# The Role of Internal Oscillators for the One-Shot Learning of Complex Temporal Sequences

Matthieu Lagarde, Pierre Andry, Philippe Gaussier

# The Role of Internal Oscillators for the One-Shot Learning of Complex Temporal Sequences.

Matthieu LAGARDE, Pierre ANDRY, Philippe GAUSSIER

ETIS, Neurocybernetic Team, UMR CNRS 8051
2, avenue Adolphe-Chauvin, University of Cergy-Pontoise, France
{lagarde,andry,gaussier}@ensea.fr

**Abstract.** We present an artificial neural network used to learn online complex temporal sequences of gestures to a robot. The system is based on a simple temporal sequences learning architecture, neurobiological inspired model using some of the properties of the cerebellum and the hippocampus, plus a diversity generator composed of CTRNN oscillators. The use of oscillators allows to remove the ambiguity of complex sequences. The associations with oscillators allow to build an internal state to disambiguate the observable state. To understand the effect of this learning mechanism, we compare the performance of (i) our model with (ii) simple sequence learning model and with (iii) the simple sequence learning model plus a competitive mechanism between inputs and oscillators. Finally, we present an experiment showing a AIBO robot, which learns and reproduces a sequence of gestures.

## 1 INTRODUCTION

Our long term goal is to build an autonomous robot able to learn sensorimotor tasks. Such a system should be (i) able to acquire new "behaviors" : gestures, objects manipulation as sequences combining multimodal elements of different levels. To do this, an autonomous system must (ii) take advantage of information of the associations between vision and motor capabilities. This paper focuses essentially on the first point : learning, predicting and reproduction of complex sensorimotor sequences.
In this scope, solutions based on neural networks are an interesting solution. Neural networks are able to learn sequences using associative mechanisms. Moreover, these networks offer a level of coding (neuron) that takes into account information about the lower sensorimotor system; such systems avoid the use of symbols or information that could separate the sequence learning component from the building of associations between sensation and action. Networks are adapted to online learning favoring easier interactions with humans and other robots. Among these models, chaotic neural networks are based on recurrent network (RN). In [1], a fully connected RN learns a sequence thanks to a single layer of neurons. The dynamics generated by the network help to learn a short sequence. After a few iterations, the learned sequence vanishes progressively. In [2], a random RN (RRN) learns a sequence thanks to a combination of two layers of

neurons. The first layer generates an internal dynamic by means of a RRN. The second layer generates a resonance phenomenon. The network learns short sequences of 7 or 8 states. But this model is highly sensitive to noises or the stimulus variations and does not learn long periods sequences. A similar model is the Echo States Network (ESN) based on RRN for short term memory [3] (STM). Under certain conditions (detailed in [4]), the activation of each neuron in the hidden layer, is a function of the input history presented to the network; this is the echo function. Once again, the idea is to use a "reservoir" of dynamics from which the desired output is learned in conjunction with the effect of the input activity.

In the context of robotics, many models concern gesture learning. By means of nonlinear dynamical systems, [5] develops control policies to approximate the recorded movements and to learn them with a fitting of mixture model using a recursive least square regression technique. In [6], the trajectories of gestures are acquired by the construction of motor skills with a probalistic representation of the movement. Trajectories can be learnt through via points [7] with parallel vector-integration-to-endpoint models [8]. In our work, we wish to be able to re-use and detect subsequences and possibly, combine them. Thus, we need to learn some of the important components of the sequence and not only to approximate the trajectory of the gesture.

In this paper, we present a biologically inspired model of neural network for temporal complex sequences learning. A first approach described in [9] proposes a neural network for the online learning of the timing between events for simple sequences (with non ambiguous states like "A B C"). We propose a model for complex sequences (with ambiguous states like A and B in "A B A C B"). In order to remove the ambiguous states or transitions, we use batteries of oscillators as a reservoir of diversity allowing to separate the inputs appearing repeatedly in the sequence. In section 3, we show results from simulations comparing the performances of 3 different systems involved in the learning and reproduction of the same set of complex sequences : (i) the system described in [9], (ii) this system plus a simple competitive mechanism between the oscillators and the input (showing the effect of adding internal dynamics in order to separate ambiguous states) and (iii) a system optimizing the use of the oscillators by using an associative learning rule in order to recruit new internal states when needed (repetition of the same input state). Section 4 details the application of our model on a real robot for the learning of a complex gesture. Finally, we conclude and point out some open problems.

## 2 A MODEL FOR TIMING AND SEQUENCE LEARNING

The architecture (Fig. 1) is based on a neurobiological model [10] inspired from some of the properties of the cerebellum and the hippocampus. This model uses associative learning rules between past inputs memorized as a STM and present inputs in order to learn the timing of simple sequences. "Simple" refers

here to the sequences in which the same state appears only once. The main advantage of this model is that the associative mechanism also learns the timing of the sequence, which allows accurate predictions of the transitions that compose the sequence. In order to learn complex sequences in which the same state is repeated several times, in our model we have added a mechanism that generates internal dynamics and that can be associated with the repeated inputs of the sequence. The association between the repeated inputs and different activities of the oscillator allows to code hidden states with different and un-ambiguous patterns of activities. As a result, our architecture manages to learn/predict and reproduce complex temporal sequences.

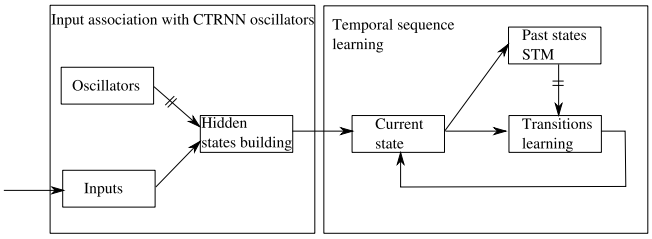


**Fig. 1.** Complex sequences learning model. Barred links are modifiable connexions. The others are associated to unmodifiable connexions. The left part is detailed in figure 3.A and 3.B. The right part is detailed in figure 4.

### 2.1 Generating internal diversity

Oscillators are very much used in robotic applications like locomotion using central pattern generator (CPG) [11]. An oscillator is a continuous time recurrent neural network (CTRNN) composed of two neurons (Fig. 2.A). The study on CTRNNs can be found in [12]. This kind of oscillators is known for stability, and resistance to the noises. CTRNN are easy to implement too. A CTRNN coupling two neurons produces an oscillator (Fig. 2.B) :

$$\tau_e.\frac{dx}{dt} = -x + S((w_{ii} * x) - (w_{ji} * y) + w_{e_{const}}) \quad (1)$$

$$\tau_i.\frac{dy}{dt} = -y + S((w_{jj} * y) + (w_{ij} * x) + w_{i_{const}}) \quad (2)$$

with $\tau_e$ a time constant for the excitatory neuron and $\tau_i$ for the inhibitory neuron. $x$ and $y$ are the activity of the excitatory and the inhibitory neuron respectively. $w_{ii}$ is the weight of the recurrent link of the excitatory neuron, $w_{jj}$ the weight of the recurrent link of the inhibitory neuron. $w_{ij}$ is the weight of the link from the excitatory neuron to inhibitory neuron. $w_{ji}$ is the weight of the link from the inhibitory neuron to excitatory neuron. $w_{e_{const}}$ and $w_{i_{const}}$ are the weights of the links from the constant inputs. And $S$ is the transfer function of each neuron. In our model, we use three oscillators with $\tau_e = \tau_i$.

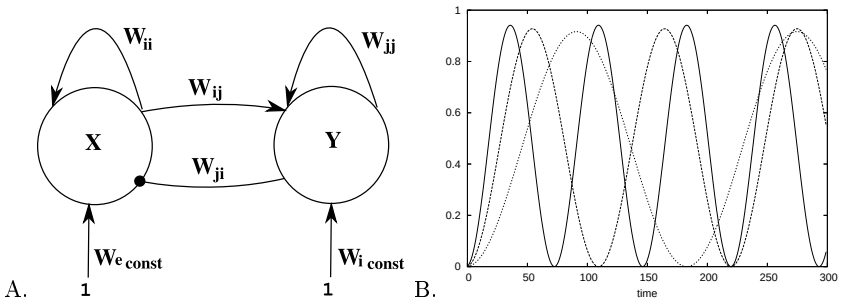


**Fig. 2.** A. Oscillator model. Left neuron is excitatory and right neuron is inhibitory. Excitatory links are : $W_{ii} = 1$, $W_{jj} = 1$, $W_{ij} = 1$. Inhibitory links is : $W_{ji} = -1$, Constant input value is equal to 1 with constant links $W_{e_{const}} = 0.45$ and $W_{i_{const}} = 0$. Initial activities of neurons are $X(0) = 0$, $Y(0) = 0$.
B. Display of the instantaneous mean frequency activity of 3 oscillators systems with $\tau_1 = 20$ (plain line), $\tau_2 = 30$ (long dashed line), $\tau_3 = 50$ (short dashed line).

### 2.2 learning of internal states

In order to use repeatedly the same input in a given sequence, different configurations of oscillators can be associated with the same input. To understand the generation of diversity and its implication in our learning algorithm, we have tested two mechanisms : a simple competition coupling input states with oscillators (Fig. 3.A) and an associative mechanism based on a learning rule (Fig. 3.B) that recruits neurons according to the activities of the oscillators and the repeated inputs.

**Competitive mechanism** The competition is computed as follow : each neuron $ij$ of the Competition group acts as an neuron performing the logical operator AND between the neurons of the Inputs group and of the Oscillators group :

$$Pot_{ij} = (w_{input_i} * x_{input_i} + w_{osci_j} * x_{osci_j}) - threshold_{ij} \tag{3}$$

with $w_{input_i} = 1$, $w_{osci_j} = 1$, $threshold_{ij} = 1.2$, $x_{input_i}$ the activity of the input at index $i$ and $x_{osci_j}$ the activity of the oscillator at index $j$.
In a second step, a competition between all neurons $ij$ of the Competition group is applied :

$$Winner_{ij} = \begin{cases} 1 \; if \; ij = Argmax_{ij}(Pot_{ij}) \\ 0 \; otherwise \end{cases} \tag{4}$$

The winner neuron becomes the input of the temporal sequence learning network (subsection 2.3). In this way, a "reservoir" of oscillator neurons can be used as a way to associate the same input with different internal patterns. Intuitively, the simple competition (no learning is required here) allows to directly select different "internal" states corresponding to the same input repeated many times in the sequence. For example in Fig. 3.A, each input (A,B,C,D) can appears up to 3 times (corresponding to the number of oscillators) in the same sequence. Moreover, such a mechanism does not disturb the prediction nor the reproduction

of the sequence. Obviously, if the competition between oscillators is an avenue worth exploring, it is still possible to have ambiguity. An input can be associated with same winner oscillator two or more times. Consequently, there is still potential ambiguities on the "internal" states of our model, and some sequences could not be reproduced correctly. A precise measure of this problem corresponds to the probability that the same state can be associated with the same oscillator several times and therefore the "internal" state partially depends of the shape, phase and number of oscillators. Typically, the problem happens when a given state comes back with the same frequency as the selected oscillator. The curves C2 and C3 on figure 5 show the performances of the competitive mechanism. To solve this problem, an associative mechanism allowing to recruit neurons coding "internal" states has been added.

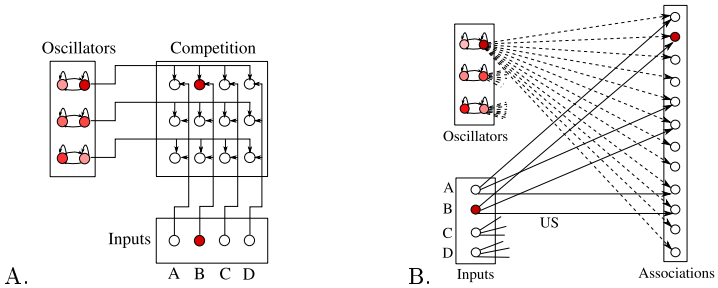


**Fig. 3.** A. Model of the neural network coupling an input state with an oscillator. All links are fixed connections. B. Model of the neural network used to associate an input state with a configuration of oscillators. Only few links are represented for the legibility. Dashed links are modifiable connections. Solid links are fixed connections.

**Associative mechanism** The learning process of an association between an input state and a configuration of oscillators is :

$$US = w_i * x_i \tag{5}$$

with $w_i$ the weight of the link from input state $i$, and $x_i$ activity of the input state $i$. If $US > threshold$, we compute the potential and the activity of the neuron as follow :

$$Pot_j = \sum_{j=0}^{M_{osci}} |(w_j - e_j)| \qquad Act_j = \frac{1}{1 + Pot_j} \tag{6}$$

with $M_{osci}$ the number of oscillators, $w_j$ the weight of the link from oscillator $j$, and $e_j$ the activity of the oscillator $j$. The neuron that has the minimum activity is recruited : $Win = Argmin_j(Act_j)$. Initial weigths of connexions have high values. The oscillators configuration is learnt according to the error of distance $\Delta w_j = \varepsilon(e_j - w_j)$ with $\varepsilon$ a learning rate, $w_j$ weight of link from Oscillator $j$, and $e_j$ activity of oscillator $j$. The Associations group becomes the new input

of the temporal sequence learning network (subsection 2.3). As showed on the figure 3.B, an input allows recruiting 3 different neurons coding "internal" states. They correspond to the connectivity of the unconditional links chosen between the Inputs group and the Association group. The associative mechanism ensures to recruit a new "internal" state for each input (A, B, C or D) from the sequence. The connectivity of links between the Input group and the Association group, has been chosen to have a number of hidden states equal for each input. This allows the comparison between the different models in our simulations. But it could be possible to change the connectivity of the links to allow the recruitement of more hidden states for each repeated input in the sequence. We have tested this mechanism in our architecture in simulation and robotic application.

### 2.3 Temporal sequences learning

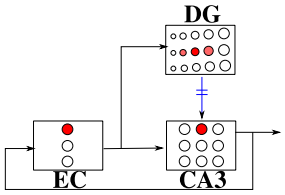


**Fig. 4.** Representation of hippocampus. Entorhinal Cortex (EC) receives inputs and transmits them to Dentate Gyrus (DG) and CA3 pyramidal cells. Between the DG group and the CA3 group there are fully connected with modifiable connections. Between the EC group and CA3 group, and the EC group and DG group, there are fixed one to neighborhood connections.

This part of model is based on a schematic representation of hippocampus [10] (Fig. 4). DG represents past state (STM), and develops a temporal activity spectrum. CA3 links allow pattern completion and recognition between incoming state from EC and previous state maintained in DG. We suppose the DG activity can be modelled as follow :

$$Act_{j,l}^{DG}(t) = \frac{1}{m_j} \cdot \exp - \frac{(t - m_j)^2}{2 \cdot \sigma_j} \tag{7}$$

with $Act_{j,l}^{DG}$ the activity of the cell at index $l$ on the line $j$, $t$ the time, $m_j$ a time constant and $\sigma_j$ the standard deviation. Neurons on one line share their activity in the time and represent a temporal trace of EC. Learning of an association is on the weights of links between CA3 and DG. The normalization of the activity coming from DG neurons is performed due to the normalization of the DG-CA3 weights.

$$W_{CA3(i,j)}^{DG(j,l)} = \begin{cases} \frac{Act_{j,l}^{DG}}{\sum_{j,l}(Act_{j,l}^{DG})^2} \; if \; Act_j^{DG} \neq 0 \\ unchanged \quad otherwise \end{cases} \tag{8}$$

Interestingly, this model has the property to work when a same input comes several times continously. Thanks to the derivative group EC, a repeated input

is stored during the total time of its presence. Consequently, two successive states are not ambiguous for the system ("A A" = "A").

## 3 SIMULATION RESULTS

A temporal sequence of states is rarely replayed two times exactly with the same rhythm. Time can vary between two states especially when demonstrating a sequence to a robot. In our simulations we apply a time variation between states and observe the consequences on three architectures. The first architecture is the model of simple sequences learning presented in subsection 2.3. The second is the same model plus the competitive mechanism presented in subsection 2.2. The third architecture is the same as the first one plus the associative mechanism (Fig. 1) seen in subsection 2.2. References sequences are generated to be successfully reproduced by the second architecture with a timing variation of 0%. All architectures are trained with the same sequences and the same maximum of timing variation (0%, 5% or 10%), but with a time variation randomly chosen between 0 and the maximum variation of the time. In our experiments, to bootstrap a sequence, we provide the first state. Consequently, this state will not be ambiguous in the sequences. For example, a complex sequences can be "D B C B A C A B" : "D" is the starting state and it will not be repeated after. Fig. 5 shows the performances of each architecture. We can see that the first ar-

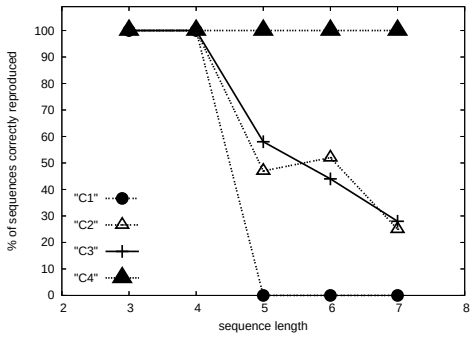


**Fig. 5.** C1 : first architecture : simple sequences learning. The results are the same with time variation of 0%, 5% and 10%. C2 : second architecture : complex sequences learning with a competitive mechanism and a time variation of 5%. C3 : second architecture : complex sequences learning with a competitive mechanism and a time variation of 10%. C4 : current architecture : complex sequences learning with an associative mechanism. The results are the same timing variation of 0%, 5% and 10%.

chitecture (subsection 2.3) has very good performances with sequences of 3 and 4 states, because those sequences have no repeated states (simple sequences). With sequences having more than 4 states, the performances fall drastically , because there is at least one state repeated in the sequences. We can see the

time variation has no effect on the performances. The architecture can not reproduce them, because CA3 group learns two transitions and, thus it predicts two states for each repeated input. The second architecture using competitive mechanism, has better performances, but, as we have seen previously in subsection 2.2, ambiguous internal states can appear and reduce this gain of sequences correctly reproduced. Consequently, like the first architecture, the CA3 group learns two "internal" states and, thus, it predicts two states from one input repeated. We can see, the performances change according to the timing variation between states in the sequences : a same input from a given sequence can be associated with two different oscillators and, consequently a different "internal" state wins. Thanks to the recruitment mechanism, the third architecture, has the best performances : 100% with all tested sequences. There are not ambiguous states or "internal" states. The time variation has no effect on the performances of the model.

# 4 ROBOTIC APPLICATION

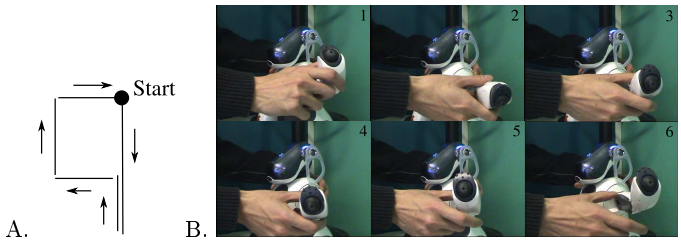


**Fig. 6.** A. Representation of desired sequence. It begins from the start point. B. We manipulate Aibo passively. It learns the succession of orientations of the movement from these front left leg motor information.

The robot used in our experiments is an Aibo ERS7 (Sony). In our application, we use only the front left leg, in a passive movement mode to learn a sequence of gestures. The sequence to be learned and reproduced is showed Fig. 6.A. In this application, we test the third architecture previously described.
During learning, we manipulate the front left leg of the robot passively (Fig. 6.B). During the execution of the movement, the neural network learns online, and in one shot the succession of the joints orientation thanks to the motors feedback information of its leg (proprioceptive signal). Hence, the inputs of our model are the orientations/angles of the leg. The recorded motors information while learning are shown in Fig. 7.X-*learning* (horizontal movements) and Fig. 7.Y-*learning* (vertical movements).
To initiate the reproduction of the sequence by the robot, we give the first state of the sequence ("down"). As Aibo can not be manipulated when motors are activated, we send the command directly to the robot. Next, Aibo plays the sequence autonomously (Fig. 7, top). With the starting state, our model predicts the next orientation and send the corresponding command to the robot.

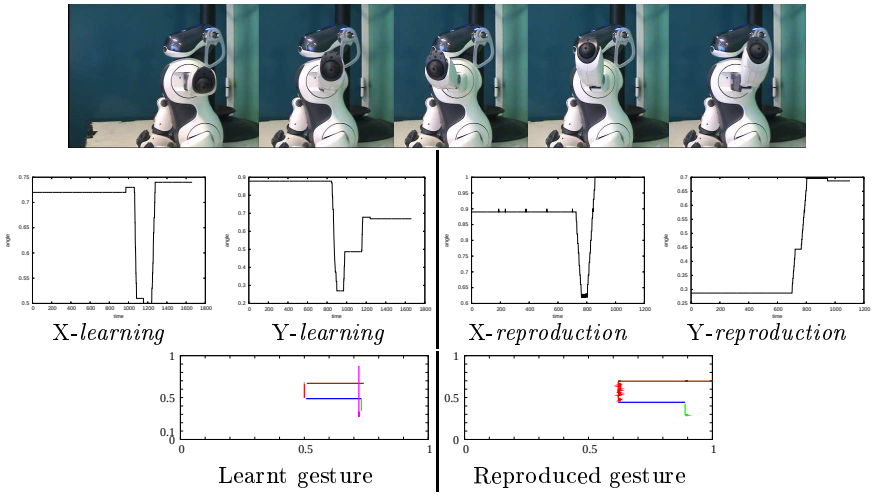


**Fig. 7.** Top : Aibo reproduces the learnt sequence. Middle : X-*learning* and Y-*learning* are respectively the horizontal and the vertical motors information while robot learns the sequence. X-*reproduction* and Y-*reproduction* are respectively the horizontal and the vertical motors information during the reproduction of the sequence. On the figure Y-*reproduction*, the first movement is not reproduced (not predicted), but given by the user in order to trigger the recall it is our bootstrap state to start the sequence. X-axis are the time and Y-axis are the angles of the motors.

## 5 CONCLUSIONS AND DISCUSSIONS

We have proposed a model of neural network for the learning of complex temporal sequences. This model introduces an associative mechanism taking advantage of a diversity generator composed of oscillators. These oscillators are based on coupled CTRNN. This model is efficient in the frame of autonomous robotics and succeed in learning in one shot the timing of sequences of gestures.
During the robotics application, we have noticed that the robot reproduces the sequence with a different amplitude of the movement. This effect comes from the speed of the displacement of the leg of Aibo. In our application, the speed of the reproduction is a predefined constant different from the user dynamic during learning. The rhythm of the sequence is respected thanks to atemporal group of neurons. A possible improvement would be to add a model like CPG [5] for each movements ("up", "down", "left" and "right") composing sequences with variable speeds. In our model, the number of neurons coding the associations between the inputs and the oscillators, represents the size of the "short term memory". In our simulations and application, the sequences learnt do not saturate the "memory". It would be interesting to analyze the behavior of the neural network with longer sequences, and test the limitations of the system when the neural limit has been reached by the recruitment mechanism. In the present system, it would mean that the already recruited neurons could be erased in order to encode new states.

In further works, this sequence learning model will complete a model for imitation based on low level sensorimotors capabilities and vision [13]. In this way, the robot will learn sensorimotors capabilities based on its vision and learn a demonstrated gesture from human or robot by imitation and reproduce it.

## 6 ACKNOWLEDGEMENTS

This work was supported by the French Region Ile de France, the network of excellence HUMAINE and the FEELIX GROWING european project. (FP6 IST-045169)

## References

1. Molter, C., Salihoglu, U., Bersini, H.: Learning cycles brings chaos in continuous hopfield networks. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN) conference. (2005)
2. Daucé, E., Quoy, M., Doyon, B.: Resonant spatio-temporal learning in large random neural networks. Biological Cybernetics **87** (2002) 185–198
3. Jaeger, H.: Short term memory in echo state networks. Technical Report GMD Report 152, German National Research Center for Information Technology (2001)
4. Jaeger, H.: The "echo state" approach to analysing and training recurrent neural networks. Technical Report GMD Report 148, German National Research Center for Information Technology (2001)
5. Ijspeert, A.J., Nakanishi, J., Shibata, T., Schaal, S.: Nonlinear dynamical systems for imitation with humanoid robots. In: Proceedings of the IEEE/RAS International Conference on Humanoids Robots (Humanoids2001). (2001) 219–226
6. Calinon, S., Billard, A. In: Learning of Gestures by Imitation in a Humanoid Robot. K. dautenhahn and c.l. nehaniv edn. Cambridge University Press (2006) in press.
7. Hersch, M., Billard, A.: A biologically-inspired model of reaching movements. In: Proceedings of the 2006 IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, Pisa (2006)
8. Bullock, D., Grossberg, S.: Neural dynamics of planned arm movements: Emergent invariants and speed-accuracy properties during trajectory formation. Psychological Review **95** (1988) 49–90
9. Ans, B., Coiton, Y., Gilhodes, J.C., Velay, J.L.: A neural network model for temporal sequence learning and motor programming. Neural Networks **7**(9) (1994) 1461–1476
10. Gaussier, P., Moga, S., Banquet, J.P., Quoy, M.: From perception-action loops to imitation processes. Applied Artificial Intelligence (AAI) **1**(7) (1998) 701–727
11. Ijspeert, A.J.: A neuromechanical investigation of salamander locomotion. In: Proceedings of the International Symposium on Adaptive Motion of Animals and Machines (AMAM 2000). (2000)
12. Yamauchi, B., Beer, R.D.: Sequential behaviour and learning in evolved dynamical neural networks. Adapt. Behav. **2**(3) (1994) 219–246
13. Andry, P., Gaussier, P., Nadel, J., Hirsbrunner, B.: Learning invariant sensorimotor behaviors: A developmental approach to imitation mechanisms. Adaptive behavior **12**(2) (2004) 117–138