Towards Pro-active Embodied Agents: On the Importance of Neural Mechanisms Suitable to Process Time Information

G. de Croon*, S. Nolfi†, E.O. Postma*

* IKAT, Universiteit Maastricht
P.O. Box 616 - 6200 MD - Maastricht - The Netherlands
Voice: 0031-433883477 Fax: 0031-433884897
e-mail: g.decroon@cs.unimaas.nl, postma@cs.unimaas.nl
http://www.cs.unimaas.nl

† Institute of Cognitive Sciences and Technologies, CNR
Viale Marx, 15 - 00137 - Rome - Italy
Voice: 0039-06-86090231 Fax: 0039-06-824737
e-mail: s.nolfi@istc.cnr.it
http://gral.ip.rm.cnr.it/nolfi

1 Introduction

A new research paradigm, that has been called Embodied Cognitive Science [18], has recently challenged the traditional view according to which intelligence is an abstract process that can be studied without taking into consideration the physical aspects of natural systems. In this new paradigm, researchers tend to stress situatedness, i.e., the importance of studying systems that are situated in an external environment [3, 4], embodiment, i.e., the importance of study systems that have bodies, receive input from their sensors and produce motor actions as output [3, 4], and emergence, i.e. the importance of viewing behaviour and intelligence as the emergent result of fine-grained interactions between the control system of an agent including its constituents parts, the body structure, and the external environment. An important consequence of this paradigm is that the agent and the environment constitute a single system, i.e. the two aspects are so intimately connected that a description of each of them in isolation does not make much sense [9, 10, 1].

Research in Embodied Cognitive Science often involves simple agents called "reactive agents" [14]. These are agents in which sensors and motors are directly linked and that always react with the same motor action to the same sensory state. In reactive agents internal states (see next section) do not play a role in determining the motor behaviour. The fact that the vast majority of research in this area focuses on simple reactive agents, however, is not due to a rejection of the importance of internal states. Rather, it is due to the difficulty of developing pro-active embodied and situated agents, that is agents able to: (a) extract internal states by integrating sensory-motor information through time and, (b) later use these internal states to modulate their motor behaviour according to the current environmental circumstances. In this paper we will focus on how pro-active agents can be developed and, more specifically, on which are the neural mechanisms that might favour the development of pro-active agents.

Given the difficulty of developing embodied and situated agents through explicit design [13] our attempt to develop pro-active agents will be based on an evolutionary robotics method [16], that is on the attempt to develop these agents through a self-organisation process that allows the evolving robots to develop their skills in interaction with the environment and without human intervention. By comparing the results of five sets of evolutionary experiments in which simulated robots are provided with different types of recurrent neural networks, we will try to understand the relation between the robots' capabilities and the characteristics of their neural controllers. In addition, we will show how special mechanisms for processing information in time facilitate the exploitation of internal states.

The paper is organised as follows. In the next section we define the term internal state. In section 3 we describe our experimental test bed which consists of a self-localisation problem that cannot be solved through simple reactive strategies. In section 4, we review five different neural models described in the literature that are potentially suitable to develop pro-active agents. In section 5, we describe the results of the experiments and the comparison of the results obtained with the five different neural architectures. Finally, in section 6, we discuss the implication of the obtained results and, in particular, the neural mechanisms that seem to constitute a pre-requisite for the emergence of powerful pro-active agents.

2 Internal state

The concept of internal state plays a central role in our investigations. In this section, we define the concept with particular reference to neural networks controllers.

An *internal state* is a set of variables of the agent's controller that might be affected by the previous sensory states perceived by the agent and/or the previous actions performed by the agent and that might co-determine, together with the current sensory states, current and future motor actions. By mediating between perception and actions, internal states allow agents to produce behaviour that is decoupled from the immediate circumstances while still remaining sensitive to them.

An internal state can consist of different entities. For example, in the case of a neural controller, they might consist of the activation states of some neurons and/or in the strength of the synaptic weights. It should be noted that there is not a one to one correspondence between the architecture of the controller and the type of strategy adopted by evolving individuals. For instance, although an individual provided with a recurrent neural network controller might potentially develop an ability to integrate information over time, it might also rely on a simple reactive strategy.

As we claimed in the previous section, agents that do not have an internal state are reactive agents, that is agents that always react with the same motor action to the same sensory state. Agents that have an internal state are pro-active instead, that is agents that are able to integrate sensory-motor information through time into internal states that co-determine the agents' motor behaviour.

In the context of neural network controllers, reactive agents are provided with feed-forward neural networks, that is neural networks in which sensory neurons are connected to motor neurons directly or through one or more layers of hidden neurons that do not have a recurrent connection (figure 1). In figure 1 s1 and s2 represent sensory neurons (also called input units). h1, h2, and h3 represent hidden neurons. o1 represents an output neuron. The bias neuron is a special neuron whose activation state is always 1.0. In these neural networks, neurons are updated in discrete time steps and the activation state of motor neurons and hidden neurons only depends on the activation state of the sensors and on the connection weights that are kept fixed during the lifetime of the agent.

Pro-active agents instead are provided with neural controllers that have an internal state. An internal state can be realised through different neural mechanisms. One possibility, for instance, is to provide a neural controller with recurrent connections. For example, in the neural network shown in figure 2, the gray hidden neuron receives connections not only from the sensory neurons but also from the hidden neurons including itself. This implies that the activation state of this hidden neuron is not only a function of the activation of the sensory neurons at time t, but also of the hidden neurons at time t-1. Given that the state of the hidden neuron at time t-1 is also affected by the state at time t-2 and so on, this implies that the activation state of this hidden neuron, that influences the state of the motors at time t, might be influenced by the sensory states previously experienced by the robot.

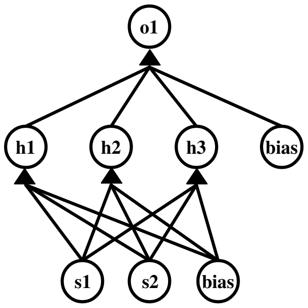

Figure 1: A feedforward neural network

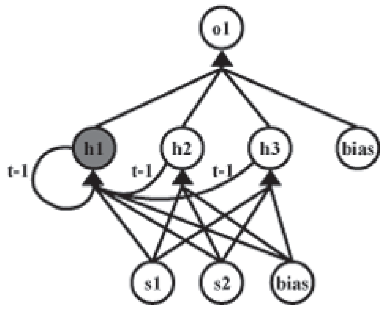

Figure 2: A recurrent neural network.

The recurrent neural connections, however, are only one of the possible neural mechanisms that might realise internal states. Other mechanisms include: (a) dynamical neurons, in which the activation state of a neuron is influenced by its previous activation state; (b) time delayed connections in which the propagation of activation through connections takes time so that the activation state of a neuron might influence the activation of other neuron after some time, (c) networks in which connection weights vary according to learning rules affected by the activation state of the neurons. One of the goals of this paper is indeed to compare the characteristics and the effectiveness of different mechanisms for realising internal states.

Given that internal states might be realised through several different mechanisms and given that these mechanisms might be also combined together we do not pretend to be exhaustive in our analysis. Indeed we will restrict our comparison and analysis only to some possible ways of realising the mechanisms (a) and (b) described above. Moreover, although in some cases different algorithms might be used to set the connection weights, we will restrict our analysis to neural controllers whose connection weights are evolved through a form of evolutionary algorithm [16]. The reason for this choice is twofold: (1) an evolutionary algorithm can be used to evolve the connection weights and other free parameters of the neural controllers independently from the particular neural architecture or neural model used, and (2) by only requiring a general criterion for evaluating how much evolving individuals are able to solve their adaptive task, they allow us to maximise the level of self-organisation and reduce the externally imposed constraints on the learning process with respect to other learning algorithms.

Before moving to the next section in which we will present our experimental setup, we should emphasize two important aspects.

First, an agent provided with a neural architecture with recurrent neural connections or other neural mechanisms that might allow it to extract internal states and use these internal states to co-determine its motor behaviour does not necessarily extract internal states or use them to co-determine its motor behaviour. In the case of the neural controller described in Figure 2, for example, due to a given configuration of the connection weights, the activation state of the gray hidden neuron might be always off or always on and therefore might not provide any information on the previous sensory states experienced by the robot. Or, the activation state of this hidden neuron might vary and might be affected by previous sensory states but it might not have any affect on the motor neurons. This implies that in order to ascertain whether an agents really is a pro-active agent we should analyze how activation states vary in time and how they influence the motor behaviour of the agent.

The second important aspect that we want to stress here is the fact that, as we will also see in the next sections, agents have often two options available in order to solve their adaptive tasks that consist of: (1) use sensory-motor coordination, that is act so to experience sensory states that allow to solve the problem through a reactive control mechanism [14], and (2) extract internal states and use them to co-determine the way in which the agents react to sensory states. Reactive solutions based on sensory-motor coordination are often simpler and easier to find through artificial evolution and are therefore preferred when available. This means that the emergence of pro-active control strategies only tend to be observed when reactive solutions and sensory-motor coordination are insufficient.

3 Self-localisation task

To investigate the issue described above we evolved the neural controllers of simulated robots that are asked to move and to self-localise in their environment [15] and we compared the results obtained by providing evolving robots with different types of neural controllers. More precisely the agent has to drive around a loopy corridor and to indicate with an output neuron in which room it is currently located. Figure 3 is a drawing of the environment for this task. The arrows indicate the direction in which the agents are forced to drive. The two rooms are painted in different shades of grey. If the agent is in the top room (light grey), the localisation output neuron has to have a value in the interval $[0, 0.5]$ to be correct. In the bottom room (dark grey) this value has to be in $\langle 0.5, 1]$. Figure 4 is also a drawing of the environment, displaying the zones in the environment that are used during evolution to stimulate agents to drive around in the environment.

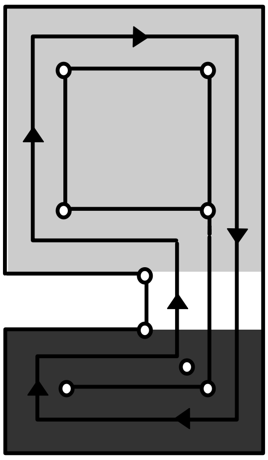

Figure 3: Environment and forced driving direction

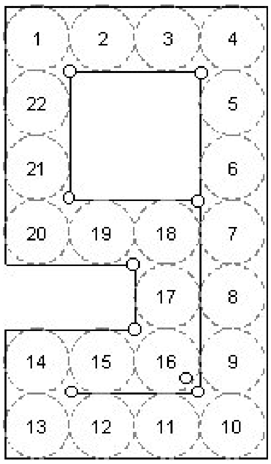

Figure 4: Zones in the environment

The agent we use for the self-localisation task is the Kephera robot [11] (shown in figure 5), a miniature mobile robot with a diameter of 55 mm and a weight of 70 g. It is supported by two lateral wheels that can rotate in both directions and two rigid pivots in the front and in the back. By spinning the wheels in opposite directions at the same speed, the robot can rotate without lateral displacement. The sensory system employs eight infrared sensors that are able to detect obstacles up to about 4 cm. Experiments were conducted in simulation by using an extended version of Evorobot [12]. In Evorobot, a sampling procedure is used to compute the activation state of the infrared sensors. Walls and cylindrical objects are sampled by placing one physical robot in front of them and by recording the state of the infrared sensors while the robot is turning 360 degrees at 20 different distances from of each object.

These recorded values are used in simulation to set the activation states of the simulated infrared sensors on the basis of the current angle and distance of the robot with respect to obstacles. This procedure allows to develop a very accurate simulation that takes into account the detailed characteristics of the individual robot used in the experiments [16].

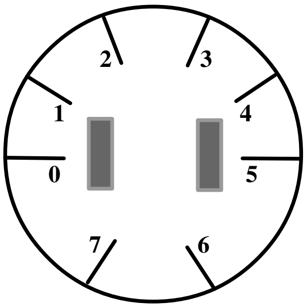

Figure 5: Diagram of a Kephera robot with its sensors

Each evolutionary run begins with an initial population that consists of 100 randomly generated genotypes. A genotype consists of a string of parameters that are encoded in the genotype with 8 bits. During evolution each individual of the population is allowed to 'live' for 4 epochs consisting of 2500 time steps (a time step lasts 100ms). In each epoch, the agent starts at a different position in the environment. The 20 fittest individuals of each generation are allowed to reproduce by generating 5 copies of their genotype with 2% of their bits replaced with a new randomly selected value. The process is repeated for 500 generations.

The fitness function is set up to reward first the ability of the robot to travel in a clockwise direction in the environment and then its ability to indicate in which room in the environment it is located.

$$F = \begin{cases} \frac{z_c}{z_t} & \text{, if } z_c < z_t \\ \\ 1 + (bottom * top) & \text{, if } z_c >= z_t \end{cases} \tag{1}$$

In which z_c is the number of zones that the robot has crossed in its lifetime. The zones are illustrated in figure 4. z_t is the zone threshold and determines how fast the robot has to drive, before its capacities for localisation are considered to determine its fitness. If the agent crosses more than z_t zones during its lifetime, the extra amount of zones has no effect on its fitness. The self-localisation output of the agent is measured continuously when it is inside one of the two rooms, but is only considered if the agent crosses the zone threshold during the epochs that it is executed. *bottom* and *top* are the percentages of good localisations in the bottom and top room, respectively. E.g., *bottom* is the number of time steps that the self-localisation output is in the interval $\langle 0.5, 1.0]$ and the agent is in the bottom room, divided by the total number of time steps that the agent is in the bottom room. Evidently, the maximal fitness that can be achieved is 2.

The self-localisation task requires the agent to use its internal state. The task is too difficult for a reactive agent, since the two different parts of the environment are largely the same from the viewpoint of the agent. If an agent has the same sensory inputs but it is required to take different actions, it faces a problem of perceptual aliasing [14]. An example of perceptual aliasing is that for the agent's sensors there is no difference between being in the top or bottom horizontal corridor, while the agent has to indicate a different room. A number of factors, such as the forced driving speed and the fact that the corridors are narrow, have as a consequence that reactive agents cannot 'escape' the perceptual aliasing by applying sensory-motor coordination. Experiments performed with reactive agents did not lead to successful individuals [15].

Five classes of experiments in which evolving agents were provided with different type of neural controllers were run. For each neural architecture three experiments with a different driving threshold (22, 23, and 25 rounds, corresponding to $z_t = 440$, $z_t = 506$, and $z_t = 550$, respectively) were run. For each experiment 10 replications were performed. In section 4 we describe the five different neural models used. In section 5 we describe the obtained results.

4 Five types of recurrent neural controllers

In this section we describe the five types of neural models used to conduct the experiments. All models might allow evolving robots to extract internal states and use these states to co-determine the agents' behaviour. However, different models rely on different neural mechanisms.

4.1 Elman network: EN

The Elman network [6] consists of a neural network with a sensory layer, a layer of hidden neurons, and an output layer. The activation state of the hidden neurons at time t-1 is copied into an additional set of input units at time t.

The architecture used in our experiments consists of 10 sensory neurons, 5 hidden neurons, and 3 output neurons. Two of the output neurons indicate the desired speed of the wheels. We will refer to them as 'motor neurons'. The agent has to indicate with the third output neuron in which room the agent is located. We will refer to this neuron as 'self-localisation output'. The sensory neurons encode the activation state of the 8 infrared sensors, and the activation state of the two motor neurons at time t-1.

Hidden and output neurons are activated according to the logistic function. More precisely the activation function of each neuron is:

$$a_i(t) = \sigma(netinput_i(t) + bias_i + in_i(t)) \tag{2}$$

$$netinput_i(t) = \sum_{j=1}^{N} w_{ji} a_j(t), \tag{3}$$

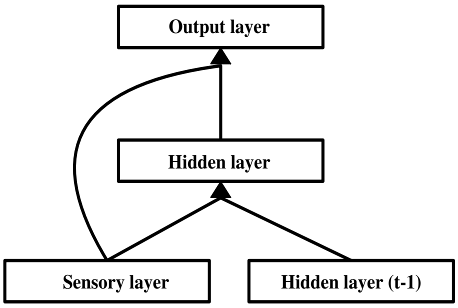

Figure 6: The architecture of the Elman network. Boxes represent collections of neurons. Arrows indicate the connection between collections of neurons (all neurons of the first box are connected to all neurons of the second box).

in which $a_i(t)$ denotes the activation of neuron i at time t and σ is the logistic function, $\sigma(x) = \frac{1}{1+e^{-x}}$. N is the number of neurons connected to neuron i. In the case of the hidden neurons this is the number of sensory neurons plus the number of neurons in the hidden layer. w_{ji} is the weight of the connection from neuron j to neuron i. The external input is represented by $in_i(t)$.

During the evolutionary process the architecture is kept fixed. Only the biases and the synaptic strengths of the connections are encoded in the genotype and allowed to change. All parameters are encoded in the genotype with 8 bits. Connection weights and biases are then normalised in the range [-5.0, 5.0].

4.2 Non-linear autoregressive model with exogeneous inputs: NARX

Nonlinear autoregressive neural networks with exogeneous inputs [7] are an extension of Elman Networks in which the activation state of the sensory neurons at time t, $t-1$, ..., $t-c_{in}$, and the activation state of the output neurons at time $t-1$, $t-2$, ..., $t-c_{out}$ determine the activation of the output and the hidden neurons at time t. The activation of the hidden neurons is also determined by the activation state of the hidden neurons at time t-1.

The architecture used in our experiments consists of 8 sensory neurons, 5 hidden neurons, and 3 output neurons. The sensory neurons encode the activation state of the 8 infrared sensors. The output neurons encode the desired speed of the two wheels and the self-localisation output.

Hidden and output neurons are activated according to the logistic function. During the evolutionary process the architecture is kept fixed. Only the synaptic strengths of the connections are encoded in the genotype and allowed to change. All parameters are encoded in the genotype with 8 bits. Connection strengths and biases are then normalised in the range [-5.0, 5.0].

4.3 Dynamical Neural Network: DNN

Dynamical neural networks [17, 15] are neural networks constituted by dynamical artificial neurons, that is neurons that tend to vary their activation state

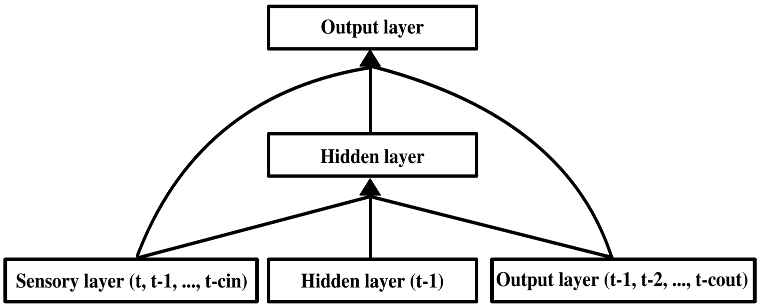

Figure 7: The architecture of the NARX network. Boxes represent collections of neurons. Arrows indicate the connection between collections of neurons (all neurons of the first box are connected to all neurons of the second box).

at different time rate according to a time constant parameter.

Neurons are updated according to the following functions:

$$a_i(t) = tc_i a_i(t-1) + (1 - tc_i)\sigma(netinput(t) + bias + in_i(t)), \tag{4}$$

where tc_i is the parameter adjusted by the evolutionary algorithm that determines the proportion of neural inertia, $tc_i \in [0, 1]$. $netinput(t)$ is defined as in equation 7. In the DNNs that we apply to the self-localisation task, only the sensory neurons and the hidden neurons apply activation function 4. As a consequence, both the activations of the sensory neurons and of the hidden neurons are part of the internal state. The activations of the hidden neurons influence future input-output mappings of the agent in two ways: by serving as neural input (through the recurrent connections) and by imposing a neural inertia.

The architecture used in our experiments consists of 10 sensory neurons, 5 hidden neuron, and 3 output neurons. The sensory neurons encode the activation state of the 8 infrared sensors, and the activation state of the motor neurons at time t-1.

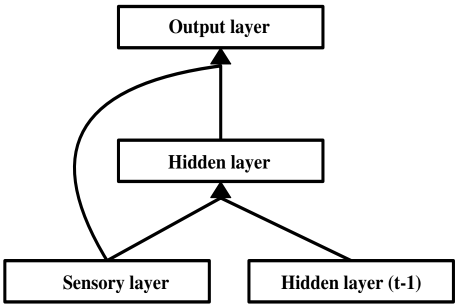

Figure 8: The architecture of the dynamical neural network (DNN). Boxes represents collection of neurons. Arrows indicate the connection between collections of neurons (all neurons of the first box are connected to all neurons of the second box).

During the evolutionary process the architecture is kept fixed. Only the time constants of neurons and the synaptic strengths of the connections are encoded in the genotype and allowed to change. All parameters are encoded in the genotype with 8 bits. Connection weights and biases are then normalised in the range [-5.0, 5.0], time constants are normalised in the range [0.0, 1.0]. The time constant of output neurons was set to 0.0 and was not allowed to change.

4.4 Continuous time recurrent neural network: CTRNN

As dynamical neural networks, continuous time recurrent neural networks [2] are neural networks constituted by dynamical artificial neurons. Also in this case, the activation state of neurons is influenced by their previous activation state. In the case of CTRNN, however, the state of a neuron is characterised by two variables: the activation potential (that corresponds to the depolarisation of the neuron membrane, in the case of real neurons), and the activity of the neuron (that corresponds to the frequency of the spikes produced, in the case of real neurons). The time constant parameter, in this case, determines the rate of change of the activation potential of the neuron.

More precisely, neurons are updated according to the following function:

$$a_i = \sigma(p_i + bias_i) \tag{5}$$

$$\dot{p}_i = \frac{1}{tc_i}(-p_i + netinput_i + g\ in_i) \tag{6}$$

$$netinput_i = \sum_{j=1}^{N} w_{ji} a_j, \tag{7}$$

where p_i is the activation potential of a neuron, g the gain of the inputs, and tc_i is the time constant of the neuron. The time constant is adjusted by the evolutionary algorithm, $\frac{1}{tc_i} \in [0, 1]$. Only the sensory neurons and the hidden neurons apply activation function 5, the outputs apply activation function 2. As a result, the activation potentials of the sensory neurons, the activation potentials of the hidden neurons, and the activations of the hidden neurons are part of the internal state. The activation potentials result in a neural inertia, while the activations of the hidden neurons serve as neural input to the hidden layer for the next time step.

We approximate the dynamics of the differential equation by using the standard Euler method (see [8]), with step size 0.1. The architecture and the parameters encoded in the genotype are the same as those described in section 4.3, but then with $\frac{1}{tc_i} \in [0, 1]$.

4.5 Time delay recurrent neural network: TDRNN

In a time delay recurrent neural network [5] the propagation of activation through connections takes time and the time delay is controlled by a parameter associated to each connection. For analysis purposes we have used a restricted form of a TDRNN in which each neuron has one common time delay for all its incoming connections.

The architecture used in our experiments consists of 10 sensory neurons, 5 hidden neurons, and 3 output neurons (figure 4.5). The sensory neurons encode the activation state of the 8 infrared sensors, and the activation state of the motor neurons encoding the desired speed at time t-1. The output neurons encode the desired speed of the two wheels and the self-localisation output.

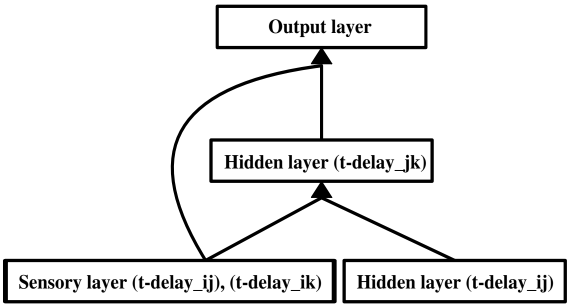

Figure 9: The architecture of the time delay recurrent neural networks (TDRNN). Boxes represents collection of neurons. Arrows indicate the connection between collection of neurons (all neurons of the first box are connected to all neurons of the second box).

Neurons are updated according to the logistic function. The genotype of evolving individuals encodes the strength and the time delay of each connection. All parameters are encoded in the genotype with 8 bits. Connection strength and biases are then normalised in the range [-5.0, 5.0]. Time delays are normalised in the range [0, 50] time steps corresponding to a delay in the propagation of the activation ranging from [0.0, 5.0] seconds.

5 Results

By running the evolutionary experiments we observed that all evolved individuals were able to travel in the environment at the required speed. However, with respect to the ability to self-localise, performance varied significantly for different neural controllers and for different driving thresholds. The performances of the best evolved agents of the five types of neural controllers applied to the three driving thresholds are shown in table 1. By analysing the performance of the best individuals of each of the five experiments in which evolving agents were provided with five different neural controllers, in fact, we observed that in all cases at least one individual evolved that is able to self-localise more than 75% of the times. At a driving threshold of 23 rounds, instead, in the case of the experiment in which agents were provided with EN networks, no individuals were able to self-localise correctly more than 75% of the times (Table 1). Finally, table 1 shows that only the DNN, CTRNN, and TDRNN neural controllers achieve good performances in the case of a driving threshold of 25 rounds .

Neural controller	20 rounds	23 rounds	25 rounds
EN	**91**	74	67
NARX	**86**	**89**	75
DNN	**97**	**93**	**92**
CTRNN	**86**	**80**	**94**
TDRNN	**81**	**82**	**79**

Table 1: Self-localisation performance in percentages of the best individual of the best replication for experiments with different neural controllers and different driving thresholds. Performances in bold (i.e. performance equal or above 1.56) indicate individuals that are able to correctly self-localise more than 75% of the times. Performances are averaged over 100 runs.

As we will see in the next sub-sections, these results can be explained by considering that evolving agents provided with EN and NARX neural controllers are unable to extract internal states that encode long term regularities and rely on simple quasi-reactive strategies that exploit sensory-motor coordination to solve the self-localisation problem. These simple strategies are based on the fact that, by producing different motor behaviours in different environmental conditions, robots might experience later on different type of sensory states even in identical environmental areas, if these areas are preceded by different environmental structures. These simple strategies however do not allow evolving agents to achieve optimal or close to optimal performance with respect to the self-localisation problem, especially with high driving thresholds that force agents to move quickly in the environment.

On the contrary, TDRNN evolved controllers exploit the time delay on activity propagation, so that past sensory states have long term effects. As a consequence, they still display good performance in the case of high driving thresholds.

Finally, evolved agents that are provided with DNN and CTRNN evolved controllers, are able to extract from sensory states internal states encoding long term regularities that allow these agents to display good and, in some replications, optimal performance.

In the next sections we will describe the control strategies developed by agents provided with different neural networks in detail.

5.1 EN

EN-Agents are able to display reasonably good performance only with the lowest driving threshold (20 rounds) and only in two out of ten replications of the experiments. By analysing the behaviour and the activity of internal neurons of the best evolved individuals of the two best replications we realised that they use the same strategy to solve to the problem. Figure 5.1 shows the behaviour of the best of these two individuals.

By looking at the activity of hidden neurons we can see that they tend to converge on two rather different equilibrium points corresponding to [0.95,

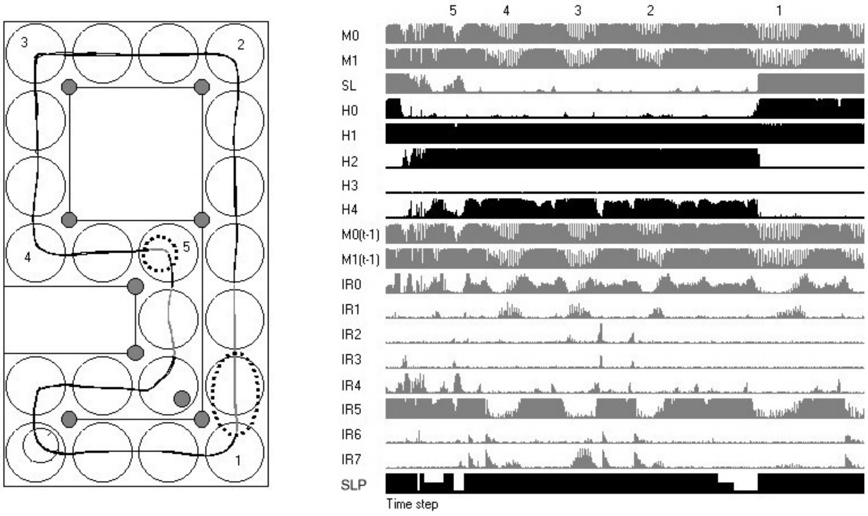

Figure 10: Trajectory and neural activity of the best evolved EN-agent in the case of the experiment with a driving threshold of 20. **Left:** the environment and the robot trajectory during a few laps of the corridor. Dotted circles indicate the areas in which the self-localisation output produced by the agent is wrong. The numbers (from 5 to 1) indicate critical points from the point of view of the ability of this agent to self-localise. **Right:** the activation state of neurons while the robot is performing the last lap of the environment. The activation value is indicated by the height of the graph with respect to the baseline. M0, M1 and SL indicate the activity of the two motor neurons and of the self-localisation output unit. H0-H4 indicate the activity of the 5 internal neurons. M0(t-1), M1(t-1) and IR0-IR7 indicate the activity of the two input units that encode the state of the two corresponding motor neurons at time t-1 and the activity of the 8 infrared sensors. SLP indicates the performance with respect to self-localisation. In this case, the height with respect to the baseline indicates respectively, when the self-localisation is correct (full height), wrong (null height), or when the agents is traveling between the two rooms (half height).

1.0, 0.02, 0.05, 0.07] and [0.03, 1.0, 1.0, 0.05, 0.84]. The transition between these two equilibrium states occurs very quickly approximately when the agents move from one room to the other. The former equilibrium state of the internal neurons and the connection weights from these neurons and the self-localisation output neuron assure that the self-localisation output is low, when the robot is in the upper room, and high, when the robot is in the lower room, as requested.

The transition between the two equilibrium states depends on the state of sensors and motors during the few time steps that precede the transition.

More specifically, in the case of the transition between the top and the bottom room, the transition between the former and the latter equilibrium point occurs when the agent negotiates the bottom-right corner of the environment. During the negotiation of this particular corner, in fact, given that the agent reaches the corner by being very close to the wall on its right side, the activation of the two back infrared sensors is almost null. This particular sensory-motor situation, that is the fact that the activity of the right motor neuron decreases in order to turn and negotiate the corner, and the fact that state of the two back infrared sensors is almost null, causes the first transition. This hypothesis has been further verified by freezing the activation state of the two back neurons to 0.05. In this case, in fact, the agents always indicate that it is in the bottom room. This strategy also explains why agents are unable to self-localise correctly in the bottom room before reaching the bottom-right corner. The fact that the robot reaches the bottom-right room by staying very close to its right-side walls is due to the fact that this evolved agent progressively approaches its right side walls while traveling in a corridor and the fact that this corner is preceded by the longest corridor of the environment.

In the case of the transition between the bottom and the top room, the transition between the latter and the former equilibrium point occurs when the agent negotiates the second left-handed corner which is located just at the beginning of the top room. Also in this case, the agents is able to discriminate between the first and the second left-handed corner on the basis of the state of the sensors and of the motors during the few time steps that precede the transition. More specifically, the transition between the latter and the former equilibrium point occurs when the agent negotiates a left-handed corner (i.e. when the activation state of the left motor neurons is lower than the activation of the right motor neuron) and the activation state of the IR3 and IR4 infrared sensors placed on the frontal-right side of the robot (see figure 5) are low. The fact that the activation of IR3 and IR4 tend to be low during the negotiation of the second left-handed corner in turn is the result of the fact that the robot tends to stay close to the wall on its left side after negotiating the first left-handed corner and the presence of an obstacle on the right side of the first left-handed corner. The discrimination between the two left-handed corners, however, is sub-optimal. In fact, due to an increase of the activation of IR4 during the negotiation of the second left-handed corner, wrong self-localisation output are produced (see figure 5.1).

5.2 NARX

Similarly to EN-agents, evolved NARX-agents are able to display reasonably good performance with low driving threshold (20 and 23 turns) but not with high driving threshold (25 turns). Moreover, like in EN-agents, the ability of NARX-agents to self-localise is based on the tendency to converge on two equilibrium states and to move from one to the other equilibrium state on the basis of few sensory and motor states preceding the transition.

Table 2 and 3 show the results obtained by running additional experiments with a driving threshold of 20 and 23 in which we also varied the number of previous sensory and motor states that are copied into additional sensory neurons.

By analyzing the behaviour and the internal states of the best evolved individuals we observed that the strategy of the agents evolved with a driving threshold of 20 are similar to the strategy described in the previous section. Below, we only describe the strategy adopted by the best individual obtained

	NARX	NARX	NARX	NARX
c_{in}	0	1	3	4
c_{out}	5	4	2	1
$\overline{F}_{best}$	1.35	1.75	1.61	1.52

Table 2: Performance of the best individual of the best out of ten replications in four experiments with different c_{in} and c_{out} numbers. In all experiments the number of internal neuron is 5 and the driving threshold is 20. $\overline{F}_{best}$ is the performance of the best evolved agent averaged over 100 runs.

	NARX	NARX	NARX	NARX
c_{in}	0	1	3	4
c_{out}	5	4	2	1
$\overline{F}_{best}$	1.72	1.80	1.83	1.61

Table 3: Performance of the best individual of the best out of ten replications in four experiments with different c_{in} and c_{out} numbers. In all experiments the number of internal neurons is 5 and the driving threshold is 23. $\overline{F}_{best}$ is the performance of the best evolved agent averaged over 100 runs.

in the experiment in which the driving threshold is 23, c_{in} is 1 and c_{out} is 4. The analysis of the other evolved individuals with different values of c_{in} and c_{out}, in any case, revealed that they adopt similar strategies (result not shown).

In this case, as illustrated in figure 5.2, the two equilibrium states are not encoded at the level of the internal neurons but directly in the state of the self-localisation output unit that tends to maintain its activation state close to 0.0 or 1.0. These two states tend to be maintained due to the large positive connection weights of the four connections that link the sensory units encoding the previous activation state of the self-localisation output unit to the unit itself.

The transition between the former and the latter equilibrium points (that correspond to the top and the bottom room, respectively) is triggered by an high activation of the left sensor (IR0), a low activation of the IR5 (i.e. the right sensor), and a null or close to null activation of IR2, IR3 and IR7 (that are activated during the negotiation of a corner). The fact that these conditions are only met when the robot reaches the middle part of the long corridor on the right side of the environment, is due to the particular way of traveling along corridors selected by this agent. As shown in figure 5.2, in fact, this agents produces a curvilinear trajectory in corridors by approaching first the wall placed on the right side of the agent and then, after a certain length, the wall placed on the left side. This curvilinear trajectory assures that the agent approaches the left side walls only in corners or at about the middle part of the long corridor.

The transition between the latter and the former equilibrium points (that correspond to the bottom and the top room, respectively) is triggered by the sensory inputs that are specific to the second turn to the left. The back sensors

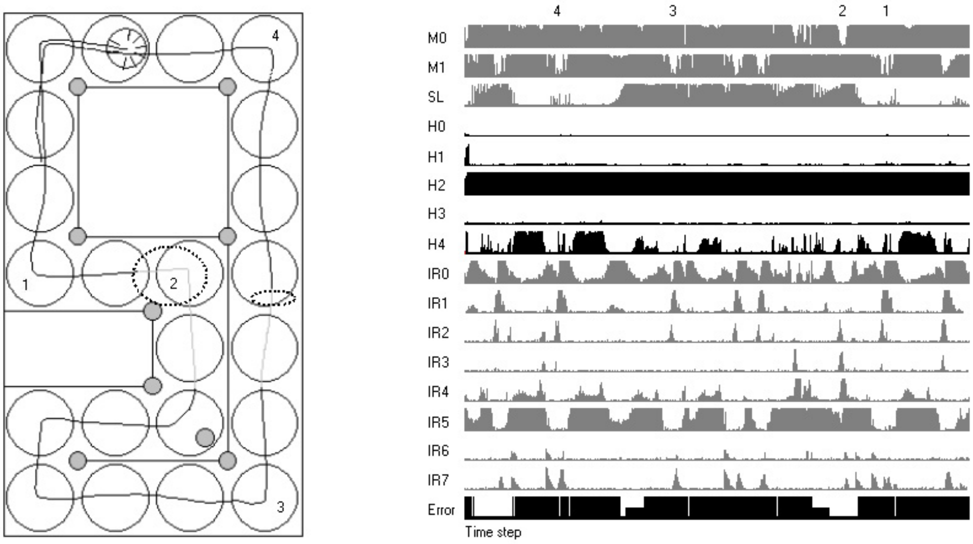

Figure 11: Trajectory and neural activity of one of the best evolved NARX-agents in the case of the experiment with a driving threshold of 23. **Left:** the environment and the robot trajectory during few laps of the corridor. The numbers (from 4 to 1) indicate critical points from the point of view of the ability of this agent to self-localise. Dotted areas indicate the areas in which the self-localisation output produced by the agent is wrong. **Right:** the activation state of neurons while the robot is performing the last lap of the environment. The activation value is indicated by the height of the graph with respect to the baseline. M0, M1 and SL indicate the activity of the two motor neurons and of the self-localisation output unit. H0-H4 indicate the activity of the 5 internal neurons. M0(t-1), M1(t-1) and IR0-IR7 indicate the activity of the two input units that encode the state of the two corresponding motor neurons at time t-1 and the activity of the 8 infrared sensors. SLP indicates performance with respect to self-localisation. In this case, the height with respect to the baseline indicate respectively, when the self-localisation is correct (full height), wrong (null height), or when the agents is traveling between the two rooms (half height).

(IR6 and IR7) are activated in the second turn to the left while IR0 shortly decreases.

As in the case of EN-agents, the fact that NARX-agents are unable to produce reasonably good performance when the driving threshold is 25 can be explained by considering that, by being asked to move at higher speed in the environment, evolving agents cannot select peculiar ways of negotiating corridors and corners (such as moving in corridors by producing curvilinear trajectories) that in turn allow them to identify the location of critical areas of the environment on the basis of a single or few sensory-motor states.

5.3 DNN

DNN-Agents are able to display reasonably good performance, and in some replication close to optimal performance, at all driving thresholds. By analysing the behaviour and the activity of neurons of the best evolved individuals we observed that the self-localisation problem tends to be solved by relying on a few or a single hidden neuron that slowly changes its activation state by always keeping its state below and above a given threshold in the top and in the bottom room respectively, or viceversa. Figure 12 shows the behaviour and the neural activity of the best evolved individual of the experiment with a driving threshold of 25.

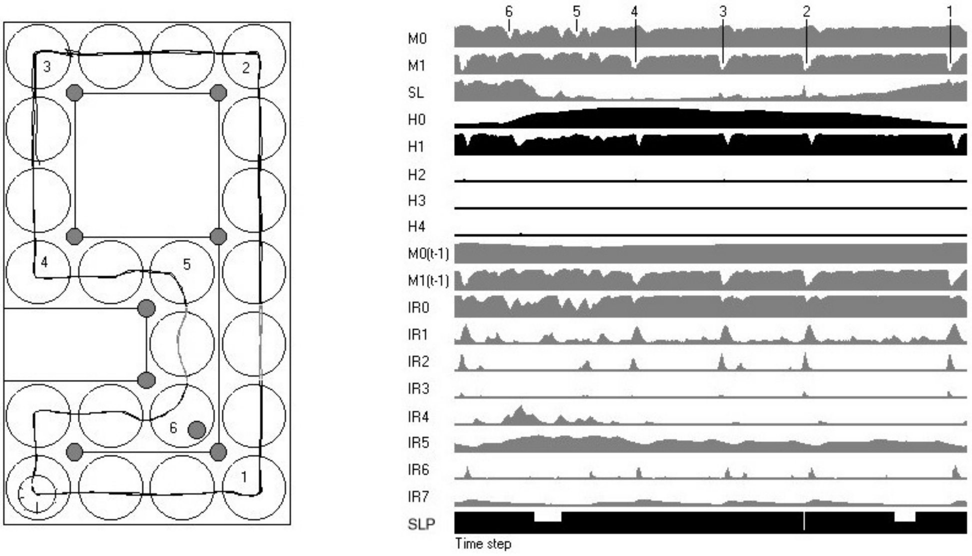

Figure 12: Trajectory and neural activity of the best evolved DNN-agent in the case of the experiment with a driving threshold of 25. **Left:** the environment and the robot trajectory during few laps of the corridor. The numbers (from 6 to 1) indicate critical points from the point of view of the ability of this agent to self-localise. **Right:** the activation state of neurons while the robot is performing the last lap of the environment. The activation value is indicated by the height of the graph with respect to the baseline. M0, M1 and SL indicate the activity of the two motor neurons and of the self-localisation output unit. H0-H4 indicate the activity of the 5 internal neurons. M0(t-1), M1(t-1) and IR0-IR7 indicate the activity of the two input units that encode the state of the two corresponding motor neurons at time t-1 and the activity of the 8 infrared sensors. SLP indicates performance with respect to self-localisation. In this case, the height with respect to the baseline indicates respectively, when the self-localisation is correct (full height), wrong (null height), or when the agents is traveling between the two rooms (half height).

As can be seen in figure 12, the activity of the self-localisation output units is mainly affected by the activity of H0 due to a strong inhibitory connection coming from this hidden unit. Given that time constant parameter of H0 is very high (0.96), the activity of this neuron tends to change slowly in time.

The fact that the activity of H0 tends to decrease slowly while the robot moves along corridors ensures that the activation of this unit progressively decreases while the robot moves along the long corridor located on the right side of the environment. As a consequence, it reaches a value below the critical threshold during the transition from the top to the bottom room.

The fact that the activity of unit H0 tends to increase slightly during the negotiation of right-handed corners assures that the activity of this unit is always above the critical threshold while the robot moves in the top room. Finally, the fact that the activity of H0 tends to increase quickly during the negotiation of left-handed corners assures that the activity of this unit overcomes the critical threshold during the transition from the bottom to the top room.

The ability to integrate information from long sequences of sensory-motor states to detect, for instance, the length of corridors, allows DNN agents to solve their problem without the need to rely on sensory-motor coordination strategies. As we saw above, sensory-motor coordination strategies might allow agents to self-localise correctly on the basis of regularities extracted by few sensory-motor states, but require special ways to negotiate the environment that do not allow to move at high speeds.

5.4 CTRNN

CTRNN-Agents are able to display good performances at all driving thresholds. By analysing the behaviour and the activity of neurons of the best evolved individuals we observed that the agents solve the self-localisation problem with strategies very similar to those exhibited by DNN-Agents. Figure 13 shows the behaviour and the neural activity of the best evolved individual of the experiment with a driving threshold of 25 rounds.

As can be seen in Figure 13, the activity of the self-localisation output is mainly affected by the activity of H3 due to a strong inhibitory connection coming from this hidden unit. Given that the time constant parameter of H3 is high ($\frac{1}{tc} = 0.02$) , the activity of this neuron tends to change slowly in time. As in the case of DNN, the fact that the activity of H3 tends to decrease slowly while the robot moves along corridors ensures that the activation of this unit progressively decreases while the robot moves along the long corridor located on the right side of the environment by reaching a value below the critical threshold during the transition from the top to the bottom room.

As in the case of DNN, the fact that the activity of unit H3 tends to increase slightly during the negotiation of right-handed corners assures that the activity of this unit is always above the critical threshold while the robot moves in the top room. Finally, the fact that the activity of H3 tends to increase quickly during the negotiation of left-handed corners assures that the activity of this unit overcomes the critical threshold during the transition from the bottom to the top room. In this case however, H3 overcomes the critical threshold

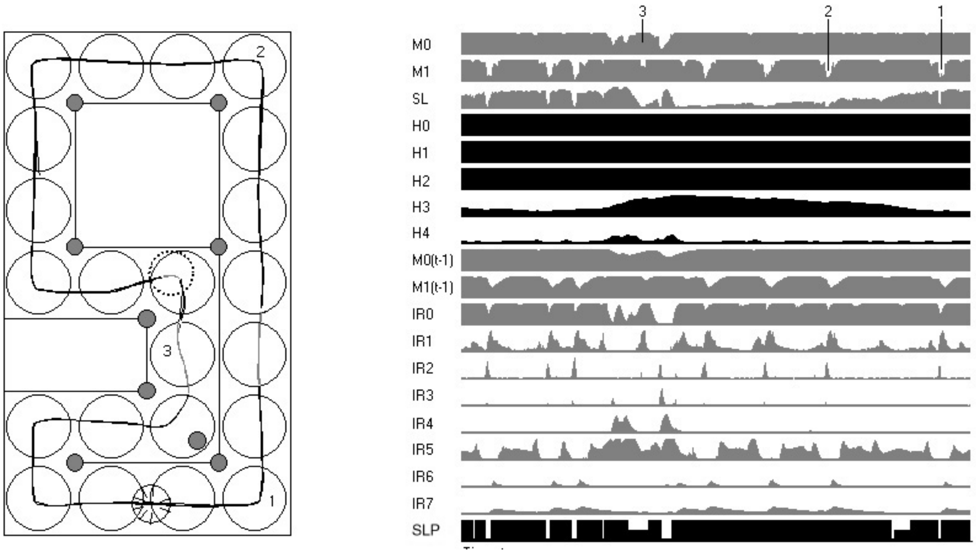

Figure 13: Trajectory and neural activity of the best evolved CTRNN-agent in the case of the experiment with a driving threshold of 25. **Left:** the environment and the robot trajectory during few laps of the corridor. The numbers (from 6 to 1) indicate critical points from the point of view of the ability of this agent to self-localise. Dotted areas indicate the areas in which the self-localisation output produced by the agent is wrong. **Right:** the activation state of neurons while the robot is performing the last lap of the environment. The activation value is indicated by the height of the graph with respect to the baseline. M0, M1 and SL indicate the activity of the two motor neurons and of the self-localisation output unit. H0-H4 indicate the activity of the 5 internal neurons. M0(t-1), M1(t-1) and IR0-IR7 indicate the activity of the two input units that encode the state of the two corresponding motor neurons at time t-1 and the activity of the 8 infrared sensors. SLP indicates performance with respect to self-localisation. In this case, the height with respect to the baseline indicates respectively, when the self-localisation is correct (full height), wrong (null height), or when the agents is traveling between the two rooms (half height).

already after the first left-handed corner. The combination of this fact and the fact that IR4, that gets activated during the negotiation of left-handed corners, contributes to activate the self-localisation output unit, causes a systematic localisation error during the negotiation of the second left-handed corner (see the dotted area indicated on the right side of figure 13).

5.5 TDRNN

TDRNN-agents are able to display reasonably good performance at all driving thresholds (see table 1). The analysis of the best-evolved agents indicates that the time delay on activity propagation plays an important role in the ability of these agents to self-localise. Figure 14 shows the trajectory and neural activations of the best TDRNN-agent evolved with a driving threshold of 23 rounds. The values of the time delay parameters are shown in table 4.

The analysis of the evolved connection strengths and time delay parameters indicate that evolved agents use time delay parameters to: (1) detect a sequence of events separated by fixed time intervals, and (2) allow internal states to produce motor effects after a fixed time interval.

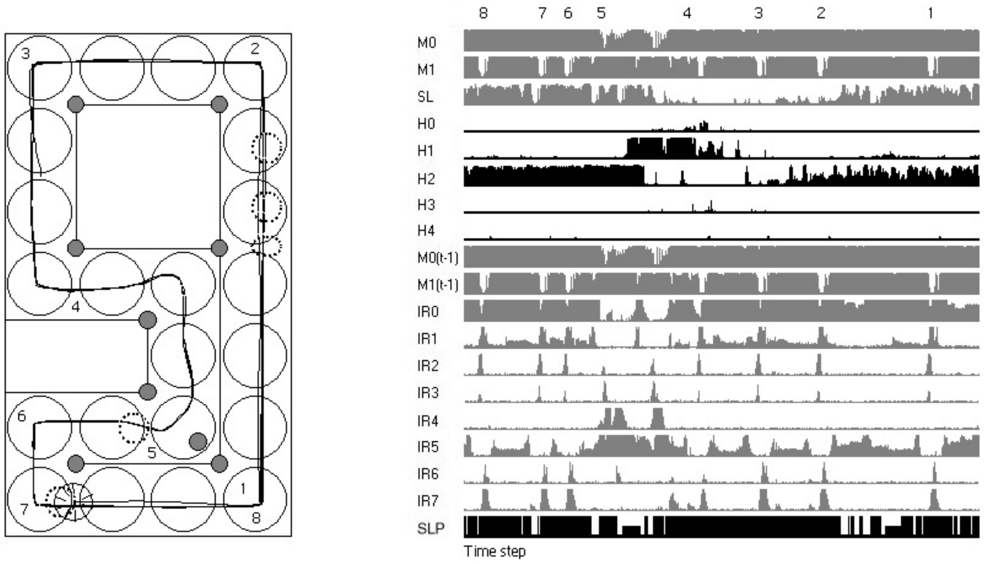

Figure 14: Trajectory and neural activity of one of the best evolved TDRNN-agent in the case of the experiment with a driving threshold of 25. **Left:** the environment and the robot trajectory during few laps of the corridor. The numbers (from 8 to 1) indicate critical points from the point of view of the ability of this agent to self-localise. **Right:** the activation state of neurons while the robot is performing the last lap of the environment. The activation value is indicated by the height of the graph with respect to the baseline. M0, M1 and SL indicate the activity of the two motor neurons and of the self-localisation output unit. H0-H4 indicate the activity of the 5 internal neurons. IR0-IR7 indicate the activity of the 8 infrared sensors. The additional input neurons encoding the state of the sensors and motors in previous time steps are not displayed but can be inferred by previous sensory and motor states. SLP indicate performance with respect to self-localisation. In this case, the height with respect to the baseline indicates respectively, when the self-localisation is correct (full height), wrong (null height), or when the agents is traveling between the two rooms (half height).

Let us consider in particular how the evolved individual shown in figure 14 is able to correctly indicate the transition from the bottom to the top room. The activity of the self-localisation output unit of the TDRNN is determined by the delayed activations of the sensory neurons and hidden neurons. In fact, the indication of the transition from the bottom to the top room depends on the sensory inputs that the agent experiences when it enters the first turn to the left, indicated with the number '5' in figure 14. The sensory neurons that are especially involved are I0, I3, I4, and I5. Their particular values have three effects.

1. The self-localisation output decreases after 43 time steps.
2. The activation of H1 increases 22 time steps later.
3. The activation of H2 decreases 36 time steps later.

The increase of H1 and decrease of H2 has as a consequence that SL remains low while the agent is traversing almost the entire top room. The explanation for this is that H1 inhibits H2 and SL, while H2 excites SL.

The agent indicates the transition of the top to the bottom room as follows. After the activation of H1 has been decreased, the activation of H2 increases. The main reason for this is the lack of inhibition from H1 and the excitation due to the sensory inputs belonging to turns to the right. For example, M1(t-1) inhibits H2 but has low values in turns to the right.

Neuron	Time delay
M0	0
M1	0
SL	43
H0	10
H1	22
H2	36
H3	46
H4	6

Table 4: Values of the time delay parameters (in time steps) of all hidden and output neurons. All incoming connections to one neuron have a common delay.

To summarize, the TDRNN-agent extracts an internal state from the sensory signals it experiences between the time steps indicated with '5' and '4'. Besides delaying the signals from the sensory neurons, the agent exploits the recurrency of the hidden layer to let the effects of the sensory signals experienced in the turns to the left fade away. The effect of this strategy is that the agent has a low self-localisation output during most of its traversal of the top room and a high self-localisation output in the bottom room.

6 Discussion

The comparison of the results obtained by providing evolving agents with different types of neural controllers indicate that the use of dynamical neurons and/or time-delayed propagation of activation potentials might constitute a necessary prerequisite for the emergence of the ability to integrate sensory-motor information through time.

In fact, although in principle agents provided with simple recurrent neural networks such as EN or NARX neural networks should be able to develop the same control strategies developed by agents provided with DNN, CTRNN and TDRNN neural networks, in practice they are unable to do so. This failure can be explained by considering that the evolvability (i.e. the probability to produce a better solution through random changes of free parameters) of EN and NARX neural networks is lower than that of DNN, CTRNN and TDRNN.

The fact that agents provided with a DNN or CTRNN neural controller are more evolvable than agents provided with EN or NARX can be explained by considering that the availability of neurons that tend to vary their state at different time rates is a useful prerequisite to solve problems that require to integrate information from sequences of sensory-motor states or to produce motor states lasting several time steps [15]. Although by properly setting the connection weights, any type of recurrent neural networks could in principle display neurons that tend to vary their activity at different time rates, neurons that vary their activity at slow time rates (i.e. time rates that are significantly slower from the time rates with which the activity of sensors and neurons are updated) are much more frequent in DNN and CTRNN.

The fact that neurons in DNN tend to vary their activity at a slower time rate than neurons in EN and NARX networks can be demonstrated by considering that the change of activation of a DNN-neuron is always smaller or equal to the change of a neuron in EN and NARX neural networks (i.e. neurons updated according to the standard logistic function). Indeed, the change in activation of a neuron in a DNN is always smaller or equal to the change of activation of a neuron in an EN or NARX, if the neurons have the same bias weight, neural inputs, and past activation, and if $\frac{1}{tc} \in [0, 1]$. First we express the neural activation function of the DNN (equation 9) in terms of the activation function of the EN (equation 8), as shown in equation 10.

$$a_{en}(t) = \sigma(\text{netinput}(t) + \text{bias} + \text{in}(t)) \tag{8}$$

$$a_{dnn}(t) = \frac{1}{tc}a(t-1) + (1 - \frac{1}{tc})\sigma(\text{netinput}(t) + \text{bias} + \text{in}(t)) \tag{9}$$

$$a_{dnn}(t) = \frac{1}{tc}a(t-1) + (1 - \frac{1}{tc})a_{en}(t) \tag{10}$$

Since $\frac{1}{tc} \in [0, 1]$ and a_{dnn} is a weighted sum of $a(t-1)$ and $a_{en}(t)$, we can conclude equation 11 and 12 from equation 10.

$$a_{dnn}(t) \in [min\{a_{en}(t), a(t-1)\}, max\{a_{en}(t), a(t-1)\}] \tag{11}$$

$$|a_{dnn}(t) - a(t-1)| \leq |a_{en}(t) - a(t-1)| \tag{12}$$

Equation 12 implies that the change in activation of a neuron in a DNN is always smaller than or equal to that of a neuron in an EN, for the same neural input, bias, past input, and external input.

The fact that agents provided with a TDRNN neural controller are more evolvable than agents provided with EN and NARX can be explained by considering that the availability of neurons that encode the state of sensors and motors at previous time steps within an adaptable time range is a useful prerequisite to integrate information from sequences of sensory-motor states and, as we have seen, to detect sequences of events separated by a given time interval.

Overall the obtained results suggest that a better understanding of the neural mechanisms suitable to process information in time might be an important step towards the development of powerful pro-active agents.

Acknowledgements

This research has been partially supported by the ECAGENTS project founded by the Future and Emerging Technologies program (IST-FET) of the European Community under EU R&D contract IST-2003-1940. The information provided is the sole responsibility of the authors and does not reflect the Community's opinion. The Community is not responsible for any use that may be made of data appearing in this publication.

References

[1] R.D. Beer. A dynamical systems perspective on agent-environment interaction. *Artificial Intelligence 72*, pages 173–215, 1995.

[2] R.D. Beer. On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior 3, (4)*, pages 469–509, 1995.

[3] R.A. Brooks. Intelligence without reason. *J. Mylopoulos and R. Reiter, editors. Proceedings of 12th International Joint Conference on Artificial Intelligence.*, 1991.

[4] A. Clark. *Being There: Putting Brain, Body and World Together Again.* Cambridge, MA: MIT Press., 1997.

[5] R. J. Duro and J. S. Reyes. Ecg beat classification with synaptic delay based artificial neural networks. *IWANN*, pages 962–970, 1997.

[6] J. L. Elman. Finding structure in time. *Cognitive Science 14*, pages 179–211, 1990.

[7] T. Lin, B. G. Horne, P. Tino, and C. Lee Giles. Learning long-term dependencies in narx recurrent neural networks. *IEEE Transactions on Neural Networks*, 1996.

[8] B. Mathayomchan and R.D. Beer. Center-crossing recurrent neural networks for the evolution of rhythmic behavior. *Neural Computation*, 14:2043–2051, 2002.

[9] H.R. Maturana and F.J. Varela. *Autopoiesis and cognition: the realization of the living.* Dordrecht: reidel., 1980.

[10] H.R. Maturana and F.J. Varela. *The tree of knowledge: the biological roots of human understanding.* 1988.

[11] R. Mondada, E. Franzi, and P. Ienne. Mobile robot miniaturization: A tool for investigation in control algorithms. *T.Y. Yoshikawa and F. Miyazaki, editors. Proceedings of the Third International Symposium on Experimental Robots.*, 1993.

[12] S. Nolfi. Evorobot 1.1 user manual. *http://gral.ip.rm.cnr.it/evorobot/simulator.html.*

[13] S. Nolfi. Evolving non-trivial behaviors on real robots: A garbage collecting robot. *Robotics and Autonomous System, 22*, pages 187–198, 1997.

[14] S. Nolfi. Power and the limits of reactive agents. *Neurocomputing, 49*, pages 119–145, 2002.

[15] S. Nolfi, G. Baldassare, and D. Marocco. Evolving robots able to self-localize in the environment: The importance of viewing cognition as the result of processes occurring at different time scales. *T. Asakura and K. Murase, editors. Proceedings of the Third International Symposium on Homan and Artificial Intelligence Systems. Fukui, Japan*, 2002.

[16] S. Nolfi and D. Floreano. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines.* Cambridge, MA, MIT Press/Bradford Books, 2000.

[17] S. Nolfi and D. Marocco. Evolving robots able to integrate sensory-motor information over time. *Theory in Biosciences, 120*, pages 287–310, 2001.

[18] R. Pfeifer and C. Scheier. *Understanding Intelligence.* MIT Press, Cambridge, MA, 1999.