DRAMA, a connectionist architecture for control and learning in autonomous robots*

Aude Billard, Gillian Hayes
Dept. of Artificial Intelligence, University of Edinburgh,
5 Forrest Hill, Edinburgh EH1 2QL, U.K.
audeb@dai.ed.ac.uk
Contact:
Tel: +44-131-650-4496
Fax: +44-131-650-6899

*This paper is published in Adaptive Behavior Journal, Vol. 7:1, Jan. 1999.

DRAMA, a connectionist architecture for learning in autonomous robots

Abstract

This work proposes a connectionist architecture, DRAMA, for dynamic control and learning of autonomous robots. DRAMA stands for dynamical recurrent associative memory architecture. It is a time-delay recurrent neural network, using Hebbian update rules. It allows learning of spatio-temporal regularities and time series in discrete sequences of inputs, in the face of an important amount of noise. The first part of this paper gives the mathematical description of the architecture and analyses theoretically and through numerical simulations its performance. The second part of this paper reports on the implementation of DRAMA in simulated and physical robotic experiments. Training and rehearsal of the DRAMA architecture is computationally fast and inexpensive, which makes the model particularly suitable for controlling 'computationally-challenged' robots. In the experiments, we use a basic hardware system with very limited computational capability and show that our robot can carry out real time computation and on-line learning of relatively complex cognitive tasks. In these experiments, two autonomous robots wander randomly in a fixed environment, collecting information about its elements. By mutually associating information of their sensors and actuators, they learn about physical regularities underlying their experience of varying stimuli. The agents learn also from their mutual interactions. We use a teacher-learner scenario, based on mutual following of the two agents, to enable transmission of a vocabulary from one robot to the other.
Keywords: Time-delay recurrent neural network; Hebbian learning; spatio-temporal associations; unsupervised dynamical learning; autonomous robots.

1 Introduction

While adaptation is considered a fundamental capability for the survival of living agents, recent robotics research investigates how artificial agents may also benefit from it. Continuous learning is an important factor in the adaptation of an individual agent to its varying environment. Adaptation through continuous learning as opposed to evolutionary adaptation seems particularly interesting for autonomous artificial agents, especially if these learning capabilities could be used for teaching the agents new skills or increasing their a-priori knowledge. This work proposes a general framework of control architecture for autonomous agents that combines continuous learning with predefined abilities.

Robotics studies tend often, when developing a learning model, to address a particular problem of sensor-actuator coordination, e.g. maze travelling ([30], [39]), spatial navigation and exploration ([13], [14], [25]) or object manipulation ([1], [34]), where often only one direction of control (from sensor to actuator) is considered. In contrast, our approach tries to develop a single control architecture which enables a robot to learn and act independently of a specific task, environment or robot used for the implementation. For this we look at the common constraints and prerequisites for learning in a dynamic, noisy environment and propose an artificial neural network architecture for learning spatio-temporal regularities and time series in discrete sequences of inputs. Learning is based on Hebbian associations across all the robot's sensor-actuator modalities. The model does not differentiate between actuator and sensor information and treats them similarly during the associations. Correlations are thus performed from sensor to sensor, actuator to sensor, sensor to actuator and actuator to actuator. These correlations are then used both to determine the robot's actions (sensor to actuator), in order to predict the effect of the robot's actions on the environment (actuator to sensor) and to predict new sensor measurements from the current ones (sensor to sensor). The architecture is a time-delay recurrent neural network, using Hebbian update rules and a winner-take-all based neural activation function. We call it DRAMA for Dynamical Recurrent Associative Memory Architecture.

Different algorithms have been used to enable learning in autonomous mobile agents, e.g. Reinforcement Learning ([1],[29],[43],[44]), Genetic Algorithms ([13], [31], [32]) and more recently Artificial Neural Network (ANN) architectures ([26],[30],[39],[45],[46]). Neural network architectures are more interesting to us than RL and GA techniques as they require little knowledge of the task by not relying on the design of a good evaluation function for the robot's performance. A general problem with most ANN learning methods developed previously is that they are often computationally too heavy to enable on-line computation. On-line computation is a fundamental requirement for creating really autonomous agents ([40]). In our case, this was a requirement quite difficult to satisfy as we use LEGO robots with poor computational capacities. Our robots are provided with a micro-controller with 512k byte EPROM space and 128kS byte Static RAM. The CPU (central Processing Unit) is a Phillips 93C100 series 68000 compatible running at 30 Mhz. It has no facilities for floating points and all calculus has to be done with integers. Another disadvantage of these learning techniques is that they often require numerous examples, i.e. a long training phase, before performing well. Thus, training the robot and then testing its performances are often two separate phases. This is undesirable considering that continuous adaptation and *life-long* learning ([40]) are necessary qualities for autonomous agents which have to interact with fast changing environments (as it is the case for environments occupied by humans). Finally, another important skill for robots to possess (if expected to interact with other agents whether human or robotic) is the ability to communicate symbolically ([11], [22]). Following these desiderata, we design a control architecture, based on the DRAMA architecture, which allows continuous learning and control of autonomous robots.

Training and rehearsal of the DRAMA architecture require inexpensive and fast computation, which allows computation to be carried out on-line, i.e. on-board of the robot. We report here on the model implementation in two physical robotic experiments, for on-line learning of spatial regularities and time series of a robot's perceptions. In these experiments, two robots, a teacher robot and a learner robot, interact dynamically with their environment and with each other. On the one hand the learner robot learns spatio-temporal regularities in its perceptions, by recognising landmarks, i.e. learning the locations of objects, and by recording the time delays between its observation of each object. On the other hand the learner robot is taught by the teacher robot a vocabulary to label each of the landmarks. The learner robot grounds the teacher's words onto its own sensor perceptions. At the end of the experiment, the two agents share a common vocabulary to describe their environment, whose words are grounded onto each agent's distinct set of perceptions. Learning and rehearsal, i.e. training and retrieval of the DRAMA network, is performed continuously during the experiments, in order to record the robot's observations and to direct the robot's actions respectively.

The rest of the paper is organised as follows. Section 2 gives a mathematical description of the DRAMA architecture and section 3 evaluates theoretically and through numerical simulations the architecture's performance. Section 4 describes the experimental set-up of the experiments carried out to test the model, while section 5 reports on their results. Section 6 discusses the performance of the DRAMA architecture, as demonstrated by the experiments of sections 3 and 5. Section 7 concludes this paper with a short summary of the main results of this work.

2 Learning Model

The development of the DRAMA architecture was first inspired from the model of associative memory proposed by Willshaw [42], which is an abstract model of the hippocampus. Its development was driven by our wish to build a control architecture to enable *real time* control and learning in a physical autonomous agent. In particular, the choice of using a connectionist model and especially a Hebbian associative memory was driven by considerations pertaining to its implementation on a real robot with limited computational power. We require 1) fast computation for the system to react in real-time 2) robustness and adaptability in the face of varying environmental constraints[1], 3) as little built-in knowledge as possible to keep the system unspecific to a particular type of implementation (task, agent or environment).

2.1 The Willshaw net

The original version of the Willshaw net was developed as a model of biological associative memory [42]. It can be thought of as a fully-connected network with symmetrical connections, whose weights are updated following a basic Hebbian rule, i.e. only the weights of connections with co-active nodes are reinforced. The patterns consist of pairs of input-output bit-strings. The patterns are presented as arrays of binary (0/1) inputs. The learning stage begins with all the weights equal to zero. When an input-output pair is presented, for a binary-encoded input, the connection weight or intersection node between two activated units, i.e. one input and one output node which are both 1, is updated to 1. Whenever a weight has been updated to 1, it will never return to zero. The recall of a memorised pattern is done by counting the positive connection weights leading to each output unit. An input pattern is presented to the net. For each output column, the number of positive

[1] The variation of the environmental constraints are, in our experiments, e.g., changes in spatial distribution of objects, variation of lighting and electro magnetic field, and changes in the timing of sequence measurements, due to the variable speed of travel of the robots.

connection weights for each corresponding input line are counted. The output nodes which have a number of positive connection weights greater than or equal to the number of active inputs are activated.

The Willshaw net works very badly with noisy data because the net has no way to distinguish nodes that have been activated erroneously by noisy data from the correct ones. Graham and Willshaw [17] investigate different alternatives to the original retrieval function to improve the model capacity in the face of noisy data or sparse connectivity. They show that knowing the exact value of the *unit usage*, that is the frequency of activation of a connection or unit node during pattern storage, would greatly improve the robustness of the architecture. However, because they model the functioning of the human neural system, they could not assume that the biological network would have this information. In contrast, our concern is to define an artificial architecture of associative memory for robotic applications without necessarily a biological plausibility. Its robustness is an important criterion and to improve it, we defined [7] an update rule for the connection parameter so that an exact record of the connection usage, that is, of the frequency of correlated activation of any two units, is kept. We call this parameter a *confidence factor*. The model was similar to Willshaw's because it kept the basic principle of the training and retrieval algorithms of the original model. This resulted in a statistical type of network whose functioning was a mixture of the classical Hebbian network and the Willshaw network. Note that the Willshaw network was originally derived from Hebbian network. For a full discussion of its functioning and implementation, the reader may refer to [7].

The new extension we describe here adds recurrent connections to each of the nodes of the network, in order to make correlations between delayed and simultaneous occurrences of different input patterns. The uniform structure of the original network is changed for a fully recurrent, non symmetrical network, whose connections are associated with two weight parameters, recording separately the spatial and temporal features of the training patterns. As in the Willshaw model, a one-time-step training algorithm is used for updating the parameters, based on Hebbian rules, and a winner-take-all algorithm is used for retrieval of the unit activity. The resulting model is a simple version of a recurrent neural network (RNN) (as compared with a RNN using back-propagation and with hidden layers) that satisfies our basic requirements, namely fast computation for real time functioning and temporal associative learning capabilities. In section 6, we discuss in more detail the differences between our present model and other RNN models.

This section presents the complete extended version we have developed from the original Willshaw model. It is organised as follows: we first give a brief overview of the model functioning for controlling learning and behaviour of robotic agents (in section 4.3, we give a detailed description of the implementation of the architecture for the experiments we report here). We then give the mathematics of the DRAMA architecture.

2.2 Control and learning process

The DRAMA architecture provides a general control architecture framework for autonomous agents. Since the model's implementation was grounded in robotics experiments, our description of it will contain notions such as actuators or sensor systems. Actuators are, in the experiments, the motors and a radio emitter. Sensors are proprioceptive (an inclination sensor, an energy level checker and a compass) and exteroceptive (light and infra-red detectors, whiskers and bumpers). By extension, we will use the term sensor system for both sensor and actuator systems when differentiating between them is unnecessary, i.e. when processing of their information is independent of the type of system that provided it.

Figure 1 shows a schematic representation of the model, i.e. how the sensory information is

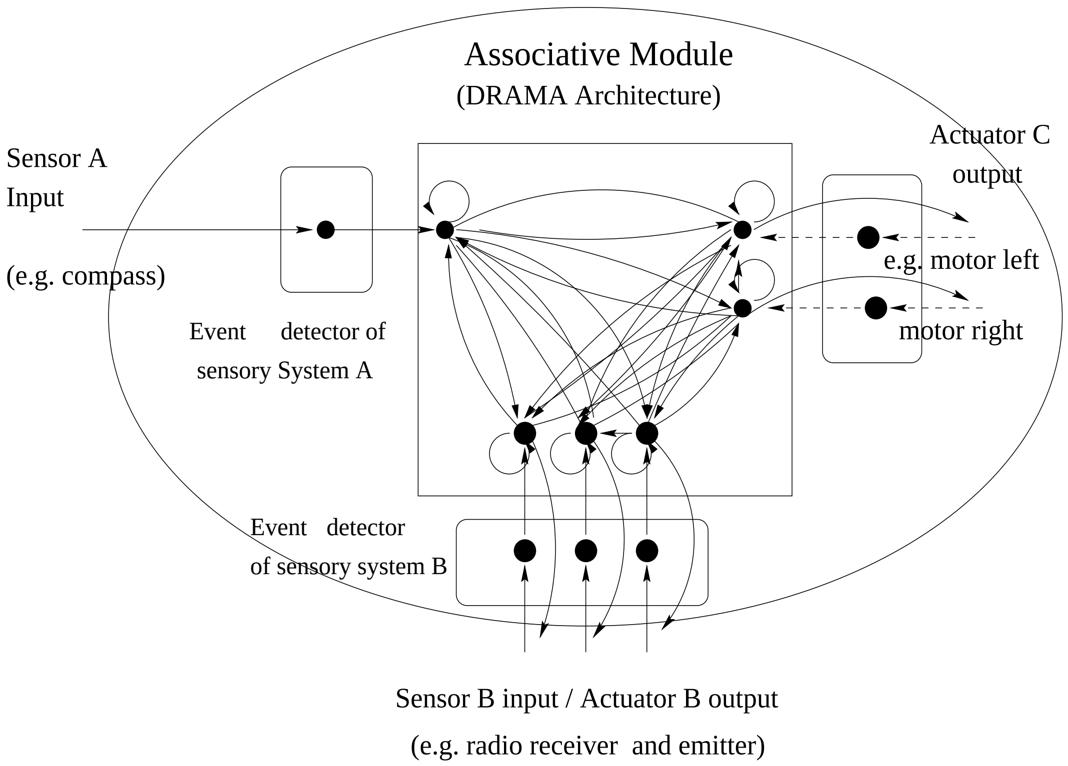

Figure 1: Schema of the robot's control system, using the DRAMA architecture.

processed through the DRAMA architecture, either for learning or for activating the actuators outputs. The structure of the system is composed of two parts: a preprocessing module of the data for *event recognition* and the DRAMA architecture. At each processing cycle, the sensor-actuator vector state is measured and its information processed through the event detector modules associated with each sensory system. Sensor-actuator inputs are presented as arrays of binary data (bit-strings) of different length for each system. Information from each sensor system is treated separately by each event detector module and, thus, an event is determined differently for each system. Each sensor is represented as a box with n input units, where the number of units associated with a sensor can vary from one sensor to another. When a variation in one sensor or actuator input has been measured (event), the novel information is forwarded to the associative architecture (DRAMA) to be correlated with all simultaneous and previously recorded events in other sensor-actuator systems. DRAMA is a fully recurrent neural network. Its recurrent structure provides a short term memory of the measurements. Long term memory is obtained by updating the internal connections following Hebbian rules. Sensory systems A, B and C in figure 1 could be interpreted e.g. as the motor, compass and radio systems of the robots. In the experiments reported in section 5, the radio signals are associated with different compass states, thus providing the agent with a 'vocabulary' (with different 'words' defined by different radio encodings) with which to express its direction of movement. The robot's actions are determined by retrieving the activity on the network connections to the actuators given a particular sensor-actuator state and inverting this into, e.g. motor speed.

Note that, in a bidirectional associative memory, such as DRAMA, the notions of input and output are interchangeable. They refer to the direction of retrieval of the associations. Thus, sensor and actuator information can be either input or output depending on whether the information is the trigger or the result of the retrieval. For instance, the actuator state is a DRAMA output when it has been determined by retrieval of the sensor to actuator association (control of the robot's actions) and a DRAMA input when its information is used to calculate the sensor state (prediction of the robot's perceptions).

2.3 Data encoding

As mentioned previously, a sensor-actuator state is encoded as a bit-string that is composed of a set of smaller bit-strings of different length, one bit-string for each sensory system. As we discuss in section 3.1, the model capacity decreases importantly when the pattern encodings overlap. Thus, in the experiments, we tried to encode all sensory information as orthogonal patterns, when this was possible. For example, information provided by the compass was encoded in a bit-string of length 8, where each bit would correspond to one of the 8 quadrants. Bit 1 corresponds to angle between 0 and 45 degrees, bit 2 to angle between 45 and 90, etc. Thus, each compass measurement would be represented by a pattern with one single bit activated. Such a representation of the data serves as a first classification of the sensory information into subclasses.

2.4 Event recognition

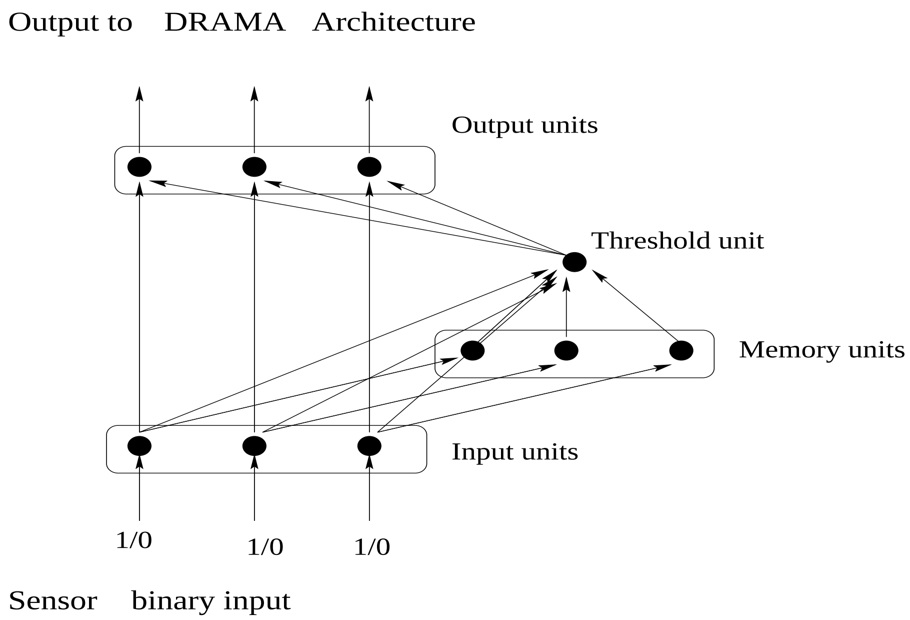

Figure 2: Event detector module

There is one *event detector* module per sensor. Each module receives n input units and outputs to n associated units in the DRAMA architecture, where n is the number of units of the particular sensor (see figure 2). The neuronal representation of the internal structure of the module is given in figure 2. Each input unit is connected to one *memory* unit, one *output* unit and to a *threshold* unit. Output $y_i^m(t)$ of the memory unit i at time t is simply the value x_i of the input unit i

at time $t-1$, $y_i^m(t) = x_i(t-1)$. Output y_i^{th} of the threshold unit is the result of the function $\theta(x, H)$ applied onto the difference between the input units and memory units outputs: $y_i^{th}(t) = \theta(\sum_{i=1}^{n} |x_i(t) - y_i^m(t)|, H)$, where the function $\theta(x, H)$ is a threshold function that outputs 1 when $x >= H$. Finally, the state of the output unit $y_i(t)$ is calculated as follows:

$$y_i = \theta(x_i(t) + y_i^{th}(t), 2) = \theta(x_i(t) + \theta(\sum_{i=1}^{n} |x_i(t) - x_i(t-1)|, H), 2) \quad (1)$$

The threshold H fixes the minimal number of unit inversions in the input before activation. For example, if $H = 1$, the threshold unit fires as soon as one input unit has changed from 0 to 1 and the output unit outputs 1 if it receives an input value $x_i(t)$ equal to 1, otherwise 0. The output vector is then equal to the input vector, once the threshold unit fires. In short, the output vector of the event detector is either equal to the input vector, if this is sufficiently different from the previous input (relative to the threshold of minimal variation), or a vector zero. So the result of this is that event units detect only $0 \rightarrow 1$ changes in the unit activation (and not the reverse). Note that, when using an orthogonal encoding for the sensor information, the event detector will be non zero only for $H <= 1$.

2.5 Associative module (DRAMA)

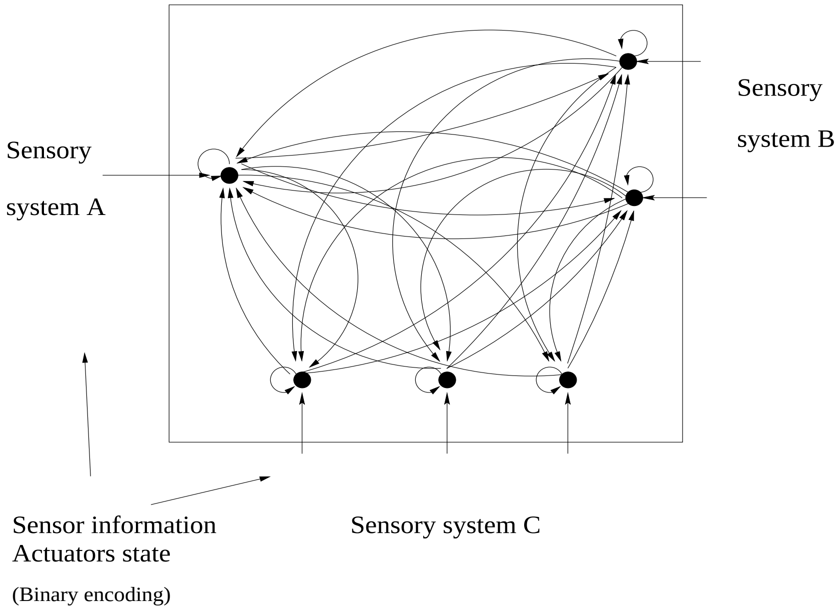

Figure 3: The DRAMA architecture.

The DRAMA architecture consists of a network composed of $\sum_{i=1}^{n} m_i$ units, where n is the number of sensors of the system and m is the number of input units associated with each sensor, different for each sensor. It is a fully connected recurrent network with non symmetrical connections, i.e. each unit is connected with all other units in the network and with itself (self-recurrent connections). Each unit also receives input from one output connection of the event recognition module. There are no hidden units. Note that, in the robotic experiments, the network is fully connected at the level of the sensory systems, that is all units i in sensor k are connected with all units j in sensor l ($l \neq k$); however, units inside the same sensory system are not interconnected (see figure 3). This was done in order to save the computation cost when running the system on-line, by reducing the size of the weight matrixes and the number of operations for training and retrieval of the network (connections among units inside the same sensor system would not have improved the learning performance in these experiments, as most patterns inside the same system were orthogonal and thus inside connections would not be updated). In the following, we present the equations of the network for the general case, in which all units are interconnected. In section 3, we analyse the network performance through numerical simulations, using a fully connected network.

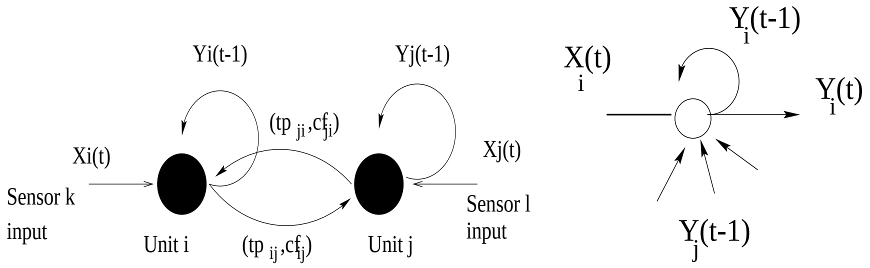

Figure 4: **Left:** Bidirectional connectivity of two network units. **Right:** One unit connectivity.

Similarly to time-delay neural networks [10, 27], each connection in the DRAMA network has two parameters associated with it instead of one: a time parameter (tp) and a confidence factor (cf) (see figure 4 left). Time parameters and confidence factors are positive numbers (real numbers in the simulation and integers in the physical implementations). They record respectively the time delay between and the frequency of two units co-activation.

2.5.1 Unit activation function

Output $y_i(t)$ of unit i at time t is a function of its input $x_i(t)$ at time t, its output $y_i(t-1)$ at time $t-1$ and the outputs $y_j(t-1)$ at time $t-1$ of all other units j (see figure 4 right). It is a real number whose value is comprised between 0 and 1. The equation is given in 2. Output of unit i is equal to the normalised sum of its input activation, its previous output activation decreased by a factor tp_{ii} and the sum of activation of other winning units, that is units which have passed the conditions encapsulated by the function G.

$$y_i(t) = F(x_i(t) + tp_{ii} \cdot y_i(t-1) + \sum_{j \neq i} G(tp_{ji}, cf_{ji}, y_j(t-1))) \tag{2}$$

where F, the transfer function, is the identity function for input value less than 1 and saturates to 1 for value greater than 1, $F(x) = x$ if $x \leq 1$, otherwise $F(x) = 1$, and G is the retrieving function whose equation is given below in equation 3 and explained in the following paragraph. The indices notation used in the equations should be interpreted as follows: cf_{ji} is the confidence factor of the connection leading from unit j to unit i.

2.5.2 Retrieval

When one input unit is activated, its activation is propagated through the internal connections of the network to all other units of the network. Unit i becomes active, i.e. $y_i = 1$ under the effect

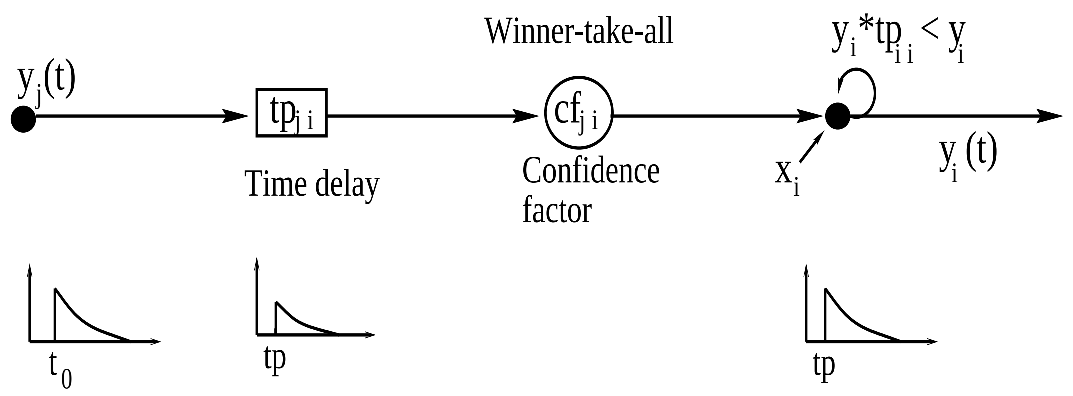

Figure 5: Propagation of the unit activity along the network connections.

of activation of unit j if function G applied on the output of unit j in equation (2) has value 1 for any unit j. The retrieving function G depends on the value of the connections parameters tp_{ji} and cf_{ji} and the output y_j of unit j and is defined as follows:

$$G(tp_{ji}, cf_{ji}, y_j(t-1)) = A(tp_{ji}) \cdot B(cf_{ji}) \tag{3}$$

$$A(tp_{ji}) = 1 - \theta(|y_j(t-1) - tp_{ji}|, e)$$

$$B(cf_{ji}) = \theta(cf_{ji}, \frac{\max_{y_j>0}(cf_{ji})}{T})$$

where $\max_k(cf_{ji})$ is the maximal value of confidence factor of all the connections between activated units j and unit i, which satisfy the temporal condition encoded in $A(tp_{ji})$. The function $\theta(x, H)$ is a threshold function that outputs 1 when $x >= H$.

The output of function G is equal to 1 when both A and B terms are equal to 1, otherwise it is zero. The temporal and spatial conditions represented by the A and B terms can be paraphrased as follows: 1) $A(tp_{ji}) = 1$ if the time delay for which the activation of unit j has been memorised before being correlated to the activation of unit i (this time delay is encoded in the value of $y_j(t)$, which decreases linearly with time when no new activation occurs, see *short term memory* paragraph) is equal to the time encoded in the time parameter tp_{ji} within an interval error e. 2) $B(cf_{ji}) = 1$ if the confidence factors cf_{ji} associated with the connection between one activated unit j and unit i, which satisfies the condition $A(tp_{ji}) = 1$, is greater than or equal to $1/T$ times the maximum confidence factor of other activated connections, $\max_{y_j>0}(cf_{ji})$. The effect of the two terms $A(tp_{ji})$ and $B(cf_{ji})$, and in particular of the threshold T and e, on the memory capacity will be discussed further in sections 3.1 and 5.3 and an algorithm for calculating the parameters T and e on-line will be presented in section 3.1.

Figure 5 shows a schematic representation of the propagation of unit activity along the network connection. The unit j activity passes the first filter on time, represented by the factor A in equation 3, when it has been activated for a time tp_{ji}. It then activates unit i at time tp, if it passes the threshold represented by the term B (winner-take-all mechanism on the cf connection parameters).

2.5.3 Short Term memory

The self-connections on the units of the network provide a short-term memory of each unit activation. If unit i receives no external activation from its input ($x_i = 0$) or other units' outputs ($y_{j \neq i} = 0$), then its output activity is equal to $y_i(t) = tp_{ii} \cdot y_i(t-1)$, that is it decreases by a ratio proportional to its temporal parameter tp_{ii}. Its value returns to 0 when the maximal decimal capacity of the system has been reached or, before that, if a limit of number of processing cycles has been set for keeping a record of the unit activity (that is fixing the duration of short-term memory). Therefore, once information from a sensor or actuator has triggered the event detector, it is then further memorised for a period M (fixed by the decrease of activation along the self-connections) during which it can be associated with any incoming event in any other sensor system. This results in a system capable of associating events delayed in time with a maximal time delay equal to the length of the short-term memory (STM), i.e. M. The effect of the value of M on the success of the learning in robotic experiments was discussed earlier in [2]. An algorithm for its update on-line is evaluated in section 3.4.

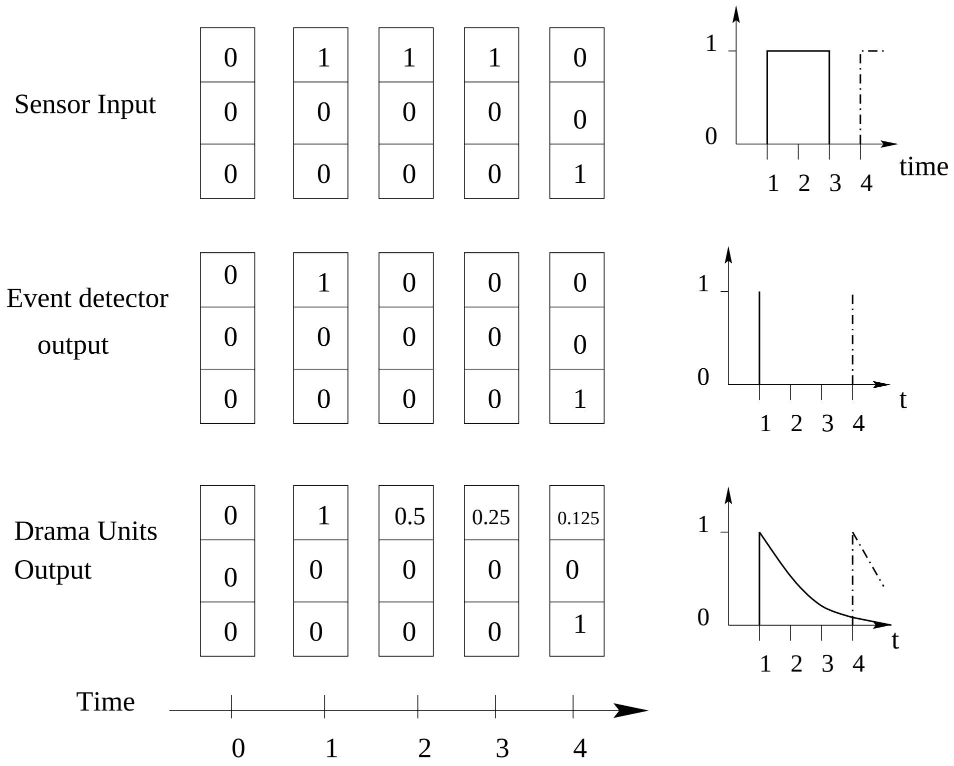

Figure 6: Transcription of sensor input into event detector output and DRAMA units output.

In summary, the output y_i of a unit i in the network takes values between 0 and 1: $y_i(t) = 1$ when (i) an event has just been detected ($x_i(t) = 1$) or (ii) when the sum of activation provided by the other units is sufficient to pass the two thresholds of time and confidence factor, represented by the G function. A value inferior to 1 represents the memory of a past full activation (value 1). For example, $y_i(t) = (tp_{ii})^{t-t'} \cdot y_i(t-t') = 0.9^3 \cdot 1 = 0.7290$ means that unit i has been activated since 3 time steps (when the decrease rate of the activation along the recurrent connections is equal to 0.9). Note that, in the experiments, all tp_{ii} are set equal to the same value, thus providing the same memory duration for all units.

Coming back to the complete control architecture, composed of the event detector module and the DRAMA module, which we presented in section 2.2, we show in figure 6 the transcription of a 4-time steps sequence of a 3-bits sensor input into the corresponding event detector output and DRAMA units output. We choose a ratio of decrease activation in the DRAMA unit self connection equal to 0.5. The diagrams on the right show the shape of unit activity for sensor, event detector and DRAMA units (straight line is the activity of unit 1 and dotted line is the activity of unit 3).

2.5.4 Training

When two patterns are presented to the associative memory, the connections between co-active units are updated; on the one hand, the confidence factors are incremented to represent the structural correlation between the input patterns and, on the other hand, the time parameters are updated to record the temporal delay between the two patterns' occurrences. The connection parameters are asymmetric and, thus, associations are directional. Training is dynamic and occurs each time the output of one event detector module is activated. Each input pattern is memorised for a period of M cycles through the self-connections (see explanation in section 2.5.3) and is correlated with all other patterns appearing during this period. Time parameters and confidence factors are updated following Hebbian rules: once a unit i is activated, i.e. its output is maximal: $y_i = 1$ (recall that the unit's output takes values between 0 and 1), afferent connections to this unit from previously or simultaneously activated units j, i.e. those for which $y_j > 0$, are updated. Only the connections directed to (and not from) the newly activated unit i are updated. Connections leading to the most recently activated unit are updated and this only during the first cycle in which this unit is fully activated. The update rules for each parameter are given in equations 4 and 5.

$$tp_{ji}(t) = \frac{tp_{ji}(t-1) \cdot \frac{cf_{ji}}{a} + \frac{y_j(t)}{y_i(t)}}{\frac{cf_{ji}}{a} + 1} \tag{4}$$

$$cf_{ji}(t) = cf_{ji}(t-1) + a \tag{5}$$

The time parameter *tp* records the time delay between the activation of the two units which are linked by the connections; the short-term memory mechanism causes all y_j values to decrease at each cycle by the same factor, as explained in *short term memory* paragraph, thus the ratio between y_i and y_j gives a notion of their relative delay of activation. The time parameter value is calculated as the arithmetic mean value of time delay over all training data and its value is between $(tp_{ii})^M$ and 1; the closer the two events, the bigger the time parameter; $tp_{ij} = 1$ when the two events are simultaneous.

The confidence factor keeps a memory of the frequency of a pattern's occurrence. Its value is incremented at each updating step by a fixed quantity a.[2] In the experiment, the increase of the confidence factor is linear following a fixed slope of value a.[3] At the beginning of the experiment, the values of confidence factors and time parameters are set to 0 for all connections apart from the self-recurrent ones, which have predefined values for these parameters determining the duration of short-term memory of the unit activation.

[2]In order to prevent the confidence factor values from becoming too large in the experiments, all values are rescaled by dividing by a factor of 100 when they reach the value of 1000; the increase factor a is also rescaled by the same factor to keep the same proportional increase between the time parameters.

[3]An interesting option is to make the slope proportional to the value of cf_{ji}, the more confident the greater the increase. This would speed up the learning and may increase the robustness of the model against noisy data, by giving a greater influence to nodes that are more often activated (see section 3 for a more general discussion of the robustness of the model).

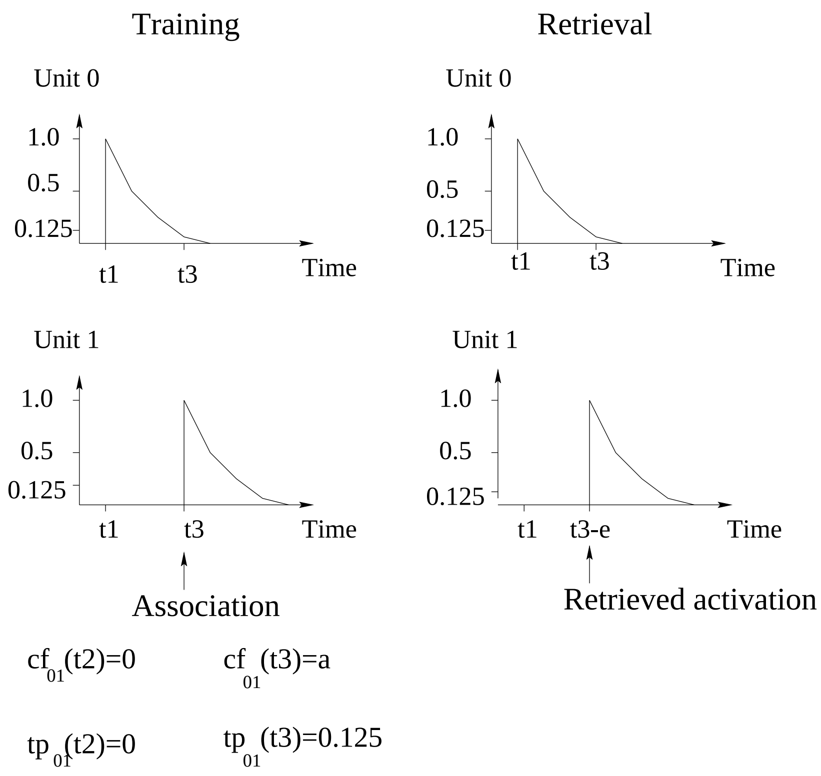

Figure 7: Example of association between unit 0 and unit 1.

In figure 7 left, we show an example of association between unit 0 and 1. Unit 0 is activated at time 0, while unit 1 is activated at time 3. Association is done at time three. The parameters of the connection from unit 0 (last activated) to unit 1 (most recently activated), i.e. cf_{01} and tp_{01}, are updated. Before association, the value of cf_{01} and tp_{01} are both zero (no correlation yet). After association, $cf_{01} = a$, where a is the increase factor of equation 5, and $tp_{01} = \frac{0.125}{1} = 0.125$, where 0.125 is the activity level of unit 0 output after three decrease steps (the ratio of decrease along the self connection is equal to 0.5). On the right hand side of figure 7, we show retrieval of activity of unit 1 after activating unit 0 at time 1. Following condition of factor A of equation 3, unit 1 is reactivated at time 3 minus e, which is the error on time delay between the two unit co-activation.

3 Model capacity and performances

The DRAMA architecture functions as an associative memory, which associates pairs of input-output patterns with delayed time of occurrence, leading to learning of time series. It has a fully recurrent structure, which provides a short-term memory of unit activation. The network's input patterns are thus recorded for a short delay, during which they are associated with any new input pattern incoming during this delay. The time delay between each pattern occurrence and the structure of unit activity of each pattern are learned separately in the two parameters attached to each network connection, namely the *time parameter* and the *confidence factor*. Once an input-output pair has been learned, presentation of the input to the net retrieves the output after the recorded time delay. Retrieval of the output units' activity results from a winner-take-all mechanism applied to the spatial and temporal structure of the pattern of input units' activity.

3.1 Capacity for storage of binary patterns without time

If the time delay between input and output patterns is constant, then the term A in equation 3 is always equal to 1, and thus the retrieval function G depends only on the term B, that is, it applies only to the spatial structure of input units' activity. In this case, the model is very similar to the Willshaw network. Correlated occurrences of two input patterns are distinguished from randomly generated ones by keeping a record of the frequency of correlated activation of these patterns' units (increment of the confidence factor (cf) parameter of the connection). A pair of input-output patterns is said to be correlated once the two patterns have been associated more often than a minimal noise threshold, corresponding to the threshold T of equation 3 in our case. Correct retrieval of each pattern of the pair, given the second one, depends on the proportion, relative to the above threshold, of correct over noisy associations each pattern has with other patterns.

Following Graham & Willshaw [16], we define the network *capacity* as the number of patterns that can be stored before there is one bit in error in the recall pattern output. When data are presented as binary inputs, as it is the case in our model, the proportion of active units overlapping between the training patterns is an important factor that limits the capacity of the associative memory (as discussed by [8, 16] for the Willshaw network). In [4], we evaluated the DRAMA maximal capacity to be equal to the number of units of the network, that is of order one of the network size. This result followed from the observation that the network could be trained on any input-output pairs, as long as no two pairs would overlap on both their input and output patterns. In other words, each unit can be activated only once in all pattern inputs (or outputs). It follows that there are at most N possible input-output pairs of patterns.

As pointed out by [8, 17], the capacity of a binary Hebbian network based on a winner-take-all retrieval mechanism, such as DRAMA, depends on choosing correctly the threshold of activation, which, in our case, are the thresholds T and e of equation 3. T should be sufficiently high to discard correlations, i.e. connections update, due to spurious unit activity, while sufficiently low to allow retrieval of patterns with different frequency of activation, i.e. different values of confidence factor for connections between patterns' activated units. e should be sufficiently large to include the maximal variation of time delay between two units co-activation, while sufficiently small to allow precise prediction of time of unit activation.

Estimating correctly the percentage of noise (spurious unit activity) in the system, and thus the correct values for T and e, before learning is often not possible, especially in unsupervised learning of robotic experiments. In [4], we presented an algorithm to calculate the values of these thresholds simultaneously to training the network. The idea was to use the information on the ratio of spurious/relevant units activation, reflected by the current values of the network connection parameters. The calculation was based on the assumption that incorrectly updated connections should have the lowest confidence factor values (not frequently updated) and the most important variation of time parameter values (no regularity in the time delay of units co-activation). The thresholds values were calculated based on a Gaussian estimation of the distribution of the parameter values of correct and incorrect connections, following equations 6 and 7.

$$\frac{\max_{\{y_i>0\}}(cf_{ij})}{T_a} = 2 \cdot \mathrm{mean}_{\{y_i>0\}}(cf_{ij}) - \frac{\max_{\{y_i>0\}}(cf_{ij}) + \min_{\{y_i>0\}}(cf_{ij})}{2} \tag{6}$$

$$e = \mathrm{mean}_{y_i>0}(e_{ij}) \ , \ e_{ij} = e_{ij}(t) = \frac{e_{ij}(t-1) \cdot \frac{cf_{ij}}{a} + |tp_{ij} - \frac{y_j(t)}{y_i(t)}|}{\frac{cf_{ij}}{a} + 1} \tag{7}$$

where $\max_{\{y_i>0\}}(cf_{ij})$, $\mathrm{mean}_{\{y_i>0\}}(cf_{ij})$ and $\min_{\{y_i>0\}}(cf_{ij})$ are the maximum, mean and minimum values of confidence factor (or of time parameter for $\mathrm{mean}_{y_i>0}(e_{ij})$) over all activated units at the time of retrieval. e_{ij} in equation 7 represents the maximal variation of the factor tp_{ij} for all activated units i, as recorded during training.

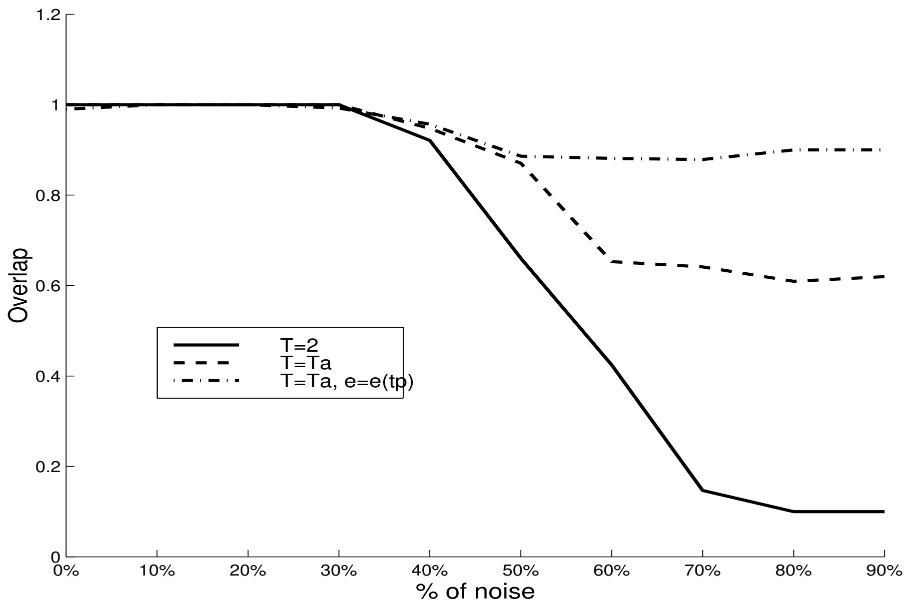

Figure 8: Recall performances of a $10x10$ units network for different values of noise and three different threshold strategies.

Figure 8 shows the result of numerical simulations, which evaluate the network recall performance, i.e. the overlap between recalled and trained patterns (mean value over all patterns), for a network of 20 units when trained at the maximum of its capacity and when varying the percentage of noise, i.e. the frequency of random activation of any unit during training. An overlap of 1 means that all patterns are perfectly retrieved. We compare the recall performance using three threshold strategies: (1) no threshold on time parameter tp (i.e. no factor A in equation 3) and a fixed threshold on cf, $T = 2$; (2) no threshold on time and time variant threshold on cf, $T = T_a$ (equation 6); (3) time variant thresholds on cf and tp, $T = T_a$, $e = e(tp)$ (equation 7). Results show that pattern recall is perfect up to a proportion of 30% of noise; otherwise the performance decreases up to a minimal proportion of 90% percent overlap between retrieved and training data with the third strategy. The performance is in average better with the time variant strategies, as it allows better recall with a bigger proportion of noise. It is especially better when introducing the threshold on time.

Capacity for storage of binary patterns with time

Note that the introduction of the time parameter improves greatly the capacity of the network, as compared with the case when one uses only the confidence factor parameter (see discussion of previous paragraph). In [4], we determined the network capacity to be equal to $N \cdot (N - 1)$, where N is the number of network units. That is, the maximal capacity of the network, given the complete retrieval function G, see equation 3, is of order two of the network size. The reasoning was based on the observation that the supplementary information given by the time parameter allowed to distinguish between patterns involving the same unit pairing. In other words, each unit of the network can now be paired with all other units apart from itself (while in the previous case, it could be paired only with one other unit).

3.2 Space and time efficiency

An important characteristic of the DRAMA architecture is that it is computationally fast and inexpensive. We here evaluate more formally what we mean by this.

Training of the network requires a number of time steps of computation equal to two times the number of connections to update the connection parameters cf and tp. This is then faster than usual backpropagation algorithm which requires n times the number of connections, where n is the number of steps needed to reduce the error to a minimum and is often bigger than 2. Retrieval of unit activity (equation 2) is also relatively fast, as it does not require calculation of derivatives, which needs several time steps of computation. Retrieval requires summing twice over all units in order to first calculate the mean, maximal and minimal values of the cf and tp parameters which determine the values of threshold factors and to then sum the vote for activation of each unit.

Good time efficiency of a neural network model is often counterbalanced by poor space efficiency. The decrease of the model's capacity in front of overlapping pattern encoding leads to a poor use of the space efficiency (see section 3.1). The maximal number of patterns, onto which a network of N units can be trained, is equal to N^2. This number is much inferior to the maximal number of combinations which can be formed with N units, which is equal to $N!$. Space efficiency is also determined by the number of global variables which are used by the network. The DRAMA architecture requires space for six times the number of connections to store the three parameters associated to all connections of the network. This is a higher number of global variables than that required by most NN, which use only one parameter per connection.

3.3 Sequence learning

In the previous section, we discussed the model performance at associating pairs of input-output patterns. We consider here the model performance at learning sequences of more than two patterns. For this discussion, we consider a variation of the DRAMA retrieval algorithm described in section 2.5.2. In addition to the two conditions on the confidence factor and time parameters encapsulated in the function G in equation 3, we add a third condition which requires that all units activated at the time of retrieval agree on the activation of the output unit for the latter to be activated. This condition is similar to that of the winner-take-all algorithm used in the original version of the Willshaw network (see section 2.1).

The present version allows us to point out the two following facts: 1) the duration of the sequence that can be learned by the network is not restricted to that of the short-term memory M, as by transitivity of the associations a sequence of n steps can be derived from the association of shorter consecutive sequences; 2) the sequence is not restricted to include only strictly different patterns, but can be composed of several occurrences of the same pattern(s) (e.g. the sequence $ABCDEFCDG$ with repetition of the subpattern CD). This second result is due to the third condition, which requires the agreement of all voting units for the activation of the output pattern. Let us consider the sequence example $ABCDEFCDG$, with the following timing $A(t) \to B(t{+}1) \to C(t+2) \to D(t+3) \to E(t+4) \to F(t+5) \to C(t+6) \to D(t+7) \to G(t+8)$ and a short-term memory duration equal to 3 steps, such that the following associations are being made among others: $B \to E, F \to G$. Activation of patterns E and G after activation of the subgroup CD is determined by the vote of patterns B and F respectively, in addition to the votes of C and D.

Correct retrieval of sequences which loop on one of the patterns or on a subgroup of patterns creates specific conditions for the patterns' structure and timing of the sequence. Let the training sequence be of the form $ABCDEFCDG$, then learning is successful if:

1. The memory duration M is long enough to allow association between the patterns preceding and following the subsequence on which the sequence loop (i.e.B, E and F, G in the example).

2. If the number of occurrences of the same subpattern in the sequence is inferior to the threshold on confidence factors T so that it becomes activated: i.e in the example if $N_{CD} = 2 < T$, so that $\frac{cf_{BE}=cf_{CE}=cf_{DE}}{cf_{CD}} >= T$.

3. If the pattern or subgroup of patterns on which the sequence loops does not occur at the beginning of the sequence, as a pattern previous to the loop is necessary to determine the activation of the correct subsequent pattern.

Note that learning a sequence with an internal loop is equivalent to learning two sequences with a common subpattern (e.g. $ABCDF$ and $FCDG$). Previous results imply then that the number of sequences that can be learned is not restricted, as long as the structure and temporal pattern of occurrence satisfy the previously mentioned conditions on M, e and T.

3.4 Learning algorithm for the short-term memory parameter

The previous sections pointed out the importance of correctly choosing the values for the three learning parameters of our system, namely T, M and e and determined bounds for these values relative to the structure of the training patterns and the percentage of noise in the system. However, as mentioned earlier, it is seldom the case that we can access this information before learning. Therefore, it is desirable to define a learning algorithm for tuning these parameters on-line, that is together with the associative learning process. In section 3.1, we gave a possible example of how to calculate on-line the values of the thresholds T and e. M can also be updated on-line according to the following algorithm:

$$\text{If } e \leq 0.1 \cdot M \text{ then } M = M + \sum_i (p_i - y_i) \tag{8}$$

where p_i is the predicted activation of unit i. M is updated only once the error on time e has settled to a small value. M is either increased or decreased depending on whether there has been more unpredicted unit activations (a zero activity prediction while measuring a unit activation) than incorrectly predicted unit activations (a non zero activity prediction while measuring no unit activation).

Performance of the learning algorithm was evaluated through simulation for learning two types of sequences, namely $ABCDE$ and $ABCDEFCDG$, for different noise proportions (the variation of the time delay of activation of each pattern) and with different starting value for the memory duration (10 times lower or bigger than the correct one). Results showed that the algorithm had converged after less than 50 trials up to 40% and 20% (by respect to each sequence) of noise. Statistical fluctuations around the correct value of M were observed when learning the sequence $ABCDEFCDG$ with more than 20% of noise. Figure 9 top shows the variation of the parameters e and M and the error (number of incorrect prediction) while learning the sequence $ABCDEFCDG$ with 20% of noise (variation in the timing of pattern occurrence) in the input. Convergence of the parameters values is achieved after 36 presentations of the sequence, that is the error is equal to zero. The time threshold has settled at a minimal value of 2, which is equal to the maximal variation in the timing of pattern occurrence (noise), and the value of the short-term memory has settled to a value comprised between the minimal and maximal value required by the conditions of section 3.3. Figure 9 bottom shows superposed plots of the retrieved (straight line) and training (dotted line) pattern activation when learning the sequence $ABCDEFCDG$ under 20% of noise.

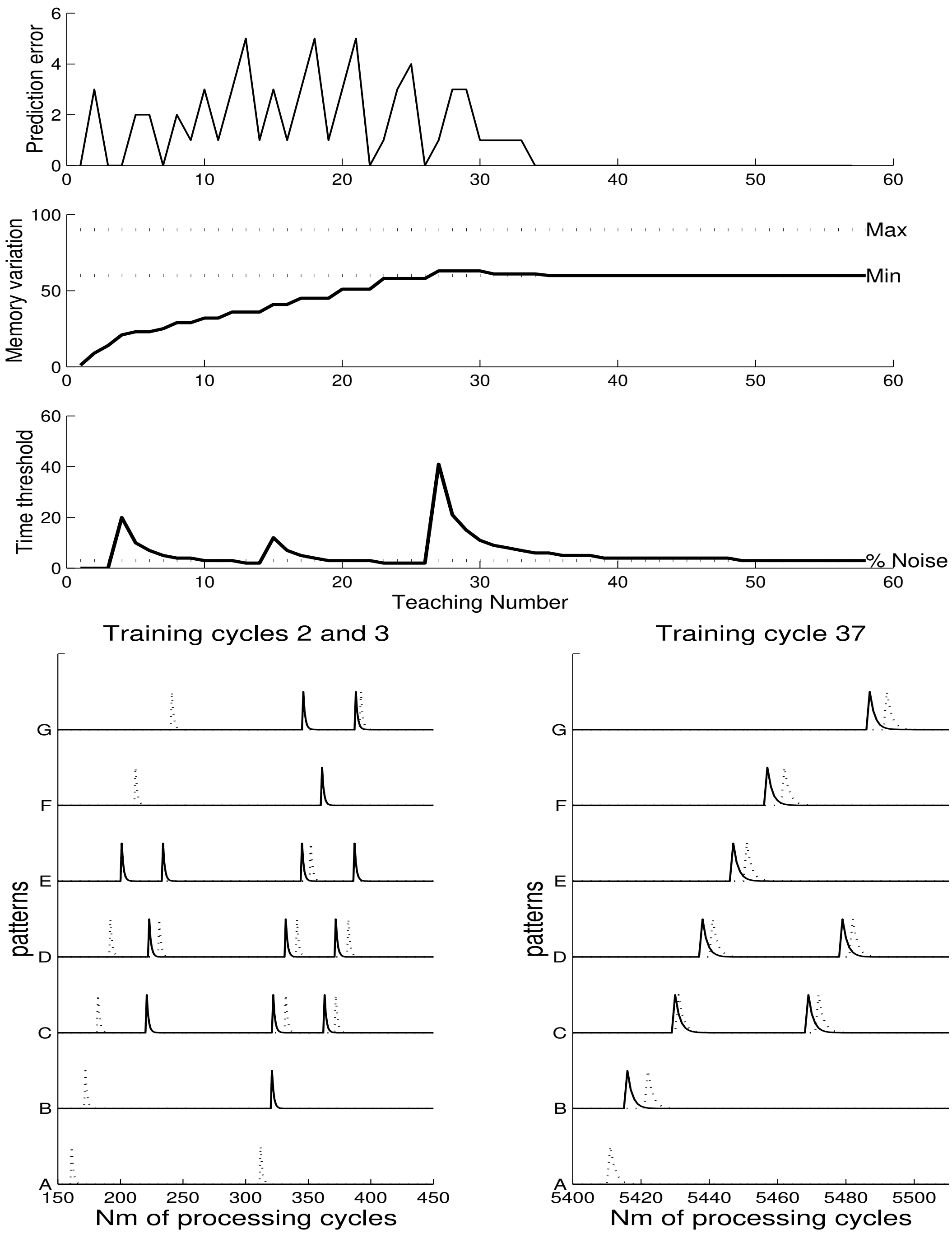

Figure 9: **Top**: Variation of short-term memory, error and time threshold during the training of the sequence $ABCDEFCDG$. **Bottom**: Superposed plots of the retrieved (straight line) and training (dotted line) pattern activation along the training; the figure shows snapshots of (left) the three first cycles of the training and (right) the 37th training cycle, that is after convergence.

The figure shows snapshots of (left) the three first cycles of the training, i.e. before convergence, and (right) the 37th training cycle, i.e. after convergence. The retrieved pattern activity begins only at the second cycle for pattern C and is incorrect in the third cycle as patterns E and G are activated twice instead of once. Retrieved and training pattern activity match for all patterns, apart from A, in the 37th cycle. Pattern A is the activation pattern for retrieval of the series; pattern A is not retrieved as it has not been correlated to any other pattern (the time lag between the end of one series, G pattern, and beginning of a new one, A pattern, is too long for A and G to be associated). The retrieved activations of patterns B, D, E, F and G occur slightly earlier than the training ones, in the margin of the 20% of noise (the variation of the time delay of activation of each pattern).

3.5 Summary

The principal properties of the DRAMA architecture can be summarised as follows:

Model structure and functioning
1) It consists of a fully connected network with self-connections on each unit and no hidden units. 2) To each connection in the network are associated two parameters: a time parameter and a confidence factor. 3) The structure of the network is dynamically updated, each time a unit has been activated by an external input (see table 1 for a complete description of the learning algorithm). 4) Time parameters and confidence factors are updated following Hebbian rules, providing an associative type of learning; the time parameters record the time delay between units' activation while the confidence factors keep a memory of the frequency of units' co-activation. 5) The self-connections on the units provide a short-term memory of the activation of the unit; the duration of the memory is fixed by the ratio of decrease of the activation along the recurrent connection. 6) The short-term memory of unit activation enables associations between patterns of unit activation that have been delayed in time, which leads, by transitivity of the associations, to learning sequences of patterns of unit activation. 7) Data retrieval depends on the value of the time parameters and confidence factors associated with the connections, which act as separate filters on the spatial and temporal features of the input; output units are activated when the two following conditions are satisfied: (i) the time delay since the input's time of occurrence is equal to the memorised temporal correlation and (ii) the confidence factor values of all active input units are greater than a fixed percentage of the maximal value of confidence factor of all active units in the network at the time of retrieval.

Model performances
1) The capacity of the network decreases importantly with overlapping encoding of the data; an orthogonal encoding is then preferable when possible, however leading to a poor space efficiency. The maximal capacity of the network of order 1 of its size for storing pairs of input-output patterns with fixed delay and of order 2 of its size for storing pairs with variable time delay. 2) Using the full capacity of the network, retrieval performance is perfect up to a proportion of 30% of noisy data; otherwise the performance decreases up to a minimal proportion of 90% overlap between retrieved and training data. 3) The model can learn sequences of pattern activation of the following types: (i) $ABCDEF$, i.e. a n steps sequence whose duration can be longer than the short-term memory, as it can be derived, by transitivity of the associations, from the association of shorter consecutive sequences (ii) a sequence composed of several occurrences of the same pattern(s) (e.g. $ABCDEFCDG$ with repetition of the subpattern CD or $ABBBCD$ with three occurrences of the pattern B). 4) Retrieval depends on correctly choosing the values of the three learning parameters, namely short-term memory duration, threshold on time parameter and threshold on confidence

Table 1: Training algorithm

1:	Present an input I to the system. Compute the output of the corresponding event detector following equation 1.
2:	Compute output y_i of all units i of the DRAMA network, according to equation 2.
3:	Update the connection parameters of the DRAMA network: If $\exists$ i and j (units of the DRAMA network), s.t. $y_i = 1$ and $y_j > 0$, **update** the parameters cf_{ji} and tp_{ji} of the connection **from unit** j **to** i according to equations 5 and 4.
4:	Update the learning parameters T, e and M according to equations 6, 7 and 8.

factor parameter; these values depends on the proportion of noise (imprecise sequence timing and spurious unit activation) and on the sequence type; theoretical boundaries are determined for these parameters. 5) An algorithm for tuning the learning parameters simultaneously to updating the network connection parameters is defined and validated through numerical simulations. 6) Finally, because training uses a one time-step algorithm, the model is computationally fast and inexpensive, which allows its implementation for real time computation and on-line learning in a basic hardware system. We describe such an implementation in the following two sections.

4 Model application to robotics experiments

The DRAMA architecture provides a general framework of control architecture for an autonomous robot. It allows on-line learning of spatio-temporal regularities across the multiple sensor-actuator modalities of the robot, that is learning of time series of sensor-actuator, actuator-sensor, sensor-sensor and actuator-actuator inputs. It provides dynamic control of the robot's behaviour through retrieval of learned or predefined sensor-actuator sequences. Basic behaviours can be determined by fixing the connections of the DRAMA network between specific sensor and actuator systems of the robot. The DRAMA architecture is general in the sense that its structure and functioning make no prerequisites on the type of robots, i.e. the robot's sensors, actuators and body structure, which should be used.

We report here on two sets of experiments, in which we study each aspect of the architecture, namely its capacity for spatial association across multiple sensor modalities (first experiment), and its capacity for learning time series of sensor stimuli (second experiment). Experiments are carried out with two autonomous mobile robots, a teacher robot and a learner robot, in both simulated and physical environments. Both robots are controlled by the DRAMA architecture. Simulation studies are first carried out in order to determine the feasibility of the experiments by studying their successes and failures in a more reliable environment. Results of physical experiments are then compared to that of simulations.

4.1 The experimental procedure

In the following, we describe the experimental procedures of each experiment separately.

4.1.1 Labelling landmarks

In the first experiment, the learner robot learns to distinguish between different objects by attaching to them different labels, i.e. names. The robot is taught by a teacher robot. Teaching occurs as part of a teacher-learner scenario based on an imitative strategy, namely mutual following of the two agents[4]. While the two agents wander randomly in the environment, following each other, the teacher sends signals, i.e. 'words', to describe its novel perceptions, i.e. recognition of objects. The learner attaches a meaning to the teacher's signals in terms of its own perception of the object's features. Teaching is provided by a robot in the simulations and by a human instructor, who holds a lamp which the robot follows, in the physical experiment. Top of figure 10 show a graphical representation of the simulated environment and a picture of the physical one for this experiment.

The experimental procedure for this experiment consists of letting the robots wander in the environment for a given time period. During its exploration, the learner robot learns to map the environment as a set of landmarks, by associating the set of features representing each object with a particular location. The simulated environment is composed of three hills and two boxes. Each object is defined by a unique set of features: two of the hills have the same inclination but different colours, while the third hill has a different inclination but the same colour as the second one. The two boxes have different colours but the same shape. The simulated robots can perceive the objects' features using light detectors (for the colours), infra-red detectors (for the shape) and inclination sensors. Boxes are distinguished from the walls encircling the arena by their shape, i.e. upper infra-red detector response signals the box, while the walls produce a response in only the lower one. The robots can also locate the objects in the environment relative to polar coordinates. At each processing cycle, they calculate their position relative to the middle of the arena in terms of distance, given by the measure of an internal energy sensor, and angle, given by the measure of a compass.

The physical environment is composed of two boxes, which the robots can perceive through the flickering of side whiskers, and of one aluminium foil lying on the ground, which the robots can perceive using light detectors fixed under the chassis.

4.1.2 Learning time series of perceptions

In the second experiment, the learner robot learns time series of sensor measurements, while travelling in a highly regular environment. The environment consists of a series of three inter-connected corridors at right angles from each others (corridors are delimited by walls on each side). In the middle of the second corridor lies an aluminium plate, which the robot can detect with light detectors. Bottom of figure 10 show a graphical representation of the simulated environment and a picture of the physical one for this experiment.

The experiment consists of letting the two robots travel several times across the three corridors. The two robots follow closely each other, the teacher in front. During a run, the robots travel ten times the series of corridors. While travelling along the corridors, the robots perceive different light and compass measurements; when crossing over the aluminium plate, the robots perceive an increase of light intensity measurement in the light detectors, which they carry underneath its body. Because the corridors are placed at right angles to each other, travelling in each corridor corresponds to measuring a compass value, which refers to a different quadrant (which we call 'South', 'West' and 'North'[5]).

[4] The following scenario results in the follower agent implicitly imitating or replicating the followed agent's movement in the 2-D plane.

[5] Note that the robots measure a distorted component of the earth magnetic field, due to the noisy magnetic

In addition, the learner robot perceives three different radio signals, while travelling in each of the three corridors. These signals are sent by the teacher robot; they represent labels for the two compass measurements made in first and third corridors (signals 'South' and 'North') and for the increase of light, measured in the second corridor when crossing over the aluminium plate (signal ('Object'). Assuming that the robots travel in the corridors with the same average speed from one circling to the other, we expect the learner robot to perceive the following series of sensor stimuli when travelling across the three corridors: *Radio signal 'South' - compass measurement 'South' - radio signal 'Object' - compass measurement 'West' - measurement of light increase - radio signal 'North' - compass measurement 'North'*.

4.1.3 Unsupervised learning strategy

In both experiments, only the learner robot is actually learning. The teacher robot's role consists of 1) directing the learner robot, by the learner following it and 2) of teaching the learner robot a vocabulary to label the objects, by the teacher sending radio signals each time it encounters an object and by the learner robot attaching the teacher's signals to its own perception of the objects.

The teacher robot's knowledge of the vocabulary is predefined, that is correlations between radio signals and the corresponding objects' features are set-up from the start in the teacher's DRAMA network (see explanations of section 4.3). The two robots' following is mutual and results from phototaxis behaviour, as each robot carries a bright light and light detectors. When the robots are sufficiently close to detect each other, they stop their random wandering and align one behind the other one (teacher in front). Then, following one another, they go on wandering in the environment. The following stops only when one robot is distracted, e.g. when avoiding a complicated set of obstacles or when attracted by another bright point in the environment (e.g. windows). Each time the teacher robot perceives one of the objects (hill, box or aluminium plate), it emits the corresponding radio signal. The learner robot then *grounds*, i.e. gives meaning to, the teacher's signals by associating them with its own observations, that is, its own sensor measurements. In the first experiment, the teacher robot teaches a vocabulary of five and two words in the simulation and physical experiment respectively, one word for each object in the environment. Each word is a different radio signal which has to be associated with the particular sensor combination that describes the features of the corresponding objects (box=colour+shape, hill=inclination+colour). In the second experiment, the learner robot learns a three words vocabulary, two words for two different compass measurements (North and South) and one word to label the aluminium plate of the second corridor.

While bounded by the following process, learner and teacher agents are set in a position from which they share a common set of perceptions. They share a similar view of the environment (face the same direction) and thus a similar but not identical (due to different sensor sensitivity) set of external perceptions. They also share similar internal perceptions, as they travel the same path (similar energy consumption and inclination). This implicit similarity in the two agents' perceptions is what enables the learner to make sense of the teacher's words (the teacher talks only of what it senses, not aware of the learner's perceptions). It is thus an unsupervised teaching strategy. However, because of the spatial displacement between the two agents due to the following, teacher and learner's observations of the landmarks are delayed relative to one another. The follower perceptions become similar to the leader's one when the follower has travelled the distance of about a body length, that is when it reaches what was previously the leader's position. Learning is then successful because the DRAMA architecture associates not only simultaneous but also

emissions of the laboratory machines; thus the labels South, West and North do not always correspond to their usual meaning.

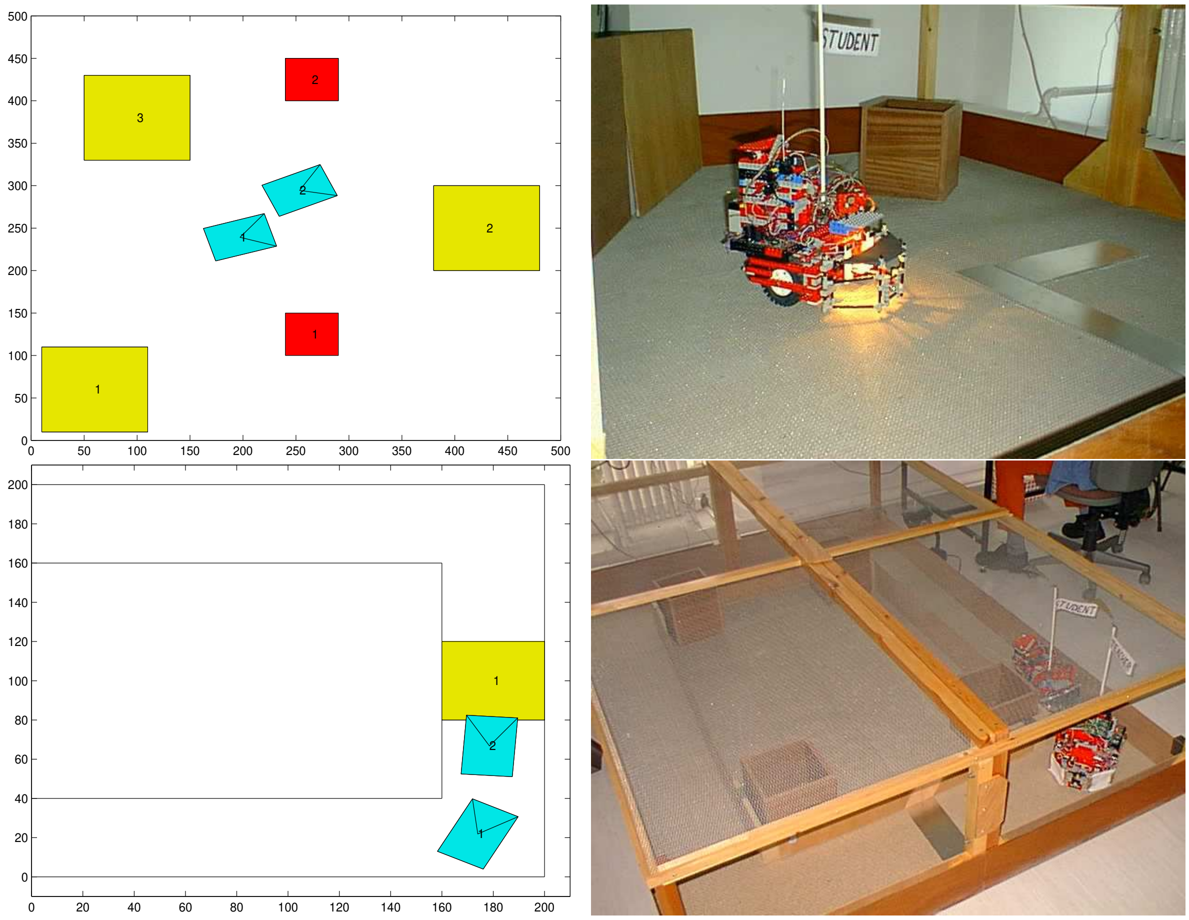

Figure 10: Simulated (left) and physical (right) environments of the first (top) and second (bottom) experiments.

Table 2: Table of the robots' sensors sensitivity and information encoding in physical and simulated experiments.

	Physical Exp.		Simulations	
Sensor type	Sensitivity	Encoding	Sensitivity	Encoding
Bumpers	0cm (touch contact)	active/inactive (1 bit)	None	-
Whiskers	0-15cm lateral	active/inactive (1 bit)	None	-
Infra-red	15^0 and 0-40cm	2 levels (2 bits)	180^0 and 0-45cm	8 quadrants/bits
Light front	180^0 and $0 - 10^5$ Lux	2 levels/sensor (4 bits)	180^0 and 0-45cm	8 quadrants/bits
Colours	None	-	3 types	3 bits
Light/Incl.	$0 - 10^5$ Lux	2 levels (2 bits)	3 levels	3 bits
Radio	418 MHz, whole arena	8 bits	whole arena	8 bits

delayed (successive) sensory stimuli. Observations of the landmark features are extracted from the continuous flow of sensor measurements by the *event detector* modules (see section 2.4), as they produce a change in the robot's measurements of inclination, light, infra-red, compass and energy level sensors. Each new incoming sensor information is then memorised for a fixed period of time, which corresponds to travelling about twice the robot's body length in the first experiment and one and a half corridors' length in the second, during which it is associated with all events measured in any other sensor system. In the experiments, the learner agent memorises the teacher's radio signals (perceived as novel radio stimulus) which it receives earlier than the corresponding sensor observation and then associates them with all sensor and actuator events it measures during a period of time following the signals' occurrence equal to the memory duration.

4.2 The set-up

Two autonomous LEGO robots (teacher and learner) are used for the real experiments. Each robot is equipped with one frontal infra-red sensor and bumper to avoid the obstacles (see pictures in figure 10 right). They also have two sets of light detectors, one set in the front (learner) or in the back (teacher) to follow each other and one set underneath their chassis to detect an aluminium-covered region of the arena. In addition, they carry a compass which measures bearings of 45 degrees. The range and sensitivity of the sensors are given in table 2. They have a radio transceiver, the means of transmission of the communication signals. Each signal is encoded in one byte with only 1-bit activated (e.g. 'North' = (01000000), 'South' = (00100000)).

The arena consists of a rectangular cage of 2.5m by 2m by 0.5m, in which the robots are continuously recharged similarly to the system used in the 'Dodgem' (bumper cars) game. Roof and bottom of the arena are electrified, creating a potential difference of 10V between them. The robots carry a long stick touching both ends of the cage from which they receive the current to power their battery and light bulb. In the picture 10 (top right), we see the learner robot in front of the aluminium covered area, and behind it the box and cardboard. In the picture 10 (bottom right), we see the teacher robot followed by the learner robot, managing the corner between first and second corridor. We estimate that about 20 to 30% of the sensor measurements are noisy: 80% of the radio transmissions are correctly received (i.e. if a signal is received, then it is perfect; the noise corresponds to the case where an emitted signal has not been received). The quadrants given by the compass are correctly detected in all cases (we did not observe any influence of the magnetic fields produced by the motors and the powering of the cage). The aluminium foil and the boxes are correctly detected in about 90% of the cases (limited sensitivity of the light detectors and flickering of the whiskers).

Simulation studies are carried out in a 2-D simulator, whose graphical representation is made using the MATLAB environment. The simulated environment consists, in the first experiment, of a rectangular arena measuring 500 by 800 units (see figure 10 top left) and, in the second experiment, of a series of corridors (figure 10 bottom left). In the graphical representation of the simulator (figures 10 left), the robots are represented as rectangles of 30 by 20 units, a triangle indicating the front; hills and boxes are represented as big rectangles on the sides of the arena and small squares in the middle respectively. In the figure 10 (top left), we see the teacher robot (No 2) followed by the learner robot (1), moving between hill 2 and 3. The simulated robots are provided with colour vision (three colours), infra-red vision to see the walls and detect the boxes, an inclination sensor to detect the hilly region (which is equivalent to the light detectors underneath the robots' body in the physical implementation). They carry also a radio transceiver to communicate, a compass that measures bearing of 45 degrees and an energy sensor that gives a relative measure of the travelled distance. The energy sensor value is incremented, at each cycle, by a factor proportional to the robot's speed (note that robot's speed varies depending on whether the robot crosses the plane or a hill, that is, its speed is slowed down or accelerated when it is climbing up or down the hill). Infra-red and light detectors are associated with a cone of vision of 180 degrees, which is segmented into eight quadrants. The measurements of these sensors are given by an 8-bit string where each bit corresponds to the values measured in each of the eight quadrants (e.g. infra-red=(11000000) stands for an infra-red activation of the first two quadrants). The range of sensitivity of the sensors is given in table 2.

The robots' behaviours are calculated in the simulation by the same routines as used in the physical robots, that is, there is one network (DRAMA architecture) per robot and the same retrieving and updating functions (equations 2, 4 and 5) are applied sequentially to them for determining the behaviour and learning of each of the robots independently. Code is written in C and is processed serially[6]. In order to produce a more realistic simulation, the following behaviour is made imperfect. Following is mutual, each agent aligns towards the other one on the basis of its light measurement. Similarly to what happens in reality, an agent is able to determine the position of the other agent with respect to its own with a precision of 20 degrees. Therefore, the alignment of the two robots is imprecise, which results in differences between the two agents' perceptions. This accounts for most of the noise, i.e. incorrect teaching-observation associations, which occur in the physical experiments. However, unlike what happens in physical reality, we did not simulate imprecision of the sensor measurements nor did we vary the external conditions (intensity of light or infra-red emissions) in the course of the experiments, an effect that can be observed in the real world. Instead, randomness was introduced in the calculation of the robot's movements in order to represent the imprecision measured in the real robot's movements. In addition, in the first experiment, the robots' direction of movement was reset to a random value every 1000 cycles (after about one complete cycle in the arena) in order for them to cover the space homogeneously, such as to approach each object from different directions (to avoid cyclic behaviour). In the real world, this occurs naturally as the effect of light variation, resulting, e.g., in the two robots to be suddenly attracted by one of the room corners or to loose sight of each other and then wander randomly in other directions.

4.3 The internal processing

Figure 11 gives a schematic representation of the processing of the sensor-actuator information through the DRAMA architecture. We restrict the schema to 4 sensor systems (radio, motors, compass, infra-red sensor) for reasons of clarity of the picture. In the simulations, 7 sensor systems (radio, motors, compass, inclination, IR, light (3 colours), energy) are used. At each processing

[6]The C programs for the simulations were run on Ultra 1 Model 140s SPARCstations.

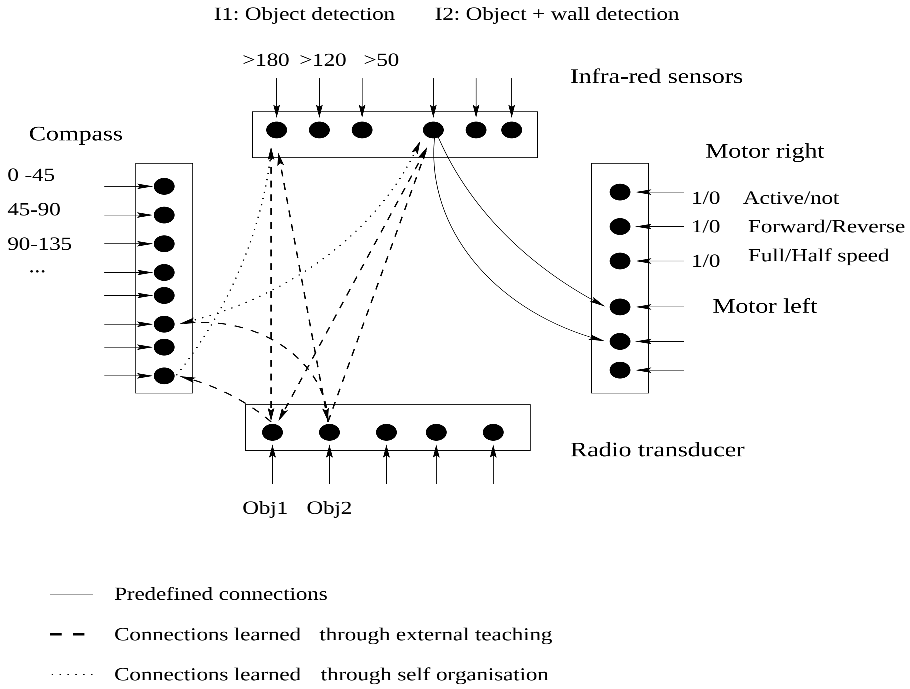

Figure 11: DRAMA connections state in the experiments.

cycle, the DRAMA network output to the robot's motors is calculated, in order to determine the motor activity, which is defined by the activity of the DRAMA motor units. The motor activity is encoded in a 3-bit string. Bit 1 determines the state of activity of the motor ($active/not = 1/0$), bit 2 encodes for the direction ($forward/reverse = 1/0$) while the third bit determines the speed ($full/half = 1/0$). Basic behaviours, such as obstacle avoidance and mutual following of the robots (phototaxis with light detectors) are predefined by setting the connection parameters (namely the confidence factors and time parameters) between the infra-red (IR) and light detector systems and the motor system. In order to perform a purely reactive behaviour, the thresholds of the event detector modules of the IR and light sensor systems were set to zero. Thus, the motor activity results from the winner-take-all retrieval mechanism applied on the inputs of these two sensors. Figure 12 shows the variation of activity of the units corresponding to the left and right motors, the compass, light, infra-red and radio sensors, during 1000 processing cycles (each sensory system is in fact represented by more than one unit; what we represent in figure 12 is the maximal activation of all units corresponding to this system). We observe that activation of the infra-red detector unit at times 210, 350, 430 and 780 produce an immediate deactivation of the right motor. The robot turns to the left when it faces an obstacle, as it was predefined by setting up the values of the connection parameters. As a result of the robot's rotation, a new value for the compass is measured at time 450. Light detection (which correspond to detect the second robot) at time 380 and 500, produce a deactivation of left and right motors alternatively. As a result, the robot aligns behind the other robot.

Teacher and learner's behaviours are controlled by the same set-up, that is they have the same predefined basic behaviours. However, no learning mechanism is used for the teacher and its knowledge of the vocabulary is defined by setting the connections between the radio sensor (words are radio signals) and other sensors, which correspond to the sensor measurement which the words describe. The learner robot uses both the input and the output of the radio module to receive the teacher's signals and to emit its answer (which corresponds to its retrieval of the learned radio signal given its current sensor measurements), which is recorded by the experimenter in order to evaluate the progress of the robot's learning during the experiment. The teacher robot uses only the output of its radio emitter to send the signals to the learner robot. That is, the learner robot's answer is not used by the teacher robot to check the efficiency of the teaching. For this reason, the teaching in these experiments is completely unsupervised.

Similarly to what is done for the motor activation, the teacher robot's ability to emit radio signals (speaking/teaching) results from retrieving the output of the radio sensor system, given the robot's current sensor-motor state. The teacher 'speaks' only when it sees the learner. The inhibition of the activation of the radio output is achieved by giving a very high value to the confidence factor of the connections between the light detector units (learner recognition) and the radio output units. The input to this connection is 1 as long as the learner is not in view, otherwise zero. Therefore, when the learner is not in view, the activation of the light units wins the competition of activation (because of its very high confidence factor value), inhibiting activation from other sensor units. As a result, all the radio units are activated, which by definition would produce no output.

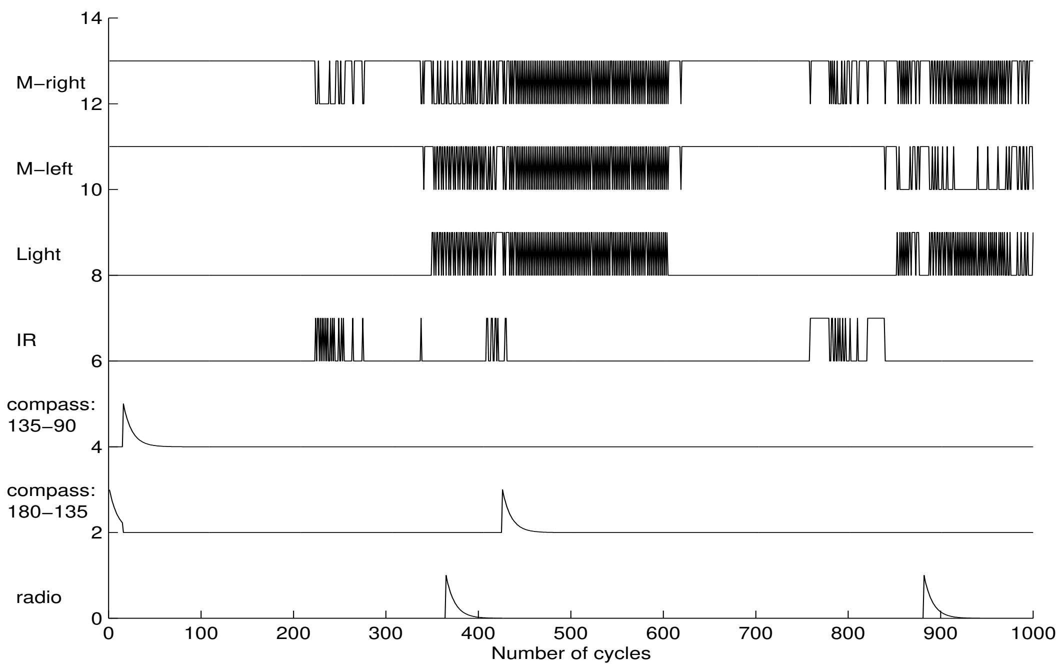

Figure 12: Variation of activity of the units corresponding to the left and right motor, the compass, light and inclination sensors, during 1000 processing cycles.

Learning occurs when one unit in one sensor system is newly activated, that is when the output of the event detector associated to this sensor system is activated. The time parameters and confidence factors of the network's connections linking previously or simultaneously activated units to the newly activated unit are then updated following equations 4 and 5, given in section 2.5.4. The time parameters give a measure of the mean time delay between consecutive activation of the two units, while the confidence factors record the frequency of co-activation of the two units.

Table 3: Comparison between results of simulations and physical experiments.

	Experiment 1		**Experiment 2**	
	Simulations Mean & Std	**Physic. exp.** Mean & Std	**Simulations** Mean & Std	**Physic. exp.** Mean & Std
1	0.97 ± 0.02	0.71 ± 0.18	0.62 ± 0.1	0.68 ± 0.03
2	0.99 ± 0.04	0.96 ± 0.10	1 ± 0	1 ± 0
3	30 ± 19	36 ± 8	583 ± 20	176 ± 15

The DRAMA network keeps a memory of each unit activation for a fixed time delay, which is determined by the rate of activation decrease along the units self-connection. In the example of figure 12, a unit's activation is conserved for about 100 cycles by the effect of the recurrent connections, similarly to what was done in the simulation of the first experiment. The level of activity decreases by a ratio of 0.9 at each cycle. In figure 12, we can see the decrease of activation of the radio and compass units. Motor and infra-red units do not decrease because they are constantly maximally activated by the new input (since there is no event detection for these sensor systems). When the radio unit is activated, at cycles 380 and 890, it is associated with the following activation of the compass unit, just before its deactivation. These bidirectional associations between simultaneous and sequential sensory activations, lead to the learning of the vocabulary, where radio bit-strings are associated with particular sensor activities. Simultaneously, other associations occur between other sensory stimuli showing physical regularities in the environment (e.g. associating the objects' features with their location in terms of compass and energy measurements).

5 Results

In this section, we report on the results of the two experiments simultaneously, comparing the results of the simulation studies with those of the physical experiments. In a first part, we evaluate the speed and stability of the learning by following the variation of the learning parameter values during the experiments. In the second part, we show the success of sequence learning in the second experiment, by running off-line rehearsal of the sequence. In the third part, we study the influence of the winner-take-all threshold T (on the confidence factors) on the determination of the learning success in the first experiment.

5.1 Speed and stability of the learning

In a Hebbian type of associative memory such as ours, the success of the learning can be measured at each time step as the ratio between the connection parameters values. At each presentation of a new example (new teaching), the connection parameters (confidence factor and time parameter) are updated. The study of the parameter values' variation during the experiment inform us about the variation of the percentage of noise and consecutively about the stability of the learning. Noise, in our case an incorrect matching between sensor information, can be due either to noisy sensor measurements (hardware imperfection) or to learner and teacher agents making different observations (i.e. observing different events). Because the following of the two agents is imperfect (bad alignment, 'zigzagging'), their respective measures of compass direction can, for instance, differ. It might also happen that an object is in the teacher's field of view cone while not in that of the learner.

Sets of 30 and 10 simulated runs were carried out for the first and second experiment respectively. In the first experiment, a run simulated 200,000 processing cycles of the robots (about 55 hours of real experiment). For each run, the robots started in a different position, randomly generated, sometimes close together and already aligned one behind the other one, sometimes far apart. To prevent any bias in the experiments we allowed homogeneous covering of the space by adding randomness in the robots' travelling, so that each word was taught about the same number of times and so that each object was approached from different directions. In the second experiment, each run started with the robots aligned one behind the other and placed at the bottom left entrance of the series of corridors. A run consisted of the robots travelling ten times along the corridors, each time in the same direction. Two sets of physical experiments were carried out to reproduce the simulated experiments 1 and 2 (with 5 runs for each set).

Table 3 gives, for each set of experiments, the mean value and standard deviation of (1) the ratio between the confidence factor (cf) values associated with the correct word and the maximal value of cf attached to all words for a given sensor measurement, i.e.
$cf(correctcorrelations)/(cf(correctcorrelations) + max(cf(incorrectcorrelations))$,
(2) the number of words learned at the end of a run and (3) the number of teachings given in a run. We observe that in all experiments, the confidence factor ratio is greater than 0.5, which means that the correct correlations have been made more often on average than the incorrect ones, hence that learning is successful. In addition, learning is stable as the standard deviation for the ratio is small, keeping its value above the threshold of 0.5. Learning of the complete vocabulary is always successful in the second experiment, but not in the first one. This is due to the fact that in the first experiment, the robots could sometimes miss an object during a run, hence making no teachings about it. This is especially the case in the simulation as the covering of the whole space depends on the randomness of the robots' behaviour; in the physical experiments, however, the human observer could 'force' the robots to cover the whole space by attracting them using an external light. Note that the fact that the vocabulary was not completely learned was not due to unsuccessful learning but to the absence of teaching for these particular objects (no update of the connection parameters for the corresponding radio unit). In the second experiment, this cannot happen since the robots' path is constrained by the corridors which forces them to perceive at each time the expected stimuli.

Figure 13 shows the variation of the time parameters values for each correct connection between radio units (signal) and corresponding sensor measurement (object) along a run. Data represent mean value over all runs. Left and right figures show the result of simulation studies and physical experiments 1 and 2. As expected, we observe that the values for the time parameter in the first experiment (figure 13 top) do not stabilise and that significant fluctuations (up to 10 times bigger than in experiment 2, outside values are out of the graphic) are measured all along the experiment. The vocabulary is learned but there is no regularity in the time of occurrence of the consecutive sensor perception. By opposition, in the second experiment(figure 13 bottom), the values stabilise quickly (after about half a run). The experimental results (figure 13 right) are similar to those of the simulations in showing a quick stabilization of the values in the experiment 2 and continuous fluctuations in the experiment 1.

5.2 Sequence rehearsal

Based on a quantitative and qualitative comparison between the connection parameter values at the end of each run, we were able to assess, in the previous section, the success of the learning of the second experiment. We observed that the robot had made the expected correlations between its different sensor measurements: it had made the correct correlations between the three radio signals and the different compass and light measurements, which was demonstrated by the ratio of confidence factors remaining above the threshold of 0.5, see table 3; it had correctly recorded

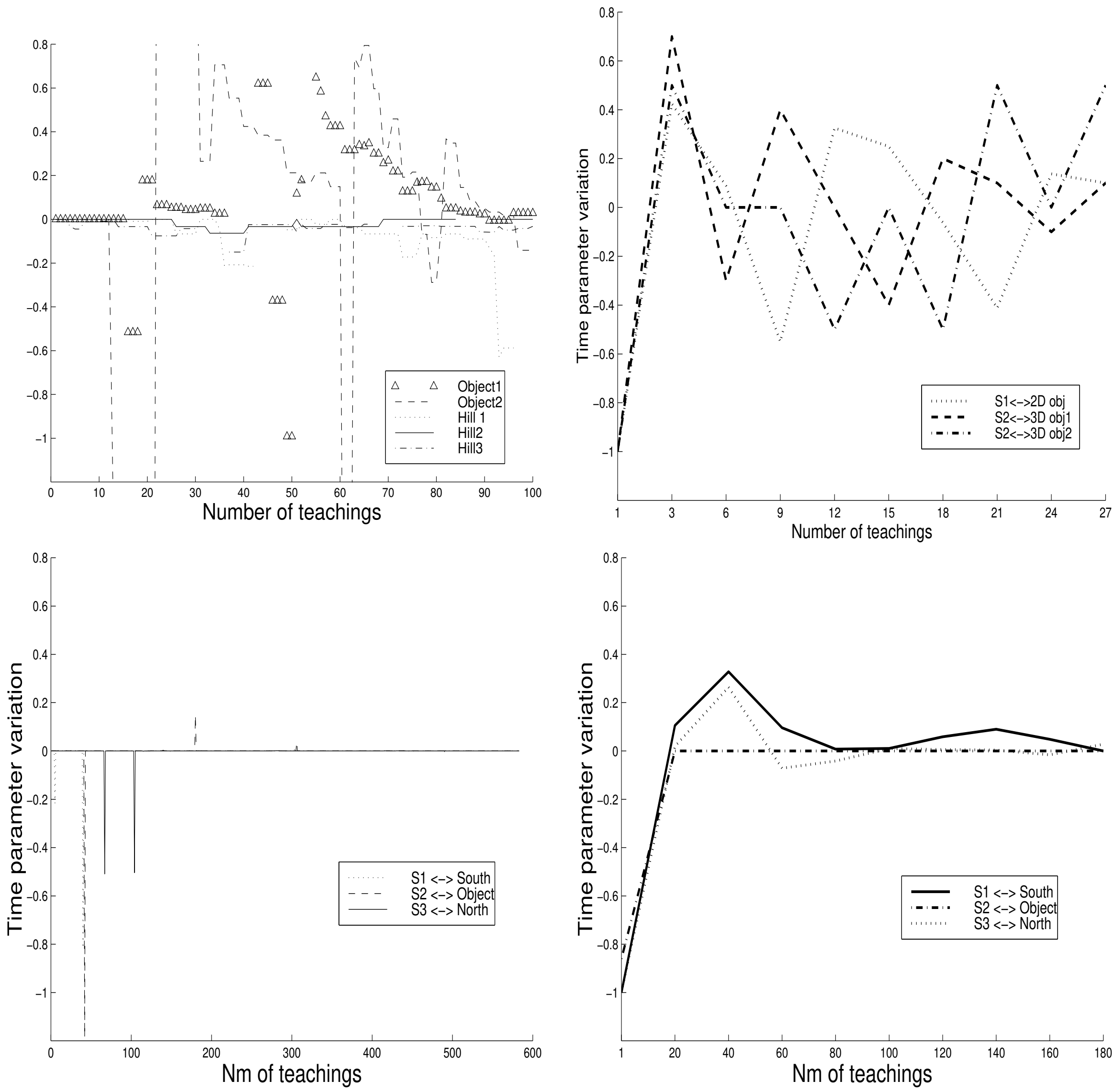

Figure 13: Variation of the time parameter values for each correct connection signal-object.

a temporal correlation between specific sensor measurements of compass and light sensors, that was demonstrated by the stabilization of the time parameter values in figure 13. However, it now remains to demonstrate that the robot has learned the correct sequence of stimuli, i.e., that it has learned the correct timing between each sensor measurement occurrence.

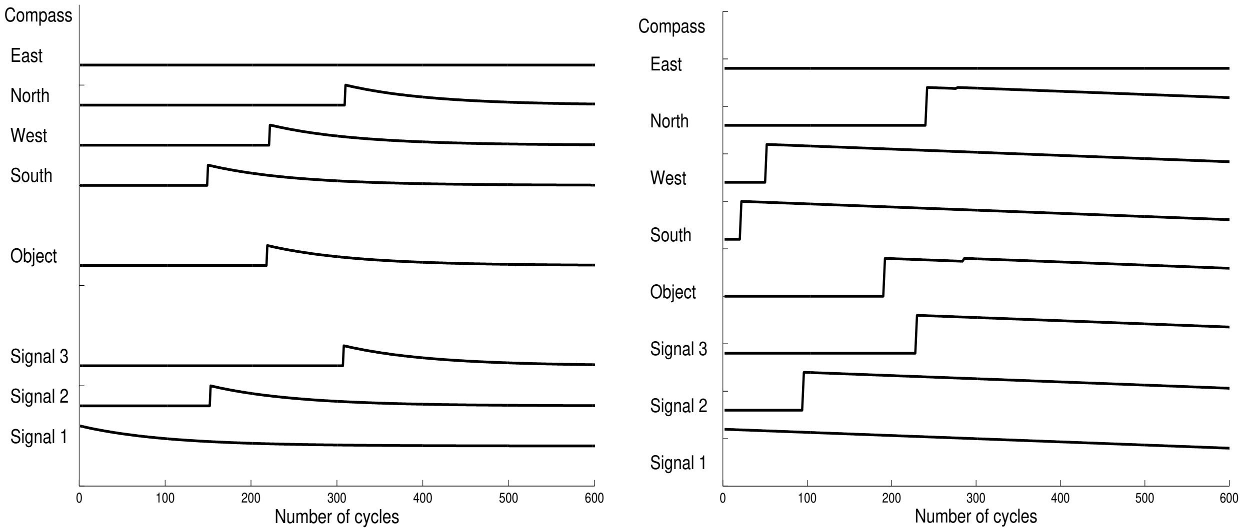

Figure 14: Retrieval of the sequence of sensor measurement with radio signal 'South' (signal 2) as start activation, for simulated (left) and physical (right) experiments.

We demonstrate this, by running off-line rehearsal of the sequence of measurements, taking the radio signal 'South' as the starting activation (the signal 'South' is supposed to be the first sensor measurement the robot perceives when entering the first corridor). Rehearsal consists of retrieving all the network units' outputs for 600 cycles (this corresponds to the time needed by the robot to make one circle across the three corridors, about 10 minutes), starting with all units' input and output set to zero, apart from the input to the third radio sensor unit which is set to 1 (this unit corresponds to the signal 'South'). Rehearsal of the network was done, using the values of connection parameters, obtained at the end of the run with the highest ratio of word learning success.

Results show that the starting activation of the radio unit 1, ('signal 1'), for 'South' is followed by a sequence of unit output activations in the radio, compass and light sensors. Figure 14 shows the sequence of sensor measurements, which results from the rehearsal process applied to the network, using parameters obtained in the simulation (figure left) and in the physical experiment (figure right). We observe that in both cases the radio signal for 'South' retrieves successively the compass value for South (1st corridor), the compass value for West (2nd corridor), the radio signal for 'Object' (signal 2), that is the patch (aluminium plate in the physical experiment) lying on the middle of the second corridor, the radio signal for 'North' (signal 3) and the corresponding value of compass. This demonstrates that the expected stimuli sequences (see the introduction of this chapter) have been correctly learned in both simulations and physical experiments. However, the exact time delay in terms of processing cycles between retrieval of each stimulus differs between simulation and physical experiments. Although the simulated set-up is quite similar to the experimental one (same proportion between corridor length and robots' size, same position and dimension for the patch), the two environments differ. The robots' wandering along the corridors in the real environment is very chaotic and varies significantly from one run to the next. In addition, the radio reception is not perfect in reality which results in significant variation in the time delay between reception of the signal and measure of the corresponding sensor stimuli[7]. These two facts account for the observed differences in time delay in the sequence rehearsal between simulated and physical experiments.

5.3 Retrieval with threshold T

When discussing the model's capacity in section 3.1, we pointed out the importance of the threshold parameter T for determining the success of the learning. The value of T determines the minimal ratio between the values of confidence factors of correctly and incorrectly updated connections. Learning is unsuccessful when the percentage of noisy examples, i.e. incorrect update of connections, exceeds $1/T$ times the total number of examples, because in this case the G function of equation 2 would output 0 when applied to the correctly updated connections, thus preventing the correct association to be retrieved. The value of T needs then to be carefully chosen, taking into account an estimation of the percentage of experimental noise. In table 15 left, we show the confidence factor values at the end of the run (mean values over all runs) for the simulation studies of the first experiment. We observe that some associations between radio units (each unit defines a different word) and boxes and hills' features are spurious. That is, for instance, the radio unit standing for hill3 is correlated with the combination of features of both box2 (colour2 + shape2) and hill3 (colour2 + inclination2), but with different values of confidence factor. Lowering too much the threshold on confidence factor could then have the effect of allowing retrieval of the two combinations of features rather than just the one for hill3 when presenting the radio signal for hill3. In table 15 right, we show the effect of varying the value of T on our determination of the correlation success for these results.

	Features					
	Color1	Color2	shape1	shape2	Incl1	Incl2
Box 1	1.20	0.00	1.20	0.00	0.02	0.00
Box 2	0.00	0.90	0.00	0.90	0.00	0.12
Hill 1	0.25	0.00	0.08	0.00	0.25	0.00
Hill 2	0.00	0.15	0.00	0.00	0.15	0.00
Hill 3	0.00	0.55	0.00	0.05	0.00	0.55

$1/T = 0.97$		$1/T = 0.5$		$1/T = 0.1$		$1/T = 0.01$	
→R	R→	→R	R→	→R	R→	→R	R→
Y	Y	Y	Y	Y	Y	N	Y
Y	N	Y	Y	N	Y	N	N
Y	N	Y	Y	N	Y	N	N
Y	Y	Y	Y	Y	Y	Y	Y
Y	N	Y	Y	N	Y	N	N

Figure 15: **Left:** Confidence factor values for connections between radio units (1 unit = 1 'word': 5 words for the three hills and 2 boxes) and colours, shape and inclination sensors units (hills and boxes features). **Right:** Success of signal-object correlation given 4 different values of threshold T. Y/N stand for Yes/No (correct/incorrect correlation); R: radio sensor, O1: Box1, O2: Box2, H1: Hill 1, H2: Hill2, H3: Hill 3. '→R' and 'R→' columns show results of learning for each direction of association (from radio sensor to object's description and vice-versa).

As mentioned previously, we estimate a proportion of at least 20% of experimental noise due to hardware imperfection, onto which we should add the noise due to the imprecise following of the two agents resulting in incorrect matchings of sensor perceptions. Taking a threshold of $1/T = 0.5$ allows correct retrieval of the data in the face of a maximum of 50% of noisy data. As expected, with this value for the threshold, only the correct correlations (from each object to its corresponding word and vice-versa) are correctly retrieved. However, when lowering the threshold, spurious correlation can also be retrieved. E.g., with a threshold $1/T = 0.1$ the radio signal for hill3 retrieves the sensor features for both box2 and hill3 and with a threshold $1/T = 0.01$ the

[7]The teacher sends the same signal about 10 times for a given stimulus in order to compensate for the loss in the reception; the variation in the time delay arises from the learner catching only the first or the latest signals.

combination of features for hill3 retrieves both signals for hill3 and box2. On the other hand, a too restrictive threshold, that is too high, means that some correlations are no longer retrieved, e.g. with $T = 0.97$, the radio signals for box2, hill2 and hill3 no longer retrieve the full set of features of the corresponding elements.

6 Discussion

An important part of this paper (sections 2 and 3) was used to describe the DRAMA (Dynamical Recurrent Associative Memory Architecture) architecture, which we developed to allow learning of spatio-temporal regularities by an autonomous robot. The model consists of a fully recurrent neural network without hidden units, which uses Hebbian update rules. Similarly to time delay networks, it uses two weight parameters for each connection, to record separately the time delay and the frequency of two input patterns co-occurrence. The DRAMA network differs from other structurally or functionally similar ANN models in two main aspects: 1) By opposition to other recurrent neural networks, it is based on an unsupervised learning algorithm, which uses Hebbian rules. 2) In contrast to other associative memory models, such as Hebbian networks, the connections of the network are associated with two parameters (instead of one) in order to keep a separate record of the spatial and temporal structure of the input patterns. In particular, the temporal parameter allows recording the real time of occurrence of the pattern.

In section 3, we analysed theoretically and through numerical simulations the properties of the model. The model was shown to cope (that is, the capacity remains maximal) with up to 30% of noise, where the noise corresponded to a 30% likelihood of spurious unit activation when presenting the training patterns and a 30% variation of time delay between consecutive activation of the input and output units of the pattern. The model was shown to be able to learn time series of inputs, while the series can overlap on one or more inputs (that is, it can learn several time series, where the series can have several similar inputs but with a different ordering of occurrence).

Training and retrieval algorithms are of one-time-step, which makes the model computationally fast and inexpensive to run and, therefore, allows its implementation for on-line learning of a computationally limited robot. The advantages of the model in terms of quick and easy computation will be further discussed in section 6.1. In the following, we discuss the general properties of the model by comparison with associative memory models and recurrent neural networks, of which it has several similar properties.

The model as an associative memory

The DRAMA architecture has several characteristics in common with associative memory models, such as Hebbian networks, as it uses a similar training algorithm (Hebbian rules) and a similar retrieval algorithm (winner-take-all). Similarly to Hebbian networks which use a binary encoding for the patterns, the capacity of the DRAMA model decreases when trained with patterns whose encoding overlaps, that is, patterns which have common units active. In section 2, we compared the DRAMA model to the Willshaw network, which is a special case of the Hebbian neural network, and from which we inspired ourselves. In addition, Simulation studies showed that the decrease of the DRAMA network capacity in the face of noisy data is graceful and remains above 90% of the maximal capacity with up to 90% of noise, which is an improvement compared to the Willshaw network, which decreases to 10% of the maximal capacity in the face of at minimum 40% of noise [16]. The DRAMA architecture differs from other Hebbian networks mainly in its recurrent structure (self-connections on the units and bidirectional asymmetric connections between the units), while other models use unidirectional or symmetric connections. The recurrent connections introduce a short-term memory of the units' activity, which allows association of temporally delayed unit activations, while the time delay between the two units' activation is unspecified, but remains within the margin of the short-term memory duration. By transitivity of the associations, time series of unit activations can be learned. Retrieval of the associations is such that each unit activates its correlated unit only when the correct time delay has passed. This property of DRAMA to introduce explicitly the time into one of the connection parameters (the time parameter) is what distinguishes it most significantly from other models of associative memory. Associative memory models that can learn sequences of patterns do exist ([18], [24], [36], [37]). However, in these models, the time delay between each pattern occurrence is fixed and is equal to one processing cycle (it has therefore no intrinsic relationship with the real time of pattern occurrence). That is, the patterns of the series are presented sequentially to the net, without delay, and are retrieved similarly.

The model as a recurrent neural network

The structure of the DRAMA network is a fully recurrent network, without hidden units. Similarly to other recurrent neural networks, it allows learning of time series of inputs. DRAMA differs from other RNN models by the fact that it uses an unsupervised training algorithm, based on Hebbian rules, while other RNNs use a supervised training algorithm, such as the backpropagation algorithm and other derivatives [9, 33]. The advantage of using Hebbian rules is that training of the network requires only one-time-step for each processing cycle, which allows to process the information (e.g. sensor information of the robot) in real time, while backpropagation needs several time steps of computation (usually of order 100 to 1000) between each information processing cycle. The drawback of using usupervised Hebbian learning algorithm is that it can not be used to train a network with hidden units, that is, with intermediary units between input and output units, whose values are unknown. The algorithms developed to train RNNs with hidden units are the Boltzmann machine learning procedure [20], Backpropagation [35] and other similar procedures, which all require several time steps of computation for each information processing cycle (which prevents on-line learning in computationally limited robotic system as ours). It would be interesting, however, to investigate how hidden units could improve the network performance and, in particular, its capacity. 'Hidden units makes it possible for the network to discover and exploit regularities of the task at hand such as symmetries or replicated structures' [33]. Therefore, using hidden units might improve the network's ability for discriminating between redundant or overlapping patterns and consequently increase the network's capacity, that is the number of patterns the network can store, by allowing more overlap between the patterns before the network fails to distinguish between them.

6.1 DRAMA performance in robotic experiments

The experiments reported in section 5 demonstrated a possible implementation of the model for controlling the behaviour and learning of autonomous robotic agents. The learning task involved in the experiments was relatively complex. Multiple associations had to be made between stimuli with variable time lags of occurrence. In particular, patterns composed of the same set of features (colour, inclination, compass measurement) but in different combinations had to be distinguished. Learning was successful, although about 30% of the data was corrupted (because of hardware noise and imprecision of the teaching/following method). The experiments showed 1) that the DRAMA architecture enables real-time computation and can be used for on-line control and learning of autonomous mobile robots; 2) that it performs static and sequential associations leading to the robot's learning of spatial regularities across the sensor-actuator space and of time series of consecutive sensor measurements.

Simulation versus physical experiments

Simulation studies were carried out before physical experiments in order to demonstrate and test the stability and success of the learning in the proposed experimental set-up. The main advantage of simulations over physical experiments was that they were repeatable, faster (simulating a 1 hour experiment takes about 5 minutes) and did not suffer unexpected hardware breakdown. The disadvantages in terms of model faithfulness are, of course, well known (for a more complete discussion of this see [41]). In our experiments, physical and simulated worlds differ in many aspects. For instance in the simulation a poor account is given of the physics of the sensors and of the world perceived by the robot (a simple field of view is defined for the light dispersion whose intensity is invariant over time, the inclination of the hills is perfect in all points, etc.). In addition, simulated and real objects are not described by the same nor the same number of features in the first set of experiment. Because of this, the results of simulations and physical experiments can only be compared qualitatively. Thus, our claims on the results (see 1 and 2 above) are on qualitative characteristics of the DRAMA architecture, which are demonstrated by both simulations and physical experiments.

Evaluation of the learning parameters

The success of the learning depends on correctly choosing the values of the learning parameters, namely the duration of the short-term memory and the values of the threshold parameters, T and e, which appear in the neuronal activation function 2. In [2], we reported on experiments where we could relate cases of success and failure of the learning to particular choices of environmental constraints (objects' relative dispersion and featural descriptions), and values of the duration of short-term memory of events and could then determine bounds on these parameters inside which learning would be successful.

The threshold factors T and e, which appear in the units activation function (see equation 2) determine also the success of the learning, by fixing the tradeoff between considering two units co-activation as spurious or relevant. The threshold e discards correlations due to spurious unit activity, by evaluating the mean time delay of two units co-activation and considering as irrelevant the association of units, whose delay of co-activation varies too importantly during the training, relative to a maximal variation fixed by e. The threshold T determines the tradeoff between spurious and relevant associations, by comparing the frequency of co-activation of the correlated units (given by the *confidence factor* parameter of the connection linking the two units). The more often the units have been co-activated, the more likely it is that this co-activation is not hazardous. In section 5.3, we discussed the influence of the threshold parameter T on our determination of the success of the first experiment. In section 3.1, we presented an algorithm to determine on-line the values of these thresholds, i.e. to calculate these values at each time step. Further, in section 3.4, we presented an algorithm for on-line tuning of the short-term memory value and tested it in simulation for learning a sequence of nine patterns. The algorithm was shown to successfully converge, for up to a proportion of 20% of noise in the input. However, as the time needed for convergence was relatively long, the algorithm was found to be too slow to be used reliably in physical robotic experiments [4].

Note finally, that other factors also influence the success of an experiment, e.g. the agent's sensor capabilities (range and sensitivity) and behaviour control (non homogeneous travelling). Further implementation of the model in different robotic set-ups, in particular in robots with more degrees of freedom and finer sensor sensitivity would allow to determine the real influence of these hardware characteristics on the success of the learning. Note, however, that the fact that we implemented the architecture in three different robotic set-ups (FischerTechnik vehicles [6], LEGO vehicles (this experiment) and a doll robot [3]), using different sensors and applied in different environments, showed that the success of the learning is not dependent on a particular type of hardware. However, it might be improved by using finer sensor capabilities, which would give more information to distinguish between the objects of the teaching (lowering the overlap between teaching patterns and thus improving the network capacity), and better actuator capacities, which would make the robot following the other robot smoother and thus less prone to incorrect measurements.

Robot's grounding of perceptions and actions

One of our requirements for starting this work was that the system should learn quickly, i.e. that it would not require a long series of examples before performing adequately. In the first experiment, we showed that correct associations between radio signals and object features were learned in less than 30 teachings, which correspond to about 15 to 30 minutes of physical experiment. Similar experiments on grounding radio signals into robots' sensor capabilities were carried out previously by Yanco & Stein [44] and Steels & Vogt [38], who used respectively reinforcement learning and evolutionary techniques. Their experiments showed that a vocabulary of 5 and 3 words was learned after 900 and 60 training examples respectively. Our method then seems faster at learning a larger or similar vocabulary. In addition, it is more general than the above mentioned methods, as we were not restricted in the sensor stimuli the robots could talk about. In Yanco & Stein's work [44], the vocabulary consisted only of the robot's actions because the learning algorithm was based on an action-selection mechanisms. In Steels & Vogt [38], the vocabulary concerned only the robots' external perceptions as these were the only perceptions they could share. By contrast, the mutual following strategy we use in our work allows the two agents to share a common context of both external (face the same direction) and internal perceptions (perform the same movement, travel the same distance and on the same ground). In addition, because the learning mechanism we use is based simply on mutual associations between inputs from any sensor or actuator systems of the agent, the vocabulary can potentially concern any proprio and extero perceptions of the agent. We reported here on experiments where the agents talked about external perceptions of objects and internal perceptions of inclination and direction. In [5], we reported on experiments where the learner agent is taught a vocabulary of eight words for describing its movements in terms of motor states *stop, move, turn right, turn left* and its direction *North, South, West, East* relative to its compass.

Robot's learning of sequences of perceptions

In the second experiment, the robot learned the timing and ordering of a sequence of compass, radio and light measurements, resulting from its regular travelling in a series of three corridors and its regular reception of the teacher robot's signals. This experiment demonstrated the DRAMA architecture capacity at learning spatio-temporal regularities and time series of sequences of a robot's perceptions.

There has been a number of studies in robotics, in which the robot learned spatio-temporal regularities in its sensor inputs. These works used, for instance, a two-layer topographical map to store separately the spatial and temporal regularities (in each layer) of the robot's visual information (CCD camera and infra-red sensors) [14, 30], or a recurrent neural network to predict sequences of a robot's perception-action, when travelling in a corridor [39]. Our experiment, using the DRAMA architecture, has two advantages compared to these works: 1) learning and retrieval can be performed on-line[8], by contrast to [30, 39] works where it was done off-line, 2) learning concerns several sensor and actuator modalities, as opposed to [14, 30, 39], which consider only association from the robot's sensor perception to the robot's actions.

Why not using another ANN architecture for the experiments

Since part of the learning task in our experiments was to learn the topography of the environment relative to landmarks, one might wonder why we did not use a self-organising map such as, e.g., Kohonen nets [23] or other models developed previously for robotic tasks (e.g. [30],[46]). The reason is simply that the associations we want to make in the sensor-actuator state are point-like, that is they have no topographical relationships in the sensor-actuator vector space (this is demonstrated in figure 15 left where we show the dispersion of the associations at the end of the learning).

One may question why we did not use one of the current recurrent neural network models (e.g. Elman net [12], Jordan net [28], dynamical recurrent net [15] and others [33, 9]). There are several reasons for that. The first one is that, because of hardware limitations, we were restricted to defining a system that would use only integers (no floating points) and that should be computationally fast (because of the limited on-board processing power). Recurrent neural networks using the backpropagation algorithm, as developed first by [35], and other extension [33, 9] had to be eliminated because of their long time computation (multiple training steps) and their complex computation (calculating derivatives). Associative Hebbian networks were very attractive because of their simplicity. One may argue in favour of buying a more powerful hardware system. Apart from the financial aspect, there is a motivation to try to do the best with what is at our disposal. We are not using expensive robots nor expensive sensors (no camera, laser, etc.), but we developed a system that is capable of more complex cognition than the simple behaviours of obstacle avoidance, wall following, etc.

Note, finally, that the DRAMA architecture's poor space efficiency (capacity of order two) was not a disadvantage in the particular robotic experiments we used it for. More important for us was the time efficiency and the capacity at learning complex time series. However, it might be a disadvantage for application using sensors with high sensitivity, such as a camera, which would require an important number of units to represent the sensor information (one unit per pixel or set of pixels). In this case, it might be relevant to pre-process the data using another ANN architecture, such as a topographical map or a feed forward NN, for a preliminary classification of the data[9]. Then, the reduced amount of information could be used by the DRAMA network for higher level classification. Such an approach has been followed by Tani [39], who uses a combination of a Hopfield associative memory network and a RNN. The robot's camera information is processed by the Hopfield net which determines categories of visual inputs. The recurrent neural network is trained on the output of the Hopfield net and on the simultaneous motor state of the robot. The net learns sequences of visual perception and action of the robot, while the robot travels in a circling corridor. Because J. Tani used backpropagation algorithm to train the RNN, he could not process the information on-line. It would be interesting to carry out the same work using the DRAMA architecture, instead of the RNN with backpropagation, and to determine whether it would allow to carry out successfully and on-line the same computation.

[8] In the experiments reported here, only learning is done on-line; in [3], we reported on on-line learning and retrieval of different sequences of actions and perceptions of the robot.

[9] Note that topographical maps are particularly relevant for sensors with high resolution as the classification often relies on finding topological invariance in the input.

7 Conclusion

We described a novel connectionist architecture, DRAMA, for dynamic control and learning of autonomous robots. DRAMA stands for dynamical recurrent associative memory architecture. It is a time-delay recurrent neural network, using Hebbian update rules. The first part of this paper presented a mathematical description of DRAMA and analysed theoretically and through numerical simulations the architecture's performance. The model was shown to allow learning of spatio-temporal regularities and time series in sequences of inputs, in the face of an important amount of noise. Training and rehearsal of the DRAMA architecture is computationally fast and inexpensive, which makes the model particularly suitable for controlling 'computationally-challenged' robots.

The second part of the paper reported on the implementation of DRAMA in simulated and physical robotic experiments, for on-line learning and control of an autonomous robot. In the first experiment, the robot extracted spatial regularities in its perceptions, which resulted in the recognition and labelling of objects in the environment. The objects' labels where taught by a second autonomous robot. In the second experiment, the robot learned time series of its perceptions, while travelling in a series of corridors.

Results of simulated and physical experiments were consistent, in showing successful learning, and therefore demonstrating the robustness of the learning architecture in the face of a significant amount of experimental noise. Grounding of the objects' names was shown to be faster than other similar robotic experiments, which used less general, that is, more task related, learning mechanisms than we did, using the DRAMA architecture. In addition, we used a single architecture, DRAMA, for enabling learning and directing of the robot's behaviour, while the learning mechanisms were not restricted to a particular direction of association between sensor-actuator states as is the case in most robotics learning experiments. However, the complexity of the experiments was limited by the poor sensor capabilities and computational power of our robots. In particular it restricted the number of things the robot could learn, as it could perceive few features and could not record an important amount of data. It would now be interesting to implement the model in more powerful robots using more complex sensor modalities.

8 Acknowledgements

Aude Billard is supported by a grant from the Swiss National Science Foundation. Facilities were provided by the University of Edinburgh. A big thanks to the technicians of the AI dept. for their precious support. Many thanks to Bruce Graham, Maja Mataric and anonymous reviewers for their useful comments on earlier versions of this paper.

References

[1] Asada M., E. Uchibe and K. Hosoda (1997), 'Cooperative Behavior Acquisition for Mobile Robots in Dynamically Changing Real Worlds via Vision-Based Reinforcement Learning and Development', Proc. of First International Workshop on Cooperative Distributed Vision, pp.127-142.

[2] Billard A. & Dautenhahn K. (1998), 'Grounding communication in autonomous robots: an experimental study', *Robotics and Autonomous Systems*, Special Issue on 'Scientific methods in mobile robotics', M. Recce and U. Nehmzow (eds.), No. 24, Vols. 1-2, 1998.

[3] A. Billard, K. Dautenhahn, G. Hayes (1998), 'Experiments on human-robot communication with Robota, an imitative learning and communicative doll robot', Proceedings of Socially Situated Intelligence Workshop, Zurich CH, as part of the Fifth Conference of the Society of Adaptive Behaviour, B. Edmond & K. Dautenhahn (eds), Centre for Policy Modelling technical report series: No. CPM-98-38.

[4] A. Billard, (1998), 'DRAMA, a connectionist model for robot learning: Experiments on grounding communication through imitation in autonomous robots', PhD thesis, Oct. 1998. Technical Report, Dept. of AI, University of Edinburgh.

[5] Billard A. & Hayes G. (1997), 'Learning to communicate through imitation in autonomous robots', Proceedings of ICANN97, 7th International Conference on Artificial Neural Networks, Wulfram Gerstner et al. (eds.), Springer-Verlag pp 763-768.

[6] Billard A. & Dautenhahn K. (1997), 'Grounding communication in situated, social robots', Proceedings of TIMR 97, Towards Intelligent Mobile Robots Conference, Manchester 1997. Tech. Rep. Series, Dept. of Computer Science, Manchester University, ISSN 1361 - 6161. Report number UMCS-97-9-1.

[7] A. Billard (1996), 'Allo Kazam, do you follow me?' Learning to speak through imitation for social robots. MSc thesis, Oct. 1996. Technical Report 43, Dept. of AI, University of Edinburgh.

[8] Buckingham, J T & Willshaw, D J (1992). 'Performance characteristics of the associative net'. *Network*, **3**, pp 407-414.

[9] Y. Chauvin & D. E. Rumelhart (1995), 'Back-Propagation: Theory, Architecture and Application', Lawrence Erlbaum Associates publisher.

[10] S. P. Day & M. R. Davenport, 'Continuous-time temporal back-propagation with adaptive time delay', IEEE transaction on neural networks, vol. 4, pp. 348-354.

[11] Dautenhahn K., (1995), 'Getting to know each other - artificial social intelligence for autonomous robots', *Robotics and Autonomous Systems* 16, pp.333-356.

[12] Elman J.L. (1990), 'Finding structure in time'. *Cognitive science*, 14, pp 179-211.

[13] Floreano, D. & Mondada, F. (1996), 'Evolution of Homing Navigation in a Real Mobile Robot', IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics, 26(3), pp.396-407.

[14] P. Gaussier, S. Moga, J.P. Banquet, M. Quoy (1998), 'From perception-action loop to imitation processes: a bottom-up approach of learning by imitation', Applied Artificial Intelligence, vol.7:1.

[15] Giles C.G, Kuhn G.M., Williams R.J. (1994), 'Dynamic recurrent neural networks: Theory and applications'. IEEE Transactions on Neural Networks, 5(2). Special Issue.

[16] Graham, B P & Willshaw, D J (1996), 'Capacity and information efficiency of a brain-like associative net'. *Advances in Neural Information Processing Systems*, **7**, pp.513-520.

[17] Graham, P B & Willshaw, D J (1995), 'Improving recall in a sparse autoassociative memory'. *Biol Cyber*, **72**, pp.337-346.

[18] Hattori M. & Hagiwara M. (1996), 'Episodic associative memories', Neurocomputing, Vol.12, No.1, pp.1-18.

[19] Hebb D.O. (1988), 'The organisation of behaviour'. Wiley, New York. Partially reprinted in J.A. Anderson and E. Rosenfeld, editors 'Neurocomputing: foundation of research' MIT Press, Cambridge.

[20] Hinton G.E. & Sejnowski T.J. (1986), 'Learning and relearning in Boltzmann machines', In D.E. Rumelhart and McClelland, editors, *Parallel Distributed Processing*, vol.1, chapt.7, pp.282-317. MIT Press, Cambridge.

[21] Hopfield J.J & Tank D.W. (1986), 'Computing with neural circuits: A model', Science 233, pp.625-633.

[22] Klingspor V., Demiris J., and Kaiser M. (1997), 'Human-Robot-Communication and Machine Learning', *Applied Artificial Intelligence Journal* 11, pp.719-746.

[23] Kohonen T. (1989), 'Self-organisation and associative memory', Springer-Verlag, Berlin, 3rd Edition.

[24] Kolen J. F. & Pollack J. B.(1991), 'Multiassociative Memory', Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society, August, pp.7-10.

[25] Kuipers B. (1987), 'A qualitative approach to robot exploration and map learning', AAAI workshop spatial reasoning and multi-sensor fusion, Chicago, pp. 774-779.

[26] Kurz A., (1996), 'Constructing maps for mobile robot navigation based on ultrasonic range data', IEEE Trans. systems, Man and Cybernetics B, Vol 26 No2.

[27] D. T. Lin, P.A. Ligomenides, J. E. Dayhoff (1993), 'Learning spatio-temporal topology using an adaptive time-delay neural network', Proceedings of World congress on neural networks, Portland, OR, vol. 1, pp. 291-294.

[28] Jordan M.I. (1986), 'Attractor dynamics and parallelism in a connectionist sequential machine'. In proceedings of the eight annual conference of the cognitive science society, pp.531-546, Amherst. Lawrence Erlbaum, Hillsdale.

[29] Mataric M.J. (1997), 'Reinforcement Learning in the Multi-Robot Domain", Autonomous Robots, Vol. 4, No. 1, January.

[30] C. Owen & U. Nehmzow (1996), 'Route Learning in Mobile Robots through Self-Organisation', Proc. Eurobot 96, 1st Euromicro Workshop on Advanced Mobile Robotics, Kaiserslautern 1996. IEEE Computer Society.

[31] Nolfi S. (1997), 'Using emergent modularity to develop control system for mobile robots', *Adaptive Behaviour*, vol. 5:3-4, pp. 343-364.

[32] Nordin P. & Banzhaf W., (1996), 'An online method to evolve behavior and to control a miniature robot in real time with genetic programming', Adaptive-Behaviour, Vol.5, no.2; Fall 1996; pp.107-40.

[33] Pearlmutter B. A. (1995), 'Gradient Calculations for Dynamic Recurrent Neural Networks: A survey', IEEE Transactions on Neural Networks 6(5), pp.1212-1228.

[34] Pfeifer, R. & Scheier, C. (1997), 'Sensory-motor coordination: the metaphor and beyond'. In R. Pfeifer and R. Brooks (Eds), *Robotics and Autonomous Systems*, special issue on practice and future of autonomous agents." 20, Nos. 2-4.

[35] Pineda F.J. (1987), 'Generalisation of backpropagation to recurrent neural networks', Phys. Rev. Lett., 18, pp.2229-2232.

[36] Rinkus G. (1995), 'A Combinatorial Neural Network Exhibiting Episodic and Semantic Memory Properties for Spatio-Temporal Patterns', Ph.D. Thesis, Dept. of Cognitive and Neural Systems, Boston University, Boston, MA 02215.

[37] Schwenker F., Sommer F. T., Palm G. (1996), 'Iterative retrieval of sparsely coded associative memory patterns', Neural Networks 9, pp.445-455.

[38] L. Steels & P. Vogt, (1997) 'Grounding adaptive language games in robotic agents', Proceedings of the Fourth European Conference on Artificial Life, pp.473-484, Brighton, U.K., MIT Press/Bradford Books.

[39] J. Tani, J. Yamamoto, H. Nishi (1997), 'Dynamical Interactions Between Visual Attention, Learning and Behavior: An Experiment with a Vision-Based Mobile Robot', Proceedings of The Fourth European Conf. on Artificial Life, pp. 309-317, MIT Press.

[40] Thrun S. (1996), 'Explanation-Based Neural Network Learning - A Lifelong Learning Approach'. Kluwer Academic Publishers, Boston, MA.

[41] Torrance M. C. (1992), 'The Case for a Realistic Mobile Robot Simulator," in Working Notes of the AAAI Fall Symposium on Applications of Artificial Intelligence to Real-World Autonomous Mobile Robots, Cambridge, MA, October.

[42] Willshaw D., Buneman O., Longuet-Higgins H.", (1969), 'Non-holographic associative memory', Nature, vol. 222, pp. 960-962.

[43] Wyatt J., Hoar J, Hayes G. (1998), 'Design, analysis and comparison of robot learners', *Robotics and Autonomous Systems*, Special Issue on 'Scientific methods in mobile robotics', M. Recce and U. Nehmzow (eds.), No. 24, Vols. 1-2, 1998.

[44] Yanco H. & Stein L. A., (1993), 'An adaptive communication protocol for cooperating mobile robots', in From Animals to Animats 2: Proceedings of the Second International Conference on the Simulation of Adaptive Behaviour, edited by J.-A. Meyer, H.L. Roitblat and S.W. Wilson. The MIT Press/Bradford Books, 1993, pp. 478-485.

[45] Zimmer U.R, (1996), 'Robust World-Modelling and Navigation in a Real World', NeuroComputing, Vol. 13, Nos. 2-4, pp. 247-260.

[46] Zrehen S. (1995), 'Elements of Brain Design for Autonomous Agents', PhD thesis, Swiss Federal Institute of Technology, Lausanne, Computer Science Department, June.