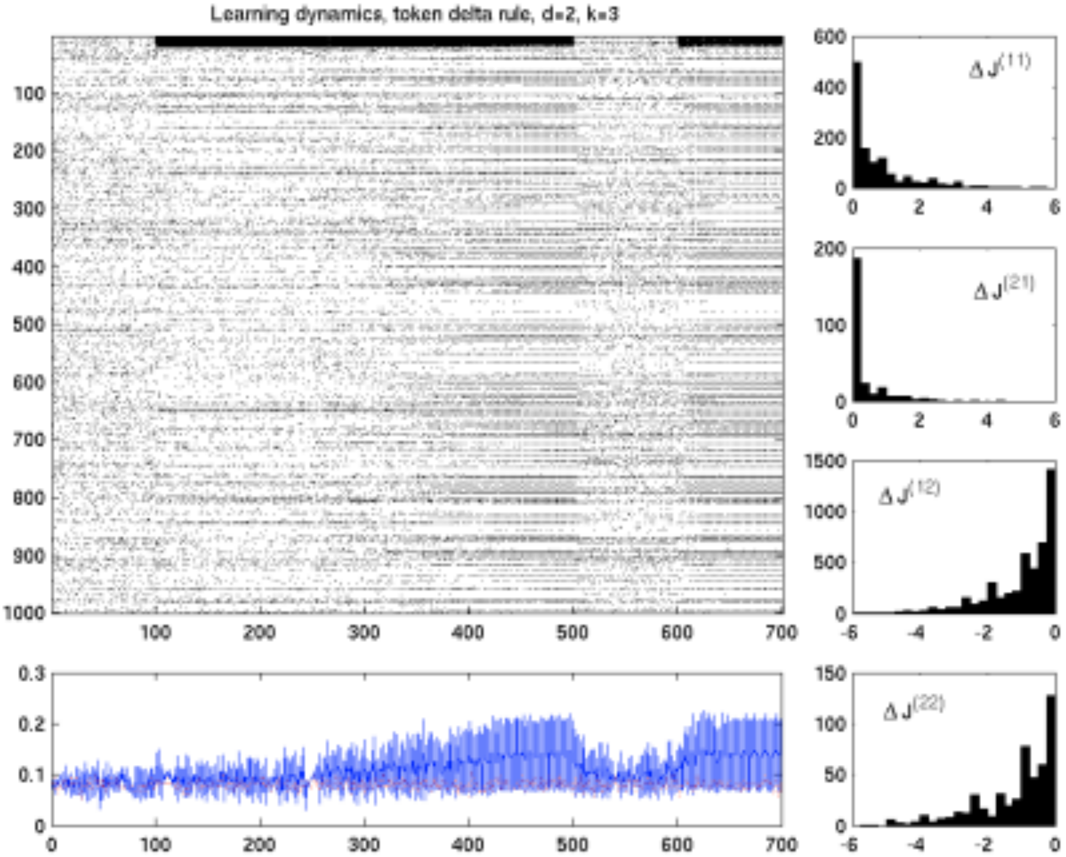

Fig. 10. Application of a TD rule with token in a network of 1000 neurons, with $\alpha = 1$. A pattern is presented at time $t = 100$ and the learning process is activated until $t = 500$. After learning, the network reaction is significantly synchronized when the known pattern is presented. The right figures give the repartition of the weights differences for every class of links.

threshold anew.

$$\begin{cases} \Delta J_{ij}(t) = \frac{\alpha T_i(t-1)}{\rho^* N}(S_i(t) - S_i(t-1))S_j(t-1) \\ T_i(t) = \begin{cases} 1 \text{ if } V_i(t) < -0.2k \\ 0 \text{ if } S_i(t) < S_i(t-1) \\ T_i(t-1) \text{ elsewhere.} \end{cases} \end{cases} \tag{30}$$

An example is given figure 10. Under those settings, the synchrony of the neurons significantly increases, but no drift of the neurons activity is observed.

3.2.4 Hebb and anti-Hebb rules

The use of a negative α gives a “anti-Hebb” rule, i.e. a rule which inverts Hebb’s principle. A anti-Hebb rule thus favors the transmissions between neurons which are *not* correlated (and diminishing the transmission between correlated neurons). The alternation between Hebb and anti-Hebb rules thus allows to control the degree of disorder. As Hebb rules favors order, anti-Hebb rules favors the disorder and drives the system toward more chaotic and less synchronized regimes.

Figure 11 illustrates the mirror effect of the successive application of Hebb and anti-Hebb rule on a network. As the Hebb rule increases the synchrony, the anti-Hebb rule diminishes the synchrony in a symmetrical fashion.

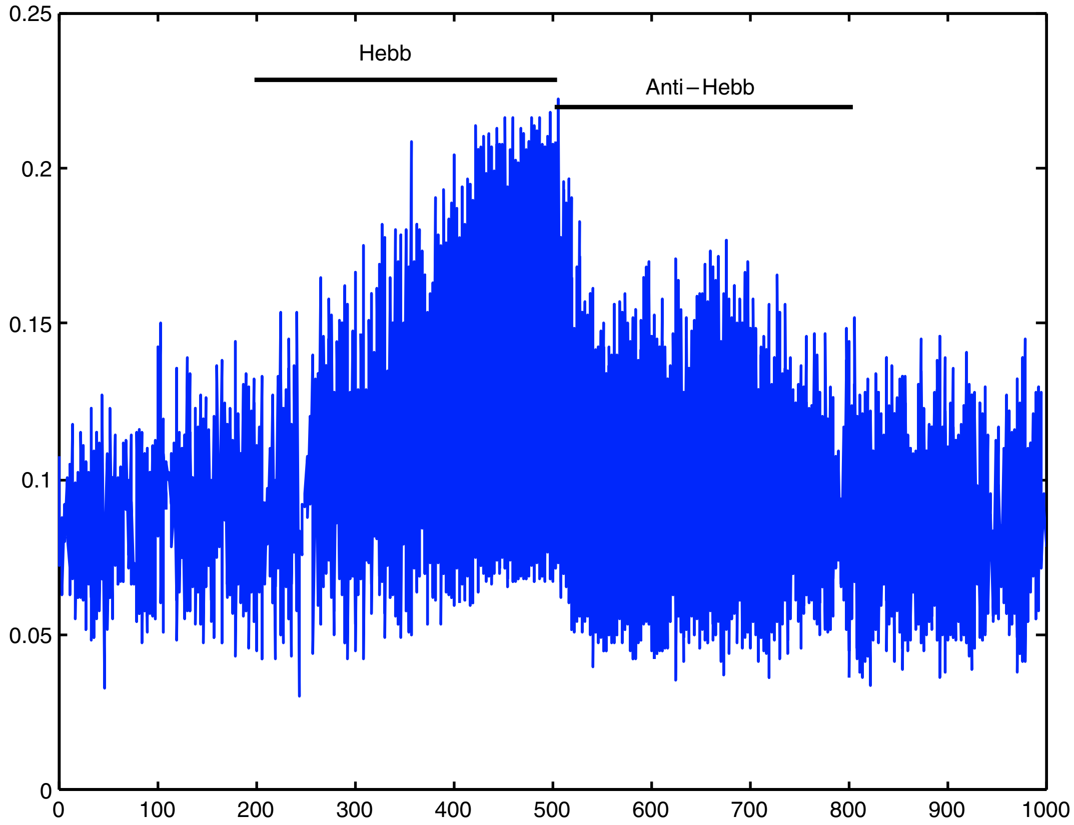

Fig. 11. Application of a TD and anti-TD rule with token in a network of 1000 neurons, with $\alpha = 1$. This figure only shows the mean activity. The TD rule is applied between $t = 200$ and $t = 500$. The anti-TD rule is applied between $t = 500$ and $t = 800$.

3.3 Transitions in biology

In biology, the development and improvement of global non-invasive techniques has enriched the knowledge of the dynamics and transitions in neuronal activity. In particular, the property of *synchronization* has been extensively observed within local assemblies [69] or between different assemblies [14]. Such synchronization may express the involvement of a particular area in a process taking place at the global level, and thus express the cooperation between the local and the global level. Some authors have also stressed the role of *desynchronization* following synchronization [14]. According to them, the important fact is the transient nature of the synchronization processes, and the fast adaptivity of the system in permanently switching from one configuration to the other. The presence of intermittencies in the dynamics of interaction seems to be a crucial point in the adaptivity of living animals.

Some remarks follow:

- From now on, there is no convincing implementation of synchronization processes for the design and control of artificial devices. The tuning of an efficient control system owning several regimes from chaos to synchrony is still a matter of projects and expectations. Its realization in realistic and plausible systems could constitute a breakthrough in the modeling and understanding of intrinsic brain dynamics.
- The question of endogenous desynchronization has not been, to our knowledge, a matter of specific interest, but the close question of the natural resetting of a recurrent short term memory model is tackled, for instance, in [70].

4 Identification models

We have seen in the previous section that the Hebbian adaptation processes drive the neuronal dynamics toward phase transitions and bifurcations. In particular, starting from disordered regimes, Hebbian-inspired weight-adaptation processes tend to reduce the complexity.

One can now ask whether such transitions may take place in the more general framework of the coupling of an agent with an environment.

We have seen in section 1.2 that two principal strategies may be used in the design and modeling of interacting agents.

- In the first one, agents own *internal models* of their environment through which they can filter and fill in noisy/ambiguous incoming patterns, and the stabilization of the action relies on the *identification* of well-known sensori-motor situation. Those situations may previously be learned with the help of a teacher (supervised learning protocol).
- In the second one, the process of movement production relies on an exploration process (trials and errors) through which the agent learns to produce the most appropriate action under its experience of previously encountered situations and associated rewards. There is no need of an explicit internal model : the action comes out of the internal self-generated dynamics, whatever shape it may have.

We present in this section the principles by which recurrent neural networks may learn to identify their environment through plausible weight adaptation processes. An efficient identifier may own some of the following properties:

- memory of past states (non-Markovian environments),
- ability to stabilize a response,
- resistance to noise and distortion,
- hidden variables and states completion,
- spatio-temporal patterns identification.

The question we address is the following: in which fashion does the principle of transition from disordered phases to ordered ones relate to the process of internal model construction, and which of the listed properties may it implement?

4.1 Principles for identification within the agent/environment interaction system

The choice of interaction system framework implies some constraints which relate to the question of the global dynamics description. The coupling of the environment and the internal dynamics gives rise to a global system (equation (1)) which may display various regimes, attraction basins and reaction times. The problem is that most of the time, one can not properly identify state variables, and thus precisely describe the trajectories and transitions taking place in the system.

At a schematic level, one can however give the following path:

- the "order" may relate to the transitions where the internal and the external processes get synchronized. In that case, the two processes are strongly coupled and easily penetrated by the other's influence.
- the "disorder" may relate to the situation where the two processes remain blind to the other's influence. They tend to produce, for instance, a chaotic pattern of interaction.

4.1.1 Coupling

can we measure whether the two sub-systems cooperate or, on the contrary, ignore each other? This question relates to the question of the *coupling*, or matching, between the two sub-processes. Such matching may be measured by the way the two sub-processes display common features in their state space, like periodicity, synchrony. The measure of the coupling between the two dynamics may empirically rely on a comparison between the embedding dimension of the global trajectory D, and the embedding dimensions of every local trajectory D_{in} and D_{out}. This point will not be developed in this paper. See for instance [71].

4.1.2 Relaxation time and action selection

How does a particular structure “take the decision” to participate to the global process? This relates to the question of action selection and decision processes. In a global system, one can not say that a certain decision is strictly taken “inside” the agent. One should better say that the environment facilitates a certain series of interaction patterns, and reciprocally the agent facilitates a certain series of interaction patterns, and the decision relies on a mutual process of convergence toward a compromise.

This time necessary to “take the decision” is the relaxation time of the global system. This means in particular that the response of the system is not instantaneous, but it may take a while before the system “makes up its mind”.

4.1.3 Teacher forcing

In the following experiments, local and unsupervised learning methods are combined with supervised “instructions” consisting in a series of perception or perception/action patterns. This method, called “Teacher forcing”, is used for instance in [72]. The patterns are sent and maintained on the system, and interact with the internal self-sustained dynamics in real time. When the learning process is activated, it participates to the reinforcement of the coupling between the external pattern and the internal dynamics. Its effect may vary from one experiment to the other. As the external drive is set regular (periodic for instance), the learning process is mainly supposed to increase the regularity *inside* the system.

This internal increase of regularity will finally change the responses of the system. After learning, the predictability of the agent’s trajectories depend on the matching between the sensori-motor prototype and the encountered situations.

4.1.4 Active/passive perception

Learning in a dynamical system may allow to model *perception* processes at a very general level. We consider perception as the dual ability:

- to dissociate the compliant part from the “unknown” (nonsense) part in the sensory flow. According to Freeman Hypothesis [5], the more the situation is well-known, the more regular is the dynamics. On the contrary, unknown situations tend to produce disordered and chaotic responses.
- to associate the compliant part of the signal with an active sensori-motor scheme. The compliant signals are indeed supposed to trigger some mechanisms which may produce some functional responses or behaviors. On the contrary, disordered regimes may maintain the system in a passive regime.

4.2 Perception and resonant memory

We give here a simple model of perception, composed of two interacting layers. It has been designed in order to testify that an efficient pattern completion mechanism can be obtained out of an on-line Hebbian process.

The principle of pattern-completion is inspired from the classical auto-associative memories [3,73]. The principle of auto-associative memories is to use the reverberant mechanisms taking place in recurrent systems to reconstruct the missing information (pattern retrieval property). A fixed number of prototypes can be learned, and the relaxation of the dynamics classifies every input pattern in one of the learned prototypes family. Such systems however learn off-line, have low capacities and do not distinguish the known from the unknown, and always give a response, which may be inappropriate.

The architecture we propose is a generic sensory structure with global analogy with auditory, visual or olfactory structures. The first layer (primary layer) is directly under the influence of a

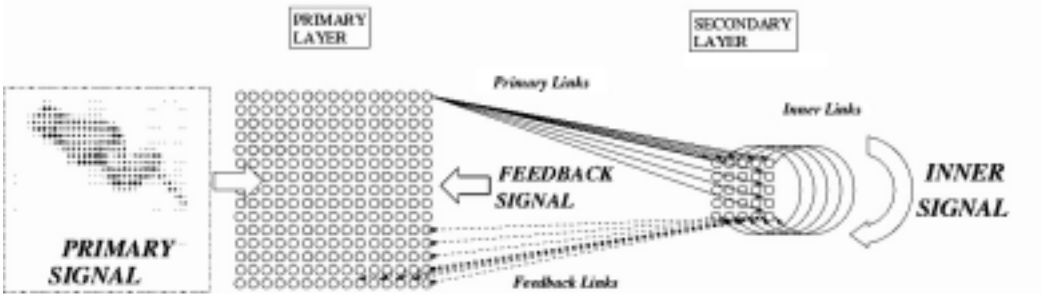

Fig. 12. Perceptive model. The model is composed of two layers of different sizes. The primary layer receives a spatio-temporal signal $\mathbf{u}^{(1)}$. The secondary layer has no input signal. The links are mono-directional. The secondary layer activity is often chaotic. the primary layer activity is both dependent on the primary signal and the feedback signal coming from the secondary layer.

sensory signal $\mathbf{u}^{(1)}$. This layer is connected to a "secondary" layer, which serves as deeper "processing" layer. The secondary layer has no direct sensory signal, and owns numerous recurrent links, so that it can maintain a self-sustained activity (see figure 12). The input signal $\mathbf{u}^{(1)}$ is composed of several patterns which activate about 5% of the primary layer neurons. Three families of links are defined: the feed forward links $\mathbf{J}^{(21)}$ which propagate the primary signal toward the secondary layer, the "internal links" $\mathbf{J}^{(22)}$ which propagate the internal dynamics, and the "feedback" links, which send back the secondary layer activity toward the primary layer (they carry the "feedback signal").

The given system can be defined with very few parameters, namely

$$\bar{J} = 0, \quad \sigma_J = \begin{pmatrix} 0 \ 0 \\ 1 \ 1 \end{pmatrix}, \quad \bar{\theta} = \begin{pmatrix} 0.5 \\ 0.4 \end{pmatrix}, \quad \sigma_\theta = 0, \quad g = 8.$$

In such a system, the focus is on the feedback links, which are expected to play the main role in the process of pattern completion. Their role is to extract out of the secondary layer some regularities which will synchronize with the current input signal. The feedback learning process starts from a blank sheet and the response progressively emerges from the coupled activity of the primary and secondary layers. Moreover, a light learning process also takes place on the internal links, which reinforces the internal regularities and stabilizes the system's response. Out of several simulation tests, we found out that the best rate between internal and feedback learning parameter is 1/5, which gives the following learning parameters:

$$\alpha = \begin{pmatrix} 0 & 0.1 \\ 0 & 0.02 \end{pmatrix}$$

The layer spontaneous dynamics is given by (14) and the learning process is order 1 covariance Hebb rule, see [32]. The primary layer is continuously stimulated. The secondary layer activity is initially chaotic and non-synchronized. At a given time, about 15% of the secondary layer neurons are active.

In our experiment (see figure 13), we use two periodic animated sequences. Each sequence is composed of k images of 1600 pixels, where k is the period of the sequence. The first animated sequence represents here a frog jump, of period 5. The second animated sequence represents a flying bird, of period 7. The choice of a visually significant pattern is set to improve the readability of our figures. The images are not orthogonal, but they are sparse, and have thus few pixels in common.

The given simulation is done on 800 time steps. The first 600 steps are devoted to the learning of sequence 1 (frog). The steps 601 to 800 test the reactivity of the system : we successively present a unknown stimulus (time 601-700) and a then a partially known one (time 701-800). The learning mechanism has to be read out of several indicators. The main indicator of the network reactivity is the amplitude and shape of the feedback signal. The feedback signal is

$$\mathbf{h}^{(12)} = \mathbf{J}^{(12)}(t-1)\mathbf{x}^{(2)}(t-1)$$

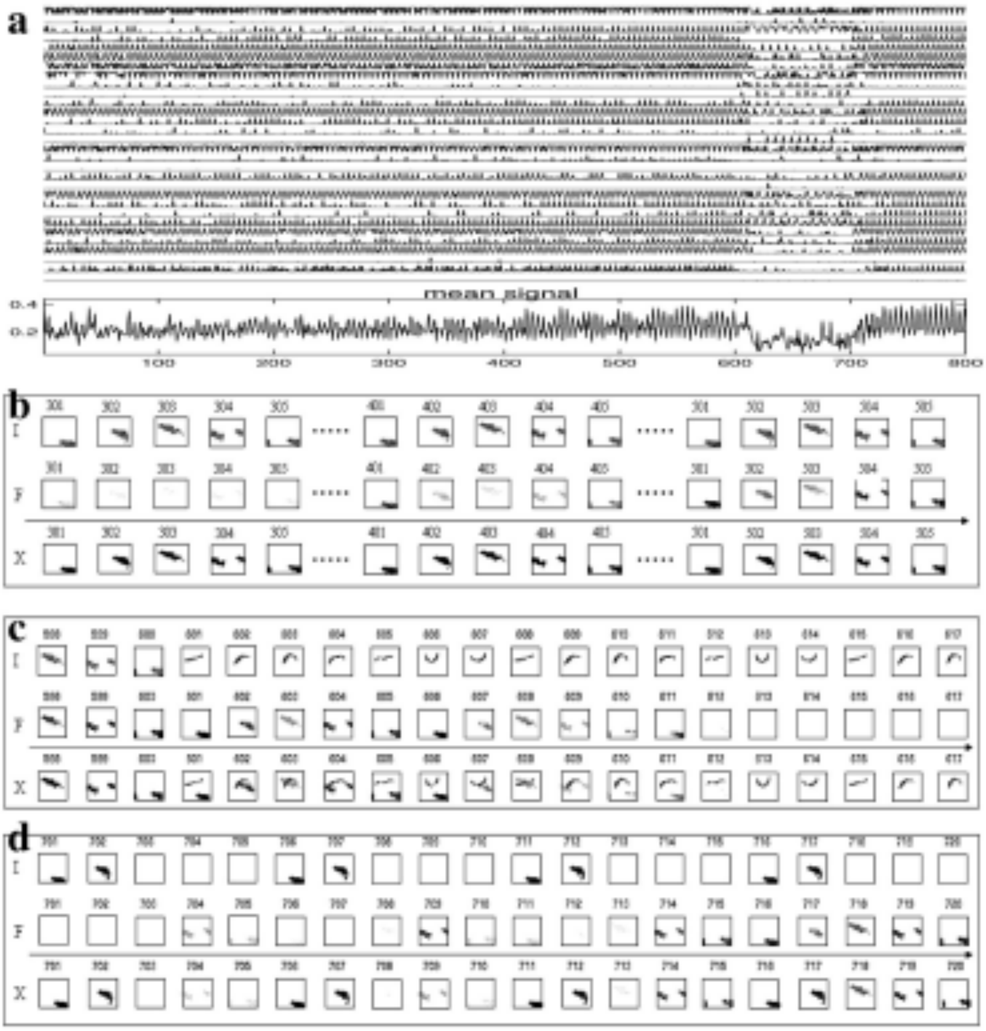

Fig. 13. Learning dynamics in a resonant model of perception (see text).

and the impact of $\mathbf{h}^{(12)}$ is measured by $\mathbf{F}(t) = f(\mathbf{h}^{(12)}, \theta^{(1)}, g)$ which displays the effective influence of the feedback signal in a practical way.

The working mechanism of pattern recognition and completion relies on the reverberation between layer 1 and layer 2, which is coupled with a light process of regularization taking place on layer 2. If we first consider the activity of layer 2 (figure 13 - a -) during the first 600 time steps, we can see that the dynamics becomes more and more regular as time goes on. This regularizing effect of the learning process is coherent with the previous experiments in various models, see for instance figure 10. Now, the reverberant mechanism is illustrated by figure 13 - b -. The figure is composed of several 40 × 40 images. Each line gives five steps samples of the time evolution of three spatio-temporal signals between $t = 301$ and $t = 505$, namely

- the primary input sequence $\mathbf{u}(t)$,
- the feedback signal $\mathbf{F}(t)$,
- the primary layer activity $\mathbf{x}^{(1)}(t)$, which is the sum of the two previous signals.

Starting from zero, the amplitude of the feedback links (positive or negative) slowly increases, finally giving rise to a sustained feedback signal. Up to $t = 300$, the feedback signal remains very weak. From $t = 400$, the feedback signal becomes strong enough to have effect on the primary neurons activity. This signal gets reinforced until $t = 600$ (input change). During this learning phase, the secondary signal is correlated in space and time with the primary signal (a "copy" of the incoming signal is produced out of the secondary layer activity). The system has entered in a resonant mode where the feedback signal is as strong as the primary signal, possibly modifying or enriching the observation vector.

The testing phase is displayed on figures 13 - c - and 13 - d -. At $t = 601$, the bird sequence is presented for 100 time steps (which is not enough for the learning process to leave a significant imprint). Figure 13- a - displays the transition taking place in the internal dynamics. Between $t = 601$ and $t = 700$, the secondary layer activity is strongly modified, globally weaker and more disordered. The figure 13- c - gives the first layer primary and feedback signals, between $t = 598$ et $t = 617$. It illustrates the relaxation time of the transition, of the order of 15 time steps, which is necessary for the system to switch from the resonant mode to the disordered one. The principal observation is the mismatch (and concurrence) on primary layer between the primary and feedback signals. The feedback signal does not fit any more with the incoming signal. After a while, the feedback signal extincts for it is no more fed by the secondary layer dynamics. The persistence of the feedback signal during about 10 time steps illustrates the stability of the learned answer : only the repetition of feedback error on several time steps allows for the dynamics to change and a new regime to be attained.

Finally, at $t = 701$, a meaningful sensory signal is presented, composed of the two first images of the learned sequence, followed by a three steps blank input (the whole signal being repeated, giving a period 5 stimulation). Figure 13- a - shows that the internal layer recovers its initial activity, with (almost) the same internal activation patterns. Even different from the learned sequence, this sequence is perceived as coherent with the imprinted sequence. This gives rise to a similar attractor which finally participates to the completion of the missing part of the signal. Figure 13- d - gives a zoom on the transition. The feedback signal is progressively enhanced until the full sequence can be recalled on the primary layer. At last, the system behaves "as if" the full sequence was present.

This simulation gives some insights on the intrinsic melting between perception and memory recall. An elusive signal with partial correspondence with a past experiment *resonates* with the internal dynamics, which finally fully evokes the past experiment. The presented model, even rough and simple in comparison with the real brain, suggests that memory recall and direct perception rely on the same mechanism, and are thus intimately linked. Under this approach, a perception system only perceives what it has been prepared to perceive. The perception is active, as the internal self-sustained activity permanently "simulates" its current environment, and this simulation is successively consonant or dissonant with the incoming signal. The system is able to perceive well what it has been prepared to perceive, in accordance with its internal model, this model being progressively constructed through its experiment.

5 Control models

We have seen previous sections that knowledge/skill acquisition/recall is often associated with a reduction of the dynamics complexity, taking place at the level of the neurons activity and/or at the behavioral level. Apart from uncertainty reduction, learning and evolution processes are also supposed to be oriented toward a better access to the sources of nutriments and a better protection against environment hazards. This point is more delicate to tackle under a dynamical systems approach. It is the core of the embodied approach. Under that approach, an agent identifies with its body, and the cognitive activity identifies with the continuous trade-off between the dynamics of self-construction and the body/environment structural couplings [74].

5.1 Coupling with environment dynamics

The previous simulations have shown that perception is rooted on memory. We see now how the given principles (dynamics reduction, resonance, active vs. passive behaviors) can be extended to the more general case of movement production. The following experiment is designed in order to study an explicit coupling between the internal dynamics and the body/environment dynamics (also called behavioral dynamics). The signal $\mathbf{u}_{\text{in}}$ that comes from the environment is now fully dependent on the agent's movements. Our experiments are now said to take place under a "closed loop" approach. A similar study addressing the question of the dual dynamics between a recurrent neural network and a robot trajectory can be found in [72].

5.1.1 A simple sensori-motor system

We present here a system which has many common points with the previous one: a recurrent secondary layer is coupled with two perceptive layers, the first perceptive layer displays visual signals, the second perceptive layer displays proprioceptive signals.

In this experiment, a wheeled robot is assigned to learn some associations between its visual environment and its movements. The network owns $P = 3$ populations, i.e. two primary layer and one secondary layer. The parameters are the same as in previous experiment. The robot movements are limited to rotations, from $-90°$ to $+90°$ by 30° steps. The visual signal is adapted from a pre-visual treatment (salient points extraction). A periodic movement is sent on the proprioceptive layer, which corresponds to a loop composed of three rotations $(+30°, +60°, +90°)$, so that 6 time steps are needed for the robot to resume its initial visual field (see figure 14). The visual signal $\mathbf{u}^{(1)}$ is thus found to have a period of 6.

After a short learning session of 40 time steps, the accuracy of the robot angular position is checked. The learning session is not long enough to produce a sustained feedback signal, but the feedback signal is strong enough to extract a motor command out of the internal layer since the motor instructions are removed.

5.1.2 Alternating behavior

During the testing phase the production of movement is autonomous. The system has thus the "choice" between one of the three possible motor responses: +30°,+60° or +90°. This choice is extracted from the feedback signal. Two typical angular trajectories are given on figure 15.

The robot is placed with an arbitrary initial angular position. After short transitory, the robot robot locks on a matching visual scene and maintains its learned periodic movement according to its previous learning session.

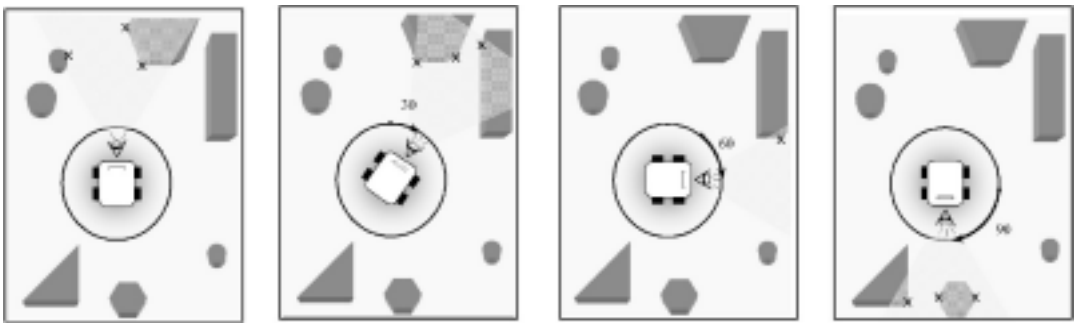

Fig. 14. Successive positions of the robot issuing from the 3 motor commands +30°,+60°,+90°. The associations between salient visual points and their angular position is the visual input. After 3 steps, the robot has made an half-turn. (issuing these commands again allows the robot to resume its initial position).

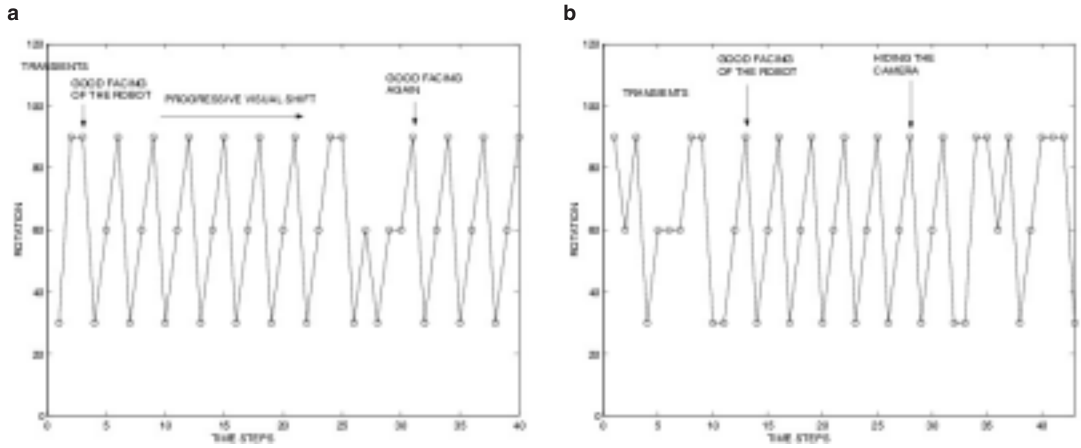

Fig. 15. Angular trajectories of the robot as a function of time. **- a -** A visual shift gives rise to a transitory phase of erratic movements (see text). **- b -** After the robot has reached his periodic phase, the visual signal is explicitly masked, which gives rise to an erratic phase (see text).

In the first experiment (figure 15 -a-), the learned sequence is reproduced during about 20 time steps (between $t = 3$ et $t = 24$), and then the system unlocks and starts erratic movements. This discontinuity relates to the unexpected friction of the wheels, which lead to an under-estimation of the real rotation. The real angular position undergoes a slow drift, and the visual field undergoes the same drift. The visual input becomes less and less consistent with the learned sequence. this lasting conflict between movement and vision leads to *sudden change* in the robot's behavior, i.e. a transition. The robot is indeed unable to make instant correction of its angular position: it is "lost", and enters a disordered phase, which produces erratic movements which have no correspondence with the learned sequences. After those new transients, the robot rallies a matching position from which it can resume its periodic behavior. In the second experiment (figure 15 -b-), the visual signal is explicitly masked after the robot has reached its periodic behavior. This change produces a transition like in previous experiment, with comparable erratic movements, which persist as soon as the camera is masked.

These experiments, even simplistic, reveal that *phase transitions* can be observed *in the behavioral domain.*

- The periodic behavior is the active *task*, which associates in a regular way visual and proprioceptive signals. This behavior is stable on a large scope of the visual field, included strongly drifted visual signals.
- The passive erratic movement is analogous to an *exploratory* behavior: the series of motor commands implicitly seek for a matching sequence. When this research can not succeed, the system remains in its erratic phase.

The intermittency between ordered and disordered phases illustrates the degree of coupling between the internal dynamics and the issued movements. A strong coupling between the internal dynamics and the environment tends to produce predictable movements, and a weak coupling leads to less predictable behaviors. A strong coupling will reduce the agent's autonomy, while a weak coupling will increases the agent autonomy according to its environment.

This experiment also helps to rethink the phenomenon of memory recall. The memory recall is indeed embedded in a large process which overtakes the agent. The recall only takes place when the internal and the external dynamics "meet one another". A joint tendency is needed for the recall to emerge for the action concretely take place in the agent's world.

5.2 Learning with rewards

5.2.1 Biologically plausible reward learning

Apart from the classical reinforcement learning paradigms, which have been presented in section 2, a lot of models have been proposed for the modeling of biologically plausible reinforcement processes [75,76]. It is frequently admitted that neurotransmitters release (dopamine, GABA, . . .) can be interpreted as *valuations* of the current processes taking place in the brain. Many studies suggest that some reinforcement mechanism may take place at the level of the basal ganglia in relation with dopaminergic neurons. The processes and paths through which neurotransmitters release is achieved are much less known (and at least as complex and finely tuned) than the axonal action potential transmission processes. They may participate, for instance, to the hippocampus memory imprint processes.

A lot of actor-critic models have been proposed in the recent years, most of which being inconsistent with the anatomy of the basal ganglia [76]. The main problems with such "high-level" models is the lack of knowledge of the real anatomy of the implied structures. It can be noticed, for instance, that some forms of operant learning have been shown to take place in simple invertebrate animals [75]. Reinforcement is suggested to be one of the most primitive nervous adaptation mechanism, though it doesn't need any explicit model or consign.

The consequences in modeling are the following:

- Apart from the activation and learning mechanisms, there must be an independent *reward mechanism* which fixes the moment when a reward is emitted. The moment of neurotransmitter release must rely on simple and perceptible environmental clues.
- In the general case, there must be positive and negative rewards, possibly corresponding to different neurotransmitters releases. they can be interpreted in several ways:
  - positive reward may reinforce the activity of the excitatory neurons while negative rewards may reinforce the activity of inhibitory neurons.
  - positive reward may correspond to a Hebbian plasticity, while negative rewards may correspond to a anti-Hebbian plasticity.
  - positive or negative rewards may have a restricted effect on some specific categories of synapses. Some synapses may be sensitive to the positive rewards, other to the negative ones.
  - at last, there may be more than only two categories (positive/negative), but also layer specific neurotransmitters. This point is not tackled here. . .
- The *value* of the rewards may correspond to the amount of neurotransmitters release.

5.2.2 Reinforcement learning with a recurrent neural network model

There is still a gap between classical TD and Q-learning methods and more biologically-relevant approaches of reinforcement learning. For that, the simulations of realistic reward learning processes may be helpful for a better understanding of the reinforcement processes taking place in the brain.

The method we use is based on the following approach:

- The movements production and exploratory processes relies on the self-generated chaotic activity.
- Learning is the selection process through which the better configurations are to be stabilized.
- Learning is based on punctual applications of positive and negative Hebbian rules. The positive Hebbian rule relates to positive rewards, the negative Hebbian rules relate to negative rewards.

The 2-population networks we have described in previous sections are now combined in order to build a perception/action network. In this implementation, we use a simple McCulloch and Pitts model composed of a sensory layer and a movement production layer, each layer owning distinct local excitatory and inhibitory neurons.

A functional module is composed of 2 populations of neurons : one excitatory population and one inhibitory population. The global parameters are the following :

$$\bar{J} = \frac{1}{2}\begin{pmatrix} 1 & -k & 0 & 0 \\ k & -1 & 1 & 0 \\ 1 & 0 & 1 & -k \\ 0 & 0 & k & -1 \end{pmatrix}, \sigma_J = \frac{1}{2d}\begin{pmatrix} 1 & \sqrt{k} & 0 & 0 \\ \sqrt{k} & 1 & 1 & 0 \\ 1 & 0 & 1 & \sqrt{k} \\ 0 & 0 & \sqrt{k} & 1 \end{pmatrix}, \bar{\theta} = \begin{pmatrix} 0.1 \\ 0.1k \\ 0.1 \\ 0.1k \end{pmatrix}, d = 6, k = 4.$$

As previously said, every module can be seen as a rough approximation of a cortical column. In this experiment, the modules are topologically organized. The connections strength are a function of the distance. A full description of topologically structured modules can be found in [77].

The two modules are comparable to the ones described in paragraph 3.1.2. One can moreover remark that the sensory module feeds the movement production module, while the movement production module tends to inhibit its perception. This choice relates to the hypothesis of temporary blindness during the realization of the movements, which favors the arrival of new percepts. The “attention” thus permanently switches from perception to action and from action to perception (see also paper 4 [81]).

Designing a perception-action network also means to specify the environment through which the system interacts, and also to specify a task. In order to minimize the environment complexity, our choice is put on a very well known and documented task: the control of an inverted pendulum.

The production of action relies on the second module. The force applied to the pendulum is dependent on the position of the peak of activity on the module excitatory layer, taking place between $-20\,\mathrm{N}$ and $+20\,\mathrm{N}$.

Our aim is to validate the mechanism of positive/negative Hebbian rule alternation. The reinforcement signal relies on the pendulum velocity: a low velocity ($<0.05\,\mathrm{m/s}$) triggers a positive reward, while a high velocity ($>0.5\,\mathrm{m/s}$) triggers a negative reward. The network is thus implicitly assigned to maintain the pendulum velocity as small as possible. Positive reward activate a positive Hebbian rule *on the excitatory links only*, and negative rewards activate a anti-Hebbian rule on the same links. The Hebbian rule has been balanced in order to avoid weights drift (see details in [78]). In accordance with [27], a trace of the most recent Hebbian terms is stored at the level of the synapses, but it is not directly imprinted on the weights (see also equation (4)). The arrival of a reward (positive or negative) activates the imprint mechanism, and modifies the weights in proportion with the reward value. Moreover, the rewards are set to be rare events (sparse rewards): we only allow a positive reward after a negative one has been sent, so that the system is sensitive to the *novelty* of its situation. The mean reward expectation is thus of the order of 0.

This principle has been tested on several networks, the learning is done on-line and the maximal number of time steps is limited to 1000 (which corresponds to a 5 seconds control), in order to experience various situations and various networks. Moreover, the simulation is stopped as soon as the pendulum angle is too large. The network progress is measured by the mean control duration. If this mean duration is close to 1000, the system is found to maintain the pendulum in equilibrium for almost every initial condition.

Figure 16 gives the mean control duration as a function of the number of positive rewards, out of 23 networks realization and 400 control attempts through the on-line learning process. We clearly obtain a strong improvement of the control capacity for most of the networks and most of the situations. This capacity is moreover maintained in the long term without deterioration or saturation (thanks to the weight drift limitation).

So, what mechanism are efficient in that case? On the contrary to standard reinforcement methods, the evaluation and action production processes are intimately linked (there is no explicit evaluation process). The *selection* of a proper action out of several possible responses relies on the versatility of the system. As the system spontaneously displays a great variety of responses, the arising of a reward at a given time helps to favor a particular response out of a set of possible responses. The choice is however limited, and the behavior of the system is not necessarily the optimal behavior, but only a *viable* one. There is no explicit prediction of the

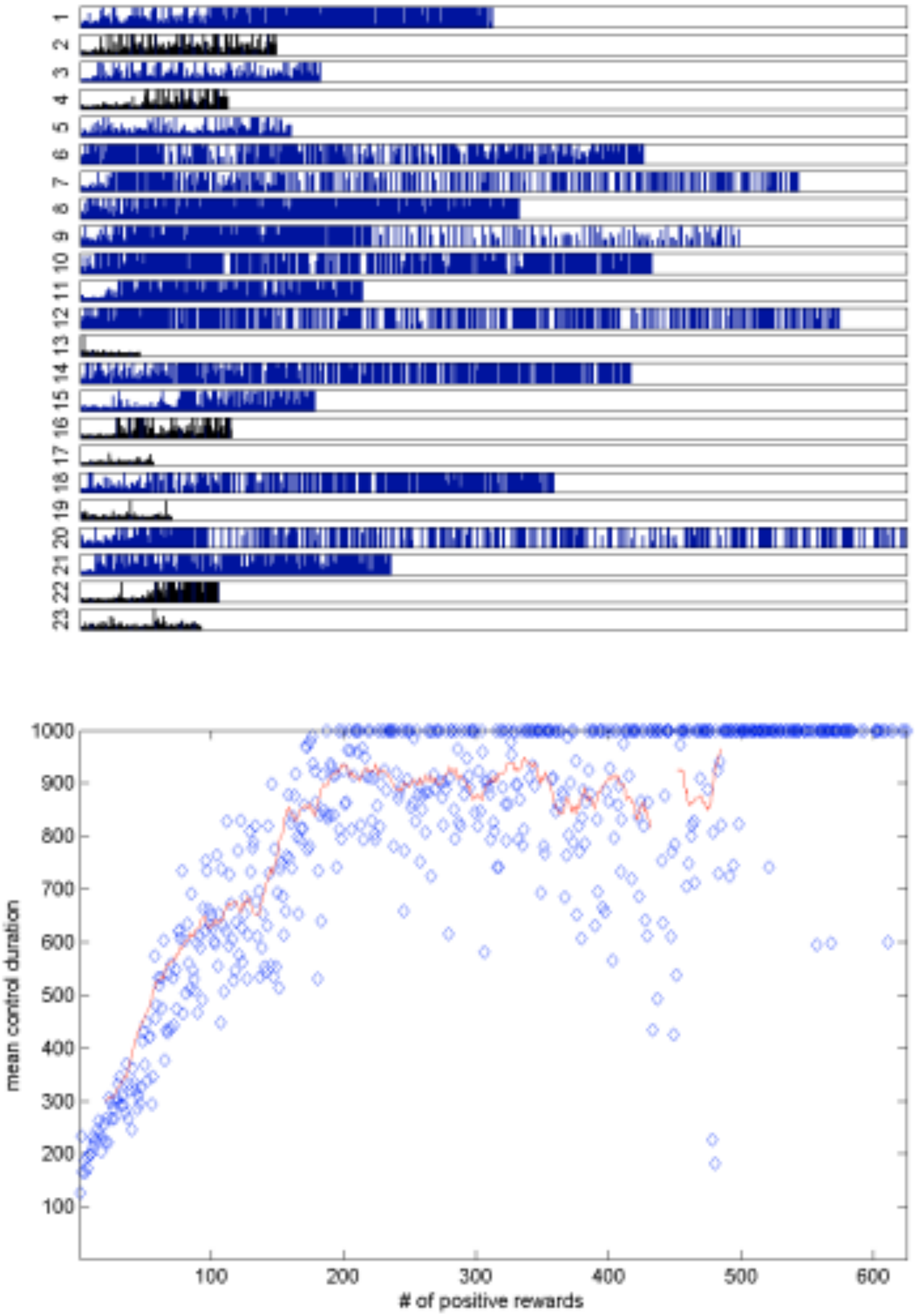

Fig. 16. Measure of the control capacity of 23 networks for an inverted pendulum task. The upper figure gives the evolution of the control duration for every network. The lower figure gives the mean control duration out of the 23 networks as a function of the number of positive rewards.

future rewards, but the system tendency is to learn the best choice for every local situation; it results in an improvement of its global behavior.

In biological terms, it is shown that an alternation of Hebbian and anti-Hebbian rules on the excitatory links allows to shape the behavior. It mostly relies on a selective excitation (or depression) of some neurons of the action production layer (Central Pattern Generator - CPG -), giving rise to a more selective response. The question remains wether such Hebbian/anti Hebbian may take place on real neurons. They may be triggered by the release of different neurotransmitters for instance dopamin and serotonin. This numerical experiment is thus a first step toward more realistic and biologically founded models.

Conclusion

This paper has shown that a combination of recurrent networks with simple Hebbian learning rules can provide efficient and biologically plausible mechanisms of knowledge acquisition. Freeman's hypothesis [5] on the role of phase transition in recognition can be implemented in an efficient way. It has been moreover shown that the agent's behaviors shape under the constraints of their environment, giving rise to a procedural knowledge which manifests in more regular trajectories. After learning, two sorts of behaviors can be identified, namely the ordered behaviors and the disordered ones. The ordered behaviors may relate to an *active* implication of the agent within its environment, while disordered ones may relate to a more *passive* attitude, possibly interpreted as an exploratory process. The transitions between various behaviors obey to a principle of locking and unlocking: the agent tends to persist in its current behavior until the mismatch is “felt” as unbearable.

Most of the presented mechanisms are found throughout various models and learning methods, provided that spike timing is finely taken into account. The models are most of the time simple in their realization, and generic in their capabilities. They would for instance ultimately unify sequence learning, identification and reinforcement learning. Even if they do not attain the optimal response, their versatility may allow for a better adaptivity. This presentation thus claims for a better account of recurrent models and local synaptic mechanisms for the modeling of learning and skill acquisition in biology and control.

References

1. S. Grossberg, Proc. Natl. Acad. Sci. USA **59**, 368–371 (1968)
2. S. Amari, IEEE Trans. Syst. Man. Cyb. **SMC-2**, 643–657 (1972)
3. J. Hopfield, Proc. Nat. Acad. Sci. **79**, 2554–2558 (1982)
4. G.A. Carpenter, S. Grossberg, Comput. Vis. Graph. Image Process. **37**, 54–115 (1987)
5. C. Skarda, W. Freeman, Behav. Brain Sci. **10**, 161–195 (1987)
6. M.A. Cohen, S. Grossberg, IEEE Trans. Syst. Man. Cybern. **SMC13**, 815–826 (1983)
7. S. Amari, IEEE Trans. Comp. **C-21**, 1197–1206 (1972)
8. H. Sompolinsky, I. Kanter, Phys. Rev. Lett. USA **57**, 2861–2864 (1986)
9. S. Dehaene, J.-P. Changeux, J.-P. Nadal, Proc. Natl. Acad. Sci. USA **84**, 2727–2731 (1987)
10. F. Delcomyn, Science **210**, 492–498 (1980)
11. P. Meyrand, J. Simmers, M. Moulins, Nature **351**, 60–63 (1991)
12. R. Rao, T. Sejnowski, Neural Comput. **13**, 2221–2237 (2001)
13. C. Leibold, R. Kempter, Neural Comput. **18**, 904–941 (2006)
14. E. Rodriguez, N. George, J.-P. Lachaux, J. Martinerie, B. Renault, F. Varela, Nature **397**, 430–433 (1999)
15. B. Ans, Y. Coiton, J.-C. Gilhodes, J.-L. Velay, Neural Netwo. **7**, 1461–1476 (1994)
16. E. Bicho, G. Schöner, Robot. Auton. Syst. **21**, 23–35 (1997)
17. S. Haykin, *Neural Networks: A Comprehensive Fundation* (Prentice-Hall, 1999)
18. G. Dreyfus, J.-M. Martinez, M. Samuelides, M.B. Gordon, F. Badran, S. Thiria, L. Hérault, *Réseaux de neurones - Méthodologie et applications* (Eyrolles, Paris, 2002)
19. M. Ghallab, D. Nau, P. Traverso, *Automated Planning, Theory and Practice* (Elsevier, Morgan-Kaufmann, 2004)
20. J.J. Gibson, *The Ecological Approach to Visual Perception* (Houghton-Mifflin, Boston, 1979)
21. R. Brooks, Artific. Intell. **47**, 139–159 (1991)
22. K. O'Regan, G. Noë, Behav. Brain Sci. (2001)
23. R. Sutton, Mach. Learn. **3**, 9–44 (1988)

24. K. Doya, Neural Comput. **12**, 219–245 (2000)
25. C. Watkins, P. Dayan, Mach. Learn. **8**, 279–292 (1992)
26. R.J. Williams, Mach. Learn. **8**, 229–256 (1992)
27. P. Bartlett, J. Baxter, Hebbian synaptic modifications in spiking neurons that learn, School of Information Science and Engineering, Australian National University, Tech. Rep. (1999)
28. H. Tuckwell, *Introduction to Theoretical Neurobiology* (Cambridge University Press, 1988)
29. A. Herz, B. Sulzer, R. Kuhn, J.L. van Hemmen, Biol. Cybern. **60**, 457–467 (1989)
30. D. Hebb, *The Organization of Behavior* (Wiley, New York, 1949)
31. G. Turrigiano, K. Leslie, N. Desai, L. Rutherford, S. Nelson, Nature **391**, 892–896 (1998)
32. T.J. Sejnowski, J. Math. Biol. **4**, 303–321 (1977)
33. T.V.P. Bliss, T. Lomo, J. Physiol. **232**, 331–356 (1973)
34. H. Markram, J. Lubke, M. Frotscher, B. Sakmann, Science **275**, 213–215 (1997)
35. G.-Q. Bi, M.-M. Poo, J. Neurosci. **18**, 10464–10472 (1998)
36. S. Song, K. Miller, L. Abbott, Nat. Neurosci. **3**, 919–926 (2000)
37. H. Sompolinsky, A. Crisanti, H. Sommers, Phys. Rev. Lett. **61**, 259–262 (1988)
38. B. Cessac, J. Phys. I (France) **5**, 409–432 (1995)
39. B. Doyon, B. Cessac, M. Quoy, M. Samuelides, Int. J. Bif. Chaos **3**, 279–291 (1993)
40. S. Grossberg, Biol. Cybern. **23**, 121–134, 196–202 (1976)
41. S. Amari, Biol. Cybern. **27**, 77–87 (1977)
42. T. Kohonen, Biol. Cybern. **43**, 59–69 (1982)
43. G. Schöner, M. Dose, C. Engels, Robot. Auton. Syst. **16**, 213–245 (1995)
44. S. Moga, P. Gaussier, A neuronal structure for learning by imitation, in *Lecture Notes in Artificial Intelligence – European Conference on Artificial Life*, edited by D. Floreano, J.-D. Nicoud, F. Mondada (Lausanne, 1999), pp. 314–318
45. S. Funahashi, C.J. Bruce, P.S. Goldman-Rakic, J. Neurophysiol. **61**, 331–349 (1989)
46. R. Ben-Yishai, R. Lev Bar-Or, H. Sompolinsky, Proc. Natl. Acad. Sci. USA **92**, 3844–3848 (1995)
47. R. Ben-Yishai, D. Hansel, H. Sompolinsky, J. Comput. Neurosci. **4**, 57–79 (1997)
48. D. Hansel, H. Sompolinsky, J. Comput. Neurosci. **3**, 7–34 (1996)
49. M. Camperi, X.-J. Wang, J. Comput. Neurosci. **5**, 383–405 (1998)
50. Wang, J. Neurosci. **19**, 9587–9603 (1999)
51. A. Compte, N. Brunel, P.S. Goldman-Rakic, X.-J. Wang, Cerebr. Cortex **10**, 910–923 (2000)
52. A.D.E. Guillot, *Approche Dynamique de la Cognition Artificielle* (Lavoisier, 2002)
53. M. Adachi, A. Kazuyuki, Neural Netw. **10**, 83–98 (1997)
54. I. Tsuda, Neural Netw. **5**, 313–326 (1992)
55. U. Riedel, R. Kuhn, J. Van Hemmen, Phys. Rev. A **38**, 1105–1108 (1988)
56. K. Ntzel, J. Kien, K. Bauer, J. Altman, U. Krey, Biol. Cybern. **70**, 553–561 (1994)
57. O. Hoshino, Y. Kashimori, T. Kambara, Biol. Cybern. **79**, 109–120 (1998)
58. S. Amari, Kybernetik **14**, 201–215 (1974)
59. W. Gerstner, R. Ritz, L. Van Hemmen, Biol. Cybern. **68**, 363–374 (1993)
60. N. Brunel, V. Hakim, Neural Comput. **11**, 1621–1676 (1999)
61. E. Daucé, O. Moynot, O. Pinaud, M. Samuelides, Neural Proc. Lett. **14**, 115–126 (2001)
62. I Ginzburg, H. Sompolinsky, Phys. Rev. E. **50**, 3171–3191 (1994)
63. O. Moynot, M. Samuelides, PTRF **123**, 41–75 (2002)
64. C. Van Vreeswijk, H. Sompolinsky, Neural Comput. **10**, 1321–1371 (1998)
65. E. Daucé, O. Moynot, O. Pinaud, M. Samuelides, B. Doyon, Mean field equations reveal synchronization in a 2-populations neural network model, in *ESANN 99*, edited by M. Verleysen (D-Facto, 1999), pp. 7–12
66. P. Bush, T. Sejnowski, J. Comput. Neurosci. **3**, 91–110 (1996)
67. E.M. Izhikevich, IEEE Trans. Neural Netw. **14**, 1569–1572 (2003)
68. E. Daucé, M. Quoy, B. Cessac, B. Doyon, M. Samuelides, Neural Networks **11**, 521–533 (1998)
69. W. Singer, Time as coding space in neocortical processing: a hypothesis, in *Temporal Coding in the Brain*, edited by G. Buzsáki (Springer-Verlag, Berlin, Heidelberg, 1994), pp. 51–79
70. N. Brunel, X.-J. Wang, J. Comput. Neurosci. **11**, 63–85 (2001)
71. A. Penn, Steps towards a quantitative analysis of individuality and its maintenance: a case study with multi-agent systems, in *Fifth German Workshop on Artificial Life: Abstracting and Synthesizing the Principles of Living Systems*, edited by D. Polani, J. Kim, T. Martinez (IOS Press, 2002), pp. 125–134

72. J. Tani, Model-based learning for mobile robot navigation from the dynamical system perspective, IEEE Trans. System, Man and Cybern. B **26**, 421–436 (1996)
73. B. Kosko, Bidirectional associative memories, IEEE Trans. Systems, Man Cybern. **18**, 49–60 (1988)
74. F. Varela, Principles of Biological Autonomy (North Holland, Amsterdam, 1979)
75. B. Brembs, F. Lorenzetti, F. Reyes, D. Baxter, J. Byrne, Science **296**, 1706–1709 (2002)
76. J. Daphna, Y. Niv, E. Ruppin, Neural Networks **15**, 535–547 (2002)
77. E. Daucé, Nat. Comp. **2**, 135–157 (2004)
78. E. Daucé, Hebbian reinforcement learning in a modular dynamic network, in *Proceedings of the Eighth International Conference on Simulation of Adaptive Behavior (SAB'04)* (2004), pp. 305–314
79. B. Cessac, M. Samuelides, Eur. Phys. J. Special Topics **142**, 7–88 (2007)
80. M. Samuelides, B. Cessac, Eur. Phys. J. Special Topics **142**, 89–122 (2007)
81. L. Perrinet, Eur. Phys. J. Special Topics **142**, 163–225 (2007)