Random recurrent neural networks for autonomous system design

Emmanuel Daucé[*] **Mathias Quoy**[**]

[*]DTIM, ONERA - Centre de Toulouse
2,avenue Edouard Belin
31055 Toulouse Cedex, France
dauce@cert.fr

[**]Neurocybernetics Team, ETIS, UCP-ENSEA
6, Av. du ponceau
95014 Cergy Pontoise Cedex, France
quoy@u-cergy.fr

Abstract

In this article, we stress the need for using dynamical systems properties in autonomous architecture design. We first study the dynamics of random recurrent neural networks (RRNN). Such systems are known to spontaneously exhibits various dynamical regimes, as they always tries to remain on an attractor, thus achieving stable dynamical behaviors. Second, we try to characterize the adaptive properties of such a system in an open environment, i.e. in a system which always interacts with external signals.Under these conditions, a change in the behavior corresponds to the switch from one attractor to another one. Such bifurcation occur for very little changes in the environment signal; our system is thus *unstable* on its inputs. We propose a local Hebbian learning rule which tends to stabilize the response of the system for given inputs. After training, the system is able to perform *recognition*, i.e to produce a specific regular cyclic attractor while the learned input is present (or even a noisy version of this learned input). Moreover, our system can make **associations** while learning process takes place under two "sensory" influences. The system can indeed perform recognition, even when one sensory signal is missing. Our RRNN is then implemented on a robotic system, under visual and sensori-motor influences. After learning periodic motor sequences in association with visual inputs, our system can now discriminate between matching and unknown visual sequences. When visual sequence matches inner sequence, the system produces regular periodic movements. On the contrary, when there is a conflict between visual inputs and inner dynamics, the system tends to produce chaotic aperiodic movements. Our work finally illustrate a very general paradigm on cognitive aspects of perception : what the system perceives depends both on input signal and inner expectations on such input.

1. Introduction

Autonomous systems have to adapt and react to a rapidly changing environment. Moreover the environment may be unknown and unpredictable. So the design of such systems may not stand alone on prewired reactions. However, some regular patterns may be found in the environment dynamics. These patterns may be the basis for sensory-motor learning and temporal sequence learning. So an autonomous system dealing with such an environment has to achieve the following properties:

- stable behaviors, though the inputs are changing
- if necessary, rapid change in behaviors
- learning new appropriate behaviors

Animals are particularly well adapted autonomous systems. So one of our source of inspiration is to capture the relevant necessary informations that make animals behave autonomously. Neurobiological research can become a source of inspiration for people who work in the conception of intelligent and autonomous systems ("animat" approach (Meyer and Wilson, 1991)). Our first source of biological inspiration comes from the pioneer "dynamical" approach to cognition performed by Freeman on the olfactory bulb of the rabbit (Skarda and Freeman, 1987, Yao and Freeman, 1990). He has shown that natural attentive waiting states correspond to chaotic dynamics, and that presentation of a known odor leads through a bifurcation to almost cyclic dynamics. Even local and specific to the olfactory bulb, these results may indicate relevance of taking inspiration from dynamical systems theory for the analysis of brain processing. Indeed, recently important progress has been made in the precision of brain signal capture, and new phenomenons are observed which stress the role of global distant correlations in brain computation. As the subject or animal carries out a cognitive act (recognition, action selection ...), one can observe global spatio-temporal patterns of activation emerging from background activity. Such patterns have a very short life (of the order of tenths of milliseconds) and their extinction leads to the emergence of new patterns (Mac Leod and Laurent, 1996, Neuenschwander et al., 1996, Jirsa et al., 1998). These transitions from one stable behavior to another can be seen as bifurcations, which can either occur through interactions with the environment or through inner dynamical constraints.

So we use artificial neural networks in order to design our control architectures. The main stream of connectionnist methods, derived from Hopfield networks, feed-forward networks and Kohonen maps, produce static signals as their input values remain constant. Such static behaviors never occur in real brain activity. A more "biologically inspired" approach which take into account a more precise modeling of a neuron (latencies of discharge and discrete pulses for instance) may lead to more complex dynamics. In this article, we do not claim such a local biological precision. Indeed, we are mainly concerned with simplifying the biological complexity in order to exhibit the simple control variables of the system. So, we start with a model of very simple analog neurons, and study some dynamical properties of such networks. Then, with the help of a learning rule, we deal with inner and outer dynamics in order to produce dynamically relevant acts of perception and recognition. The idea is to show that systems that spontaneously exhibit several dynamical regimes with different stimulations can be of good help for the conception of agents that have to interact with the real world, and can also give clues for the comprehension of brain computation. Then, the system that has to be taken into account is the inner dynamics *plus* the environment inputs (open systems). Hence the emerging attractor does not only correspond to the inner state of the system, but is a combination of the inner dynamics and the inputs. So, in our system, a "cognitive act" should both depend on the input (command) and on inner dynamical constraints.

Spontaneous dynamical behaviors occur under certain conditions in recurrent neural networks (RNN). Lots of recurrent models exhibit cyclic sequential behaviors when properly configurated (Hertz and Prugel-Bennett, 1996) and can learn from observation some characteristics of a given dynamical system (Williams and Zipser, 1989, Tani and Fukumura, 1995). Some works have yet applied RNN to robotic control. Tani (Tani and Nolfi, 1998) used a hierarchy of RNN for categorizing different sensory-motor situation. With a different approach, Schöner (Schöner et al., 1995, Bicho and Schöner, 1997) uses the Neural Field formalism (Amari, 1977) for controlling a mobile robot. In its system, the attractor is always a fixed point. The system goes continuously from one attractor to the other through bifurcations depending on the change of external inputs.
In this article, we first present the RRNN model used, in particular, the various dynamical regimes it may exhibit, and how they may be related to external inputs. Then we propose a local Hebbian learning rule stabilizing stimulus-associated dynamical patterns. Finally, we apply our model to a simple control task on a mobile robot.

2. Basic properties of the model

2.1 A random recurrent model

Our dynamical system (1) is defined as a pool of N interacting units, whose state is described with an activation vector $\mathbf{x}(t)$. The external world is represented by the input vector signal $\mathbf{I}(t)$. This system is a discrete-time neural network, with random recurrent connections and analog neurons. Random neural networks have been introduced by Amari (Amari, 1972) in a study of their large size properties. Our model keeps the global recurrent architecture of Hopfield networks (Hopfield, 1982); the main difference stands on the fact that the initial weights are randomly chosen.

$$x_i(t) = f_g\left(-\theta + I_i(t) + \sum_{j=1}^{N} J_{ij} x_j(t-1)\right) \qquad (1)$$

The J_{ij}'s are the synaptic weights, and θ is the activation threshold. The J_{ij}'s values are randomly defined at the creation of the system, with Gaussian laws $\mathcal{N}(0, 1/N)$. This means in particular that our connection pattern is *not symmetric* , which is a necessary condition for developing complex dynamics. We take a monotonic non-linear sigmoidal transfer function $f_g(u) = (1 + \tanh(gu))/2$, whose gain is $g/2$. This function takes its values in $]0, 1[$. Notice that our system is deterministic as soon as the input signal does not include noise.
The dynamics can be characterized by the observable $m_N(t)$ defined as the mean signal of $\mathbf{x}(t)$:

$$m_N(t) = \frac{1}{N}\sum_{i=1}^{N} x_i(t) \qquad (2)$$

2.2 Autonomous dynamics and attractors

If the external signal $\mathbf{I}(t)$ is static (does not change with time), the system is called *autonomous* and its dynamical activity only stands on inner interactions. Our system is dissipative, so that any trajectory tends to converge towards a small invariant structure whose volume is null: the *attractor* of the system. The characteristics of this attractor determine the *dynamical regime* of the system. In one given system (defined by its weights and thresholds values), the transitions from one dynamical regime to the other occurs through bifurcations while continuously increasing the gain parameter g. We typically find four different dynamical regimes: fixed point, limit cycle, T2-torus and chaos (see Figure 1). This generic process of

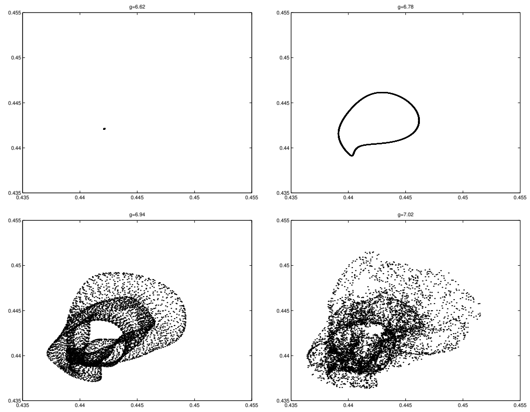

Figure 1: **Quasi-periodicity route to chaos.** Four return maps are represented, on the basis of mean signal $m_N(t)$, for increasing values of gain parameter g (i.e $g = 6.62$ – fixed point –, $g = 6.78$ – limit cycle –, $g = 6.94$ – T2-torus – and $g = 7.02$ – chaos –). Principle of return map : $m_N(t)$ is on the x-axis, $m_N(t+1)$ is on the y-axis. Transients have been discarded. Other parameters are $N = 200$, $\theta = 0.1$.

the dynamics becoming more complex is called a quasi-periodicity route to chaos (Bergé et al., 1992). More details on such quasi-periodicity route in our systems can be found in (Doyon et al., 1993, Cessac et al., 1994).

2.3 Cluster formation

We still consider here that our system is autonomous. We thus take $\mathbf{I}(t) = 0$. An accurate study of the individual signals is necessary in order to characterize the spontaneous dynamical organization taking place in our system. As soon as the gain parameter g is high enough, every random network tends to produce a complex spatio-temporal pattern of activation. We will see here that the non-linear transfer function f_g has a very important structuring role.

Theoretical results on the statistical repartition of activation in our model when the size tends towards infinity (thermodynamic limit) have been previously studied (Cessac, 1995). There are two possible regimes at the thermodynamic limit: fixed point and chaos. Considering potential signals $u_i(t) = -\theta + \sum_{j=1}^{N} J_{ij} x_j(t-1)$, the chaotic regime is analog to a Gaussian process at the thermodynamic limit:

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

Where $\mathbf{u}^*$ is a Gaussian static vector and $\mathbf{b}(t)$ a white noise.

We will now consider a finite-size system in a cyclic regime, near destabilization, with a *real*[1] positive period τ. Even if such cyclic regimes are not described at the thermodynamic limit, they however imitate the characteristics of a Gaussian process, in particular:

1. Mean potentials u_i^* repartition tends to obey to a Gaussian law.

2. Individual potential signals tend to be desynchronized. This means that individual phases are uniformly distributed in $[0,\tau[$.

We now consider activation signals $x_i(t) = f_g(u_i(t))$. Due to nonlinearity of f_g, we see that

1. Neurons whose mean potential u_i^* is strongly positive or negative have almost constant output signals. Such neurons are called inactive or quiescent. Only neurons whose potentials oscillate around zero have their signal amplified by the transfer function. Such neurons are called *active* neurons. They are responsible for the propagation of the inner signal (coming from the interactions within the RRNN) throughout the system. For usual parameter values, active neurons represent about 30% of the whole population.

2. Activation signal $x_i(t)$ tends to be sharpened, with peaks (or gaps) of activation corresponding to maxima (minima) of the potential signal $u_i(t)$. This leads to increase the correlation between neurons whose potential signals have neighbour phases. Such neurons have their activation signal almost phased-locked on discrete instants. We thus have the emergence of clusters of neurons which produce closely correlated activation signals. If we consider two clusters whose phase shift is 1, it seems that the first cluster propagates its signal towards the second cluster. From one cluster to another, we finally have a circular dynamical organization. This organization, which strongly depends on the value of τ, can not be deduced from the synaptic weights, but emerges from global interactions.

These two points help us figure out the dynamical organization of our networks. We have a majority of inactive neurons, and a minority of active neurons which tend to clusterize and propagate the inner signal in an *activation loop*. Moreover, for one given system, this circular dynamical organization remains stable in different regimes. Figure 2 shows that two neural clusters whose phase is opposite in cyclic regime keep this phase opposition in a chaotic regime.

2.4 Constraint dynamics

We now consider that the input signal $\mathbf{I}(t)$ is non-constant with time, so that there is a competition between inner influences and outer influences (the signal coming from the external world). The important point

[1] Even if our system is discrete time, the value of τ depends on the spectrum of the linearized system near the fixed point at destabilization value g_c. This implies that τ takes its value in $[2, +\infty[$.

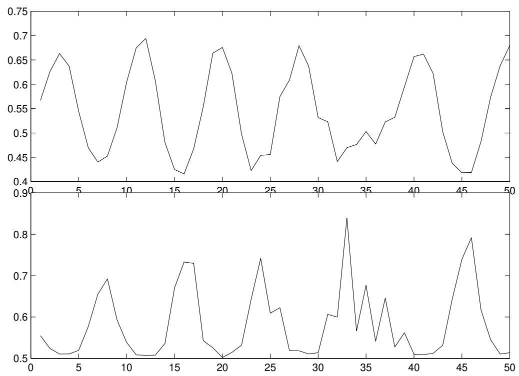

Figure 2: **Mean signals from two phase-opposite clusters in chaotic regime.** Parameters : $g = 5.5$, $N = 100$, $\theta = 0.1$.

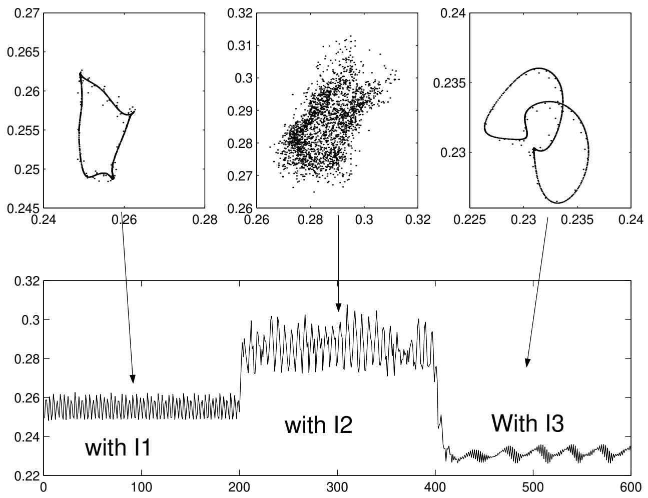

Figure 3: **Evolution of the dynamics while the system is successively submitted to 3 strongly correlated random static inputs.** Mean signal $m_N(t)$ and return maps are represented. We find two cyclic regimes and one chaotic regime. Parameters $g = 5$, $N = 200$, $\theta = 0.4$, $\sigma_I = 0.3$.

is that any change in spatial or temporal input characteristics modifies the nature of the dynamical system itself.

In simulations, input signals can be static or dynamic. A static input is a Gaussian random vector (of law $\mathcal{N}(0, \sigma_I^2)$) which is maintained during a period of time T (T is of the order of 100 time steps). A dynamic periodic input is a looping sequence of k random vectors so that the period of input signal is $\tau_I = k$.

Static inputs When a static input is presented during a period of time T, strong changes occur in dynamical organization. There is a change in the level of activity of neurons, a change in periodicity, etc... The important point is that two very close static inputs can produce very different patterns of activation. Figure 3 shows the evolution of the mean signal $m_N(t)$ when the system is submitted to 3 different static inputs $\mathbf{I_1}$, $\mathbf{I_2}$ and $\mathbf{I_3}$, for periods of 200 time steps. The three random vectors are strongly correlated (mean correlation: 0.95). However, dynamical regimes, periodicity and repartition of activity are very different in the three cases. It finally appears that our system is extremely sensitive to small changes on inputs values. The dynamical organization is thus *unstable* relatively to the system's inputs.

Periodic inputs When the system is submitted to a periodic input $\mathbf{I}(t)$, there is a competition between inner and outer periodicity (even in a chaotic regime where one can find residual periodicity). The system reveals to be very sensitive to external periodicity. This means that once again, the system reconfigures its dynamical organization in order to adapt to the external constraint.

3. Learning

So, in a constantly changing environment, any dynamical organization that emerges in the system is necessarily transient. In that sense, the system organization spontaneously adapts to its environment. However, because of its instability on the inputs, one can not say that the response of the system is specific to the input. Indeed, if one given static input $\mathbf{I_1}$ induces one given attractor, neighbour or noisy versions of $\mathbf{I_1}$ will not necessarily produce the same attractor. However, we want to build a system which can generalize, i.e which can react specifically to a neighbourhood of $\mathbf{I_1}$. We thus need to *stabilize* the dynamical response associated to $\mathbf{I_1}$. We will now implement a learning rule reinforcing dynamical regularities and activation loops in our system.

The learning rule obeys the following properties:

- Unsupervised learning: the dynamical response may not come from an outer command. The system determines its own response.
- On-line learning: there is no *a priori* knowledge on the nature of the inputs to learn.
- Local learning rule: the change in the synaptic weights only stands on the activity of afferent and efferent neurons.

A simple Hebbian learning rule was proposed on our model in (Daucé et al., 1998). However, this first rule induced severe limitations in terms of storage capacity.

We use in our present implementation a second rule based on an *habituation* principle. This choice relies on the idea that a neuron will favor afferent signals that often change with time, and ignore static signals. Our rule thus reinforces the *effective covariance* between afferent and efferent neurons. In particular, the constant components of afferent activation signals are substracted. Using a sliding window, we estimate the mean activation

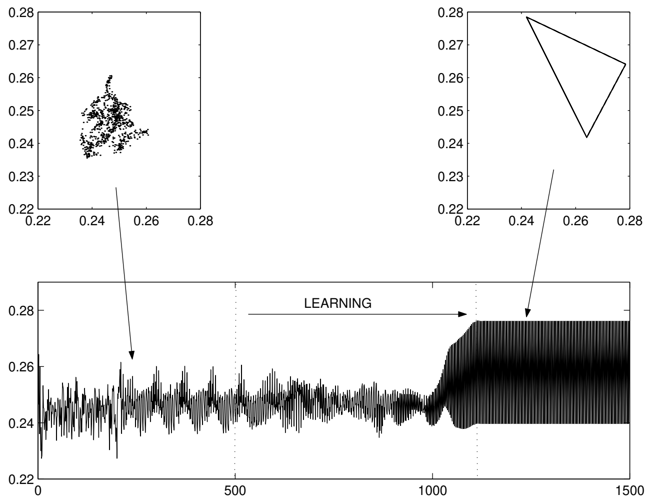

Figure 4: **A learning experiment.** Evolution of the mean signal $m_N(t)$ during the learning process, when the system is submitted to a static input $\mathbf{I_1}$. Upper left: return map of $m_N(t)$ before learning. Upper right: return map of $m_N(t)$ after learning (dots linked by lines). Main signal: time evolution of $m_N(t)$ during the whole experiment. Learning process is activated for $500 < t < 1100$. Parameters: $N = 200$, $g = 6.3$, $\alpha = 0.1$, $\theta = 0.4$, $\sigma_I = 0.3$.

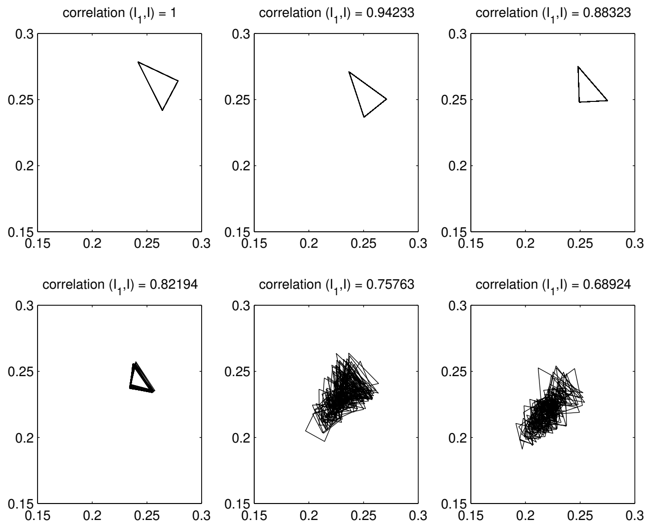

Figure 5: **stabilization of dynamical response in the neighbouhood of the learned input.** Six return maps on the basis of $m_N(t)$ signal after learning input $\mathbf{I_1}$. The first return map corresponds to the dynamics associated with I_1. The next 5 return maps correpond to 5 different inputs, whose correlation to $\mathbf{I_1}$ is decreasing from left to right. Dots are linked by lines in order to see 3-periodic attractors. Parameters: $N = 200$, $g = 6.3$, $\alpha = 0.1$, $\theta = 0.4$, $\sigma_I = 0.3$.

$\bar{x}_j$ (we only take into account the most recent values of activation). The covariance rule was first introduced by Sejnowski (Sejnowski, 1977), and has been adapted to our problem (capture of dynamical regularities) by taking into account the temporal shift induced by the transmission delay (in our case, the delay is 1). The rule is:

$$J_{ij}(t) = J_{ij}(t-1) + \frac{\alpha}{N}\,(x_i(t) - \bar{x}_i)(x_j(t-1) - \bar{x}_j)$$

We have seen that a majority of neurons in our system have an almost constant activation value (silent or saturated). Such neurons are not involved in the learning process. If two active neurons belong to clusters whose phase shift equals one, then the link from one to the other is systematically reinforced by the learning rule. Indeed, even when the dynamics is chaotic with a residual periodicity, the learning process tends to *simplify* it towards cyclic dynamics. It thus reinforces the activation loop structure, and increases both amplitude and periodicity of the inner dynamics.

Recognition and generalization Figure 4 shows a typical learning process on a system submitted to a static input $\mathbf{I_1}$. Before learning, the dynamics is chaotic, with a residual periodicity close to period 3. After learning, the dynamics is strictly periodic of period 3. When the learning process is stopped, the new system is now able to *recognize* $\mathbf{I_1}$, i.e. the system systematically switches its dynamics towards this specific 3-periodic dynamical organization when $\mathbf{I_1}$ is presented.

An important point is that after learning, the dynamical response associated to $\mathbf{I_1}$ is *stabilized.* This means that in the neighbourhood of $\mathbf{I_1}$, the system tends to produce a dynamical response which is very close to the one associated to $\mathbf{I_1}$ (see Figure 5).

So, learning displays two new properties in our system:

- **Recognition:** Our system is now able to discriminate between learned inputs and non-learned inputs. *Recognition occurs with a change in dynamical regime (bifurcation) leading to periodic attractors* associated to every learned input.

- **Generalization:** Our system produces almost similar dynamical organization in a neighbourhood of the learned input. This relies on the ability to produce the same actions in closely related situations.

Associativity The property of associativity can be shown when learning occurs with inputs composed of two different independent signals, i.e. $\mathbf{I}(t) = \mathbf{I_1}(t) + \mathbf{I_2}(t)$. $\mathbf{I_1}(t)$ is called the main stimulus and $\mathbf{I_2}(t)$ associate stimulus.

We will first take a simple example, which illustrates that this kind of system may:

- associate static and periodic inputs

- learn several associations

The learned signals are made of two components: $\mathbf{L_k}(t) = \mathbf{I_k}(t) + \mathbf{D_k}$, where $\mathbf{I_k}(t)$ is a periodic input

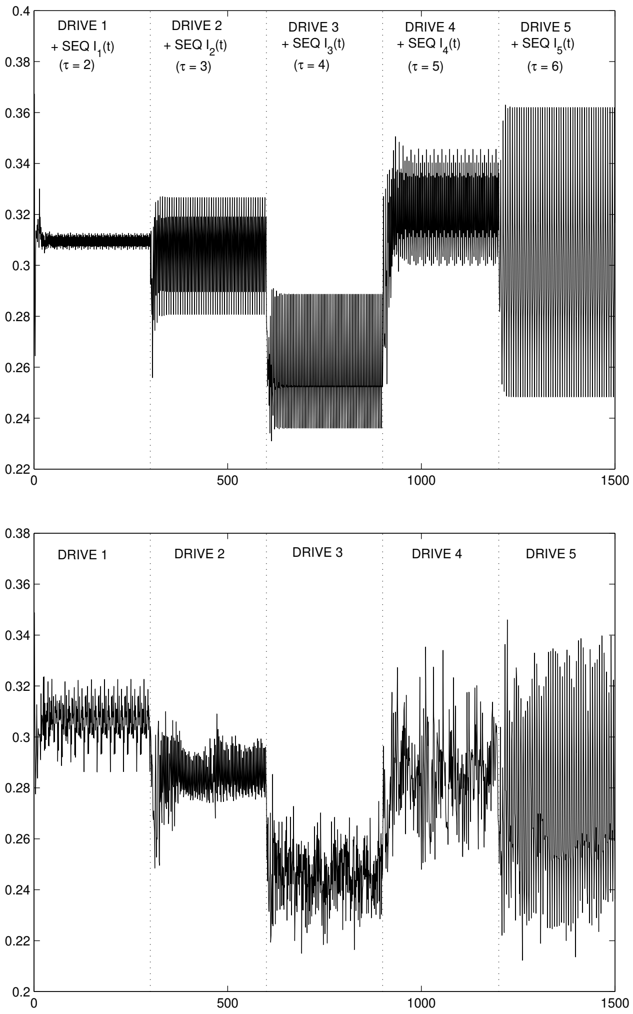

Figure 6: **Dynamical signals obtained with full stimulation (upper figure) and associate static stimulation (lower figure)** Five combined signals $\mathbf{L_k}(t) = \mathbf{I_k}(t) + \mathbf{D_k}$, for k in 1..5, have been learned on the same system. Time is on the x-axis, mean signal $m_N(t)$ on the y-axis. Every stimulus is presented for 300 time steps. Parameters: $N = 200$, $g = 8$, $\alpha = 0.1$, $\theta = 0.4$, $\sigma_I = 0.2$, $\sigma_D = 0.3$.

of period τ_k, made of centered gaussian vectors issued of $(\mathcal{N}(0, \sigma_I^2))$ and $\mathbf{D_k}$ a static gaussian vector issued of $(\mathcal{N}(0, \sigma_D^2))$. Figure 6 shows the dynamical behavior of a network which has learned five such associations. After learning, signals associated to the full input $\mathbf{L_k}(t)$ are in cyclic regime, and signals associated to static inputs $\mathbf{D_k}$ alone remain in chaotic regime. However, the same period and a correlation of the order of 0.4 between individual activation signals can be found in associated dynamics, while periods are different and correlations equal to zero in non-associated dynamics.

This property of associativity is very important in real world applications, as one want to see whether the system can produce a learned action in presence of an associate stimulus. The point now is to see how our system can produce a motor command on the basis of inner dynamics. For that, we need to build a system with interfaces that can interpret inner dynamical response.

4. Multi-layer models

The recognition and associativity properties denoted in our single population model can be usefully adapted in real applications such as robot navigation. The idea is to build interfaces which both display external signals to the RRNN and receive a signal from it. A nearly similar neural architecture can be found in (Williams and Zipser, 1989), but the back-propagating learning rule used by these authors prevents them from reaching "unstable" dynamics. Our purpose here is to start from such unstable dynamics in order to regularize it according to the input signal.

The learning rule is then extended to every link between the interface and the RRNN. This means in particular that when no external signal is displayed on the interface, the links from the RRNN produce a signal which shows what *should be present* on the interface, according to what has previously been learned.

In case we have two interfaces, one corresponding to visual perception and the other corresponding to motor movements, we build a system which should be able to produce movements according to both visual entries and learning-induced representations.

5. Global control architecture

The basis of the global control architecture is the PerAc block developed in our team (Gaussier and Zrehen, 1995). The robotic platform is a Koala robot provided by the K-Team. The main information source is obtained through a CCD camera. The pictures are not processed as a whole, but are split into several 32x32 subimages. These subimages are taken around some feature points which may vary depending on the application, but which mainly are high curvature points (corners ...) extracted from the gradient of the image. Then a log-polar transformation is applied to the subimages giving some invariance with respect to shift and distance. The subimages are afterwards learned on a Probabilistic Topological Map (PTM) (Gaussier and Zrehen, 1994). In a navigation context, the correspondence between a subimage (called now "landmark") with its angular position (azimuth) in respect with an absolute direction (north given by a compass for instance) gives the position of that landmark in the environment. The set of (landmark, azimuth) gives the position of the robot in the environment. Merging landmark and azimuth information is performed on an associative map. These configurations are learned on an other map (fig. 7). The neurons of this map may be linked to a particular movement. Thus, in association with each position in the environment, it is possible to learn a movement. This process enables to reach a goal by successively going from one learned position to the other (Gaussier et al., 2000).

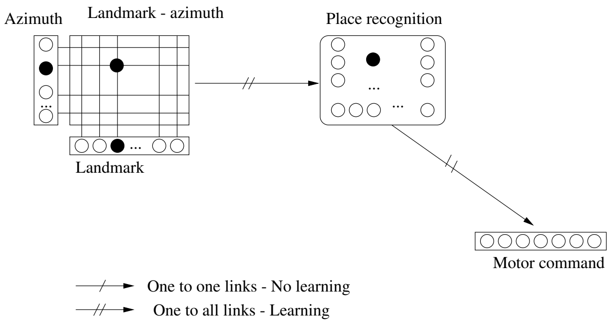

Figure 7: **Global architecture for merging landmark and azimuth informations, and learning a location.** The correspondence between a landmark and its azimuth is learned in the (landmark-azimuth) map. Each set of landmarks and azimuths (thus corresponding to one location) is learned by one neuron in the "Place Recognition" map. Then this neuron may be linked with one neuron in the "Motor Command" map. This gives the movement to perform at the learned location.

6. Robotic experiment

With the robotic experiment, we demonstrate that a RRNN may be used in real world applications. Our RRNN has the property to learn both spatial and temporal informations. So, it may learn the association between a position and a movement, and simultaneously a sequence of movements. Hence, we have replaced the map learning the set of landmark and azimuth information with a RRNN, and we have added a motor input to this RRNN in order to learn the sequence of movements (which was not possible in the architecture described in the previous section) (fig. 8).

In the experiments, the robot is limited to rotative movements corresponding to 7 possible rotations from $-90°$ to $90°$, with $30°$ steps. The choice between these actions is made according to the activation states of the motor interface. The interface is made of seven neurons, each associated to one motor command, so that the neuron with maximal activation determines the movement. At each position, what is learned is the movement to perform and the set of (landmark, azimuth) taken from the robot camera. In the experiments, the RRNN has $N = 100$ interconnected neurons.

Together with the image, we force a signal on the motor interface. This signal is periodic, and correspond to the sequence $(+30°, +60°, +90°)$, so that after one sequence the robot has made a half-turn. A different image is seen at each position. After two sequences the robot goes back to a previously learned position. So an image is only associated to one movement, but the same movement is associated to two different images (front

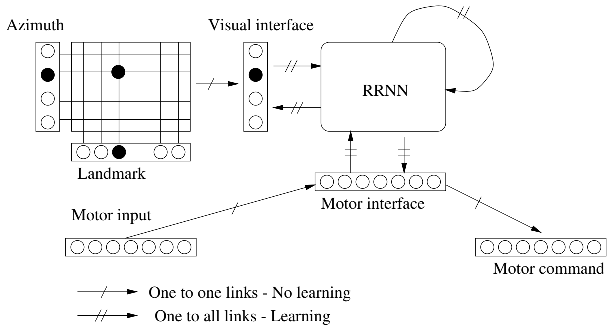

Figure 8: **Global architecture of the core of the system.** The previous "Place recognition" map has been replaced by the RRNN and two interface modules. In addition, in order to learn a motor sequence, a "Motor Input" map codes the movement to learn.

and back images after one sequence) (fig. 9).

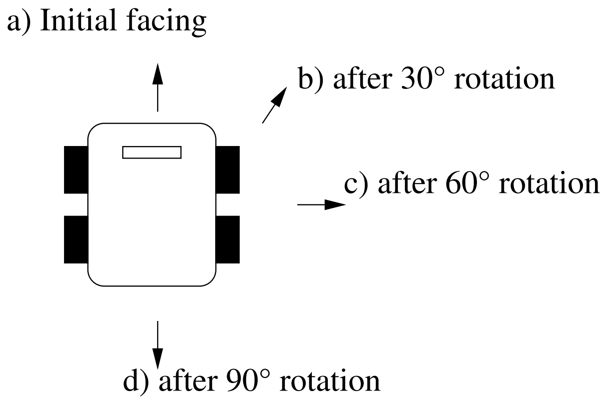

Figure 9: **Successive positions of the robot after the 3 rotation commands.** After these commands, the robot is facing backwards. Issuing these commands again let the robot go back to its initial position.

There are two stages in the training process. First, we iterate the dynamics without changing the weights, until the system reaches its stationary dynamics. Second, we iterate the learning rule while the robot is moving. Due to friction between wheels and ground, the real rotation performed is different from the command issued. So during the training process, the robot is moved back to a learned position when the shift is too big. The learning process is lasting 20 time steps (one time step corresponds to one movement).

After this learning process, the resulting system is tested. The forcing motor signal is removed, so that the robot determines its movement from the information issued by the RRNN. After a transient time of around 10 time steps, the robot performs the succession of learned rotations. As stated before, due to friction, there is a progressive shift of the robot orientation so that after a while the image in front of it is associated to a movement which is different from the movement it would perform if following the sequence. So there is a *conflict* between the movement associated with the perceived image and the movement proposed by the learned sequence. What happens is not a take over of one movement over the other. For some time steps the movements performed are not following the sequence anymore, nor correspond to the ones associated with the image (fig. 10). But eventually, the robot goes back to a learned position, triggers the associated movement and now resumes the good sequence (which now matches the good pictures). So the robot is able to recalibrate its position based on a recognized image.

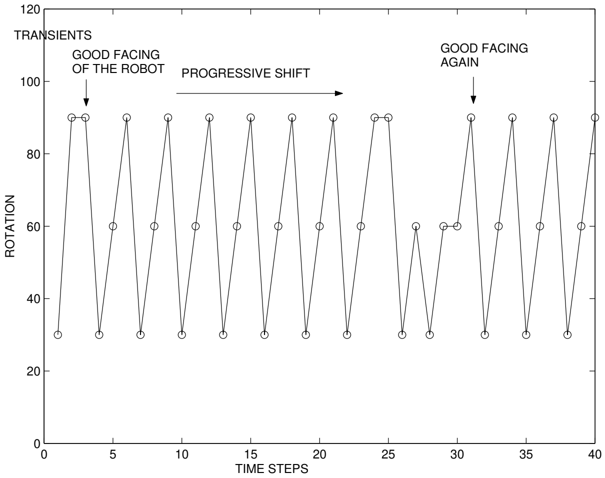

Figure 10: **Example of recalibration after a shift.** Rotation angle versus time when the robot is gradually shifting. The first two steps are transients. Then the rotations show the learned periodic sequence ($+30°, +60°, +90°$) corresponding to the visual inputs. The real robot angle shifts and suddenly the robot looses the correspondence between the image in front of him and the associated movement. Finally the robots finds a good matching and resumes the periodic sequence.

The second experiment we have conducted is masking the camera once the robot is performing the good sequence. The movement sequence is not the good one anymore, though there are some patterns of it. This shows that what has been learned is no the movement sequence alone. When the robot can see again, it is able to go back to the learned sequence based on a recognized image (fig. 11).

7. Conclusion

We can make several comments on this work. First, the RRNN we have presented has both properties of learning input-output correlation (sensory-motor associations) and temporal sequences. Hence it may act as a working memory where *representations* of a combination of the external perception and the internal state are coded. The complexity of the coding is not an obstacle for using such a system in real world applications as demonstrated on the robot experiment. However,

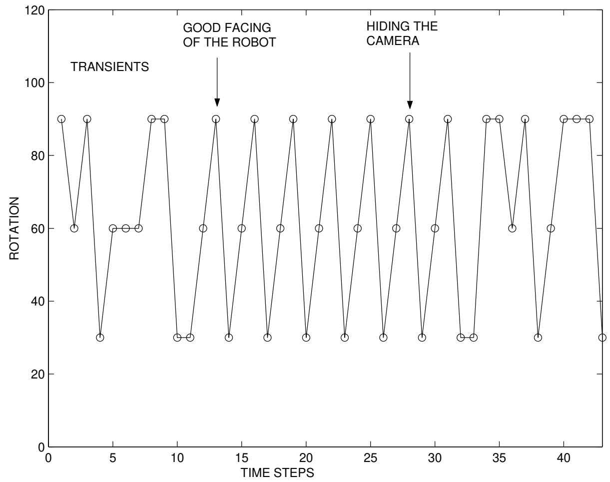

Figure 11: **Example of visual input masking.** Rotation angle versus time when the camera is suddenly masked. The first twelve steps are transients. When the camera is hidden, the robot looses the periodic sequence.

these working representations have to be transfered to another part of the system for long time storage. Moreover, this storage may not necessary be of the same nature as the one in the working memory. In biological terms, what we have constructed plays the same role as an hippocampus, our RRNN playing the role of the CA3 structure. This architecture is linked with a planning structure enabling action selection and motor control (Quoy et al., 1999b, Quoy et al., 1999a). It remains now to be seen how the RRNN may be linked with such a planning system. It is already possible to link an image and a movement with an internal motivation (searching for a particular object, or going to the power station ...). Another kind of hippocampal model has been implemented in our group for learning temporal sequences (Gaussier et al., 1999) and sensory-motor associations (Gaussier et al., 2000).

Second, the learning rule increase the coherency between the inner chaotic dynamics of the RRNN and the evolving values of visual and motor inputs. This higher coherency corresponds to a regularization of the dynamics (it becomes less chaotic), and the possibility to use such regularity to produce a motor command. When the visual input matches the previously learned visual sequence, the dynamics remains regular and the motor commands correspond to the learned sequence. When the visual information tends to misfit the learned visual sequence, the dynamics qualitatively changes and gets more chaotic, so that the system is able to perform an *exploration* of its visual environment in order to find the matching visual sequence. When there is no possible match (for instance when the scene is hidden, or when the robot is moved to another place), the dynamics remains chaotic, and the robot stands searching for a matching sequence in its environment.

Third, our system has still to be improved. For the moment, the RRNN is working with discrete time steps. But the real stimuli are continuous by nature. So we need first to develop the same kind of RRNN under a continuous time formalism. Next, the main drawback is the need to tell the system when to stop learning. We have chosen to stop after a fixed number of iterations. If learning reduces the dynamics on a limit cycle, it is easy to recognize such a regular signal and stop learning. However, it is still unclear about an optimal learning stopping criterion when the system is still chaotic. Moreover, we believe there is no such optimal criterion ...

In conclusion, both aspects of association and sequence learning may be found in our system, but the most interesting point stands in its ability to dynamically adapt its behavior to external changes. This adaptability is not dependent on an ad hoc parameter tuning, nor a set of specific rules, but emerges from the interactions between the robot and its environment. These interactions are taken into account by a learning rule. Not only biologically founded, these new results emphasize the role of RRNN for designing adaptive systems.

This work was supported by a French GIS contract on Cognitive Sciences entitled "Dynamical coding of informations by asymmetrical recurrent neural networks". The collaboration includes the following labs: ETIS - Cergy-Pontoise (M. Quoy - P. Gaussier), ONERA-CERT - Toulouse (E. Daucé - M. Samuelides), INSERM U455 - Toulouse (B. Doyon) and INLN - Nice (B. Cessac). We would thank all these colleagues for helpful comments and discussions.

References

Amari, S. (1972). Characteristics of random nets of analog neuron-like elements. *IEEE Trans. Syst. Man. Cyb.*, 2(3).

Amari, S. (1977). Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27:77–87.

Bergé, P., Pomeau, Y., and Vidal, C. (1988,1992). *L'ordre dans le chaos.* Hermann.

Bicho, E. and Schöner, G. (1997). The dynamics approach to autonomous robotics demonstrated on a low-level vehicle platform. *Robotics and Autonomous System*, 21:23–35.

Cessac, B. (1995). Increase in complexity in random neural networks. *Journal de Physique I*, 5:409–432.

Cessac, B., Doyon, B., Quoy, M., and Samuelides, M. (1994). Mean-field equations, bifurcation map and route to chaos in discrete time neural networks. *Physica D*, 74:24–44.

Daucé, E., Quoy, M., Cessac, B., Doyon, B., and Samuelides, M. (1998). Self-organization and dynamics reduction in recurrent networks: stimulus presentation and learning. *Neural Networks*, 11:521–533.

Doyon, B., Cessac, B., Quoy, M., and Samuelides, M. (1993). Control of the transition to chaos in neural networks with random connectivity. *Int. J. of Bif. and Chaos*, 3(2):279–291.

Gaussier, P., Joulain, C., Banquet, J., Leprêtre, S., and Revel, A. (2000). The visual homing problem: an example of robotics/biology cross fertilization. *Robotics and Autonomous Systems*, 30:155–180.

Gaussier, P., Moga, S., Banquet, J., and Quoy, M. (1999). From perception-action loops to imitation processes. *Applied Artificial Intelligence*, 1(7).

Gaussier, P. and Zrehen, S. (1994). A topological map for on-line learning : Emergence of obstacle avoidance in a mobile robot. In *From Animals to Animats: SAB'94*, pages 282–290, Brighton. MIT Press.

Gaussier, P. and Zrehen, S. (1995). Perac: A neural architecture to control artificial animals. *Robotics and Autonomous Systems*, 16(2-4):291–320.

Hertz, J. and Prugel-Bennett, A. (1996). Learning synfire chains: turning noise into signal. *Int. J. Neural Systems*, 7:445–450.

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci.*, 79:2554–2558.

Jirsa, V. K., Fuchs, A., and Kelso, J. A. S. (1998). Connecting cortical and behaviorall dynamics: Bimanual coordination. *Neural Computation*, 10:2019–2045.

Mac Leod, K. and Laurent, G. (1996). Distinct mechanisms for synchronization and temporal patterning of odor-encoding cell assemblies. *Science*, 274:976–979.

Meyer, J. and Wilson, S. (1991). From animals to animats. In Press, M., (Ed.), *First International Conference on Simulation of Adaptive Behavior.* Bardford Books.

Neuenschwander, S., Engel, A., Konig, P., Singer, W., and Varela, F. (1996). Synchronization of neuronal responses in the optic tectum of awake pigeons. *Visual Neurosci*, (13):575–584.

Quoy, M., Gaussier, P., Leprêtre, S., Revel, A., Joulain, C., and Banquet, J. (1999a). A neural model for the visual navigation and planning of a mobile robot. In *European Conference on Artificial life, ECAL99*, Lausanne.

Quoy, M., Gaussier, P., Leprêtre, S., Revel, A., Joulain, C., and Banquet, J. (1999b). A planning map for mobile robots: speed control and paths finding in a changing environment. In *European Workshop on Learning Robots, EWLR99*, Lausanne.

Schöner, G., Dose, M., and Engels, C. (1995). Dynamics of behavior: theory and applications for autonomous robot architectures. *Robotics and Autonomous System*, 16(2-4):213–245.

Sejnowski, T. J. (1977). Strong covariance with nonlinearly interacting neurons. *Journal of Mathematical biology*, 4:303–321.

Skarda, C. and Freeman, W. (1987). How brains make chaos in order to make sense of the world. *Behav. Brain Sci.*, 10:161–195.

Tani, J. and Fukumura, N. (1995). Embedding a grammatical description in deterministic chaos: an experiment in recurrent neural learning. *Biological Cybernetics*, 72:365–370.

Tani, J. and Nolfi, S. (1998). Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems. In Pfeifer, R., Blumberg, B., Meyer, J., and Wilson, S., (Eds.), *From Animals to Animats: Simulation of Adaptive Behavior SAB'98*, pages 270–279. MIT Press.

Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running recurrent neural networks. *Neural Computation*, 1(2):270–280.

Yao, Y. and Freeman, W. (1990). Model of biological pattern recognition with spatially chaotic dynamics. *Neural networks*, 3:153–170.