A Bioinspired Hierarchical Reinforcement Learning Architecture for Modeling Learning of Multiple Skills with Continuous States and Actions

Daniele Caligiore **Marco Mirolli** **Domenico Parisi**
Gianluca Baldassarre
Laboratory of Computational Embodied Neuroscience, Istituto di Scienze e Tecnologie della Cognizione, Consiglio Nazionale delle Ricerche (LOCEN-ISTC-CNR), Via San Martino della Battaglia 44, I-00185 Roma, Italy,
{daniele.caligiore, marco.mirolli, domenico.parisi, gianluca.baldassarre}@istc.cnr.it

Abstract

Organisms, and especially primates, are able to learn several skills while avoiding catastrophic interference and enhancing generalisation. This paper proposes a novel hierarchical reinforcement learning (RL) architecture with a number of features that make it suitable to investigate such phenomena. The proposed system combines the mixture of experts architecture with the neural-network actor-critic architecture trained with the TD(λ) reinforcement learning algorithm. In particular, *responsibility signals* provided by two gating networks (one for the actor and one for the critic) are used both to weight the outputs of the respective multiple (*expert*) controllers and to modulate their learning. The system is tested with a simulated dynamic 2D robotic arm that autonomously learns to reach a target in (up to) three different conditions. The results show that the system is able to appropriately allocate experts to tasks on the basis of the differences and similarities among the required sensorimotor mappings.

1. Introduction

During development children acquire a large repertoire of skills by autonomously interacting with the environment. In particular, children learn to do many differnt things in many different contexts. Although social interactions are fundamental for human development, individual learning processes, in particular those based on trial and error, have at least a comparable importance.

The goal of the present research is to develop a bioinspired hierarchical and (softly) modular reinforcement learning system suitable for studying the autonomous acquisition of different skills through trial and error learning. In particular, we are looking for a system that may allow the study of the brain processes for which skills that require similar sensorimotor mappings are stored in the same neural structures, so to favor *generalization* and fast learning, whereas skills that require different sensorimotor mappings are stored in different neural structures, so to avoid *catastrophic interference*. In this way, we might be able to explain the neural mechanisms behind the sensorimotor processes proposed by Piaget (1953) to explain children development. In particular, the reuse of the same neural structures for two different skills might correspond to the Piagetian concept of *assimilation*, while the acquisition of new skills through the formation of new neural representations might be related to the concept of *accommodation*.

1.1 Biology of skill acquisition

Neuroscience suggests that *basal ganglia* are the principal brain structures that support the acquisition of multiple skills (Houk et al., 1995) as they seem to underly both trial-and-error learning processes and action selection. The striatum, which represents the basal ganglia input, is formed by two kinds of structures: the *matriosomes*, which are supposed to encode *actions* at various levels of abstraction, and the *striosomes*, which respond to rewards and cues that predict them. Moreover, the striosomes are connected to the areas (the substantia nigra pars compacta and the ventral tegmental area) that produce the *dopaminergic learning signals* that seem to modulate the plasticity of the synapses of both the matriosomes and the striosomes.

The basal ganglia have also a *hierarchical* structure that is based on (partially) *segregated loops* linked to different cortical areas. Different loops encode, for example, motor actions (e.g., the loops with

motor and *premotor cortex*), or context and goals (e.g., loops with *prefrontal cortex*). These loops seem to be characterized by a (soft) modularity, possibly encoding different actions and goals. Functionally, hierarchy and modularity might have the two important functions of (a) helping to avoid *catastrophic interference* and (b) enhancing *generalisation*, in particular the storing of different behaviours involving similar sensorimotor mappings in the same neural structures.

1.2 Constraints used to build the model

The system presented here was developed with the following constraints in mind: (a) using reinforcement learning (RL) algorithms (Sutton and Barto, 1998) as models of skill acquisition based on individual trial-and-error learning processes; (b) in particular, using the RL *actor-critic* architecture (Sutton and Barto, 1998) as a model of biological action learning in the basal ganglia; in particular, the *actor* is supposed to correspond to the matriosomes, the *critic* to the striosomes, and the *TD-error* learning signal to phasic dopamine (Houk et al., 1995); (c) using neural-networks (linear function appproximators) to ease the comparison with brain structures and processes; (d) developing a hierarchical system, in analogy with the basal ganglia, that may autonomously decide whether encoding skills in the same or in different neural structures on the basis of the similarities and differences between the required sensorimotor mappings; (e) developing a system that can control an embodied system (here a simulated robot) interacting with an environment with *continuous states* through *continuous actions*. Note that the simultaneous satesfaction of all these constraints makes the proposed system considerably novel (see sec. 1.3).

1.3 Related models

In the literature on neural networks the problem of how avoiding catastrophic interference and exploiting generalisation has been tackled with *mixture of experts* models (Jacobs et al., 1991). This model has a hierarchical modular architecture formed by a number of *experts*, which compete to learn the training patterns, and a *gating network*, which learns to decide when each expert should act and learn. This system is central for this work but is wholly based on supervised learning.

Within the RL framework, few models have been developed for working with continuous actions and states (e.g. Doya, 2000; Peters and Schaal, 2008) and have been shown to work within embodied systems. However, these systems are not hierarchical and have not been designed to acquire multiple skills.

Hierarchical RL systems are particularly well-suited for our purposes (see Barto and Mahadevan, 2003 for a review). These systems are capable of performing task-decomposition, usually on the basis of learning sub-tasks from a final goal. However, the vast majority of these systems assume discrete states and action spaces and have not been used in the continuum. Konidaris and Barto (2009) and Mugan and Kuipers (inpr) have proposed two hierarchical systems that build skills in continuous spaces. The first system is based on the idea of using the sets of states from which a skill can be accomplished (initiation sets) as the goal states of new skills to be acquired. The second system (QLAP) builds models of the environment and uses them for discretising continuos state variables and for learning actions that can reliably lead to certain effects. Although very interesting, these systems do not directly face the problem tackled here of how storing different skills in the same or different experts. Furthermore, they have non-neural aspects that might hinder their mapping to brain structures and processes.

Doya et al. (2002) have developed a Multiple Model-Based Reinforcement Learning system (MMRL) that can perform autonomous task decomposition in continuous state-action spaces. The model is based on several experts, each of which is formed by a controller and a forward model. Control is allocated to the experts on the basis of the performance of their forward models. Hence, in this system task decomposition is based on the dynamical characteristics of different parts of the sensorimotor space, and not on the capabilities of each modules to learn each skill.

Finally, Baldassarre (2002) proposed a modular RL system that combines the mixture of experts idea with the actor-critic RL, but was capable of dealing only with discrete actions. In this paper we present an evolution of this system that can deal with tasks requiring continuous actions. In particular, our system is tested in a task in which it controls a dynamic simulated robotic arm that learns to reach the handle of a cup with different orientations.

The rest of the paper is structured as follows: sec. 2 presents the simulated robot and environment; sec. 3 presents the model; sec. 4 presents the results of the tests; finally, sec. 5 draws the conclusions.

2. Setup

2.1 The Simulated Robot and the Task

Fig. 1 shows the simulated robot and environment. The simulated robot is formed by three components: a simulated RGB camera, a 3D arm-hand (a simulation of the *iCub* robot based on the 3D physics engine *Newton*: cf. Caligiore et al., 2008), and simplified simulated muscles.

The camera always fixates the target of reaching

(cup-handle) on the basis of a simple hardwired *fixation reflex* that makes the system look at the centroid of the pixels having the colour of the target (cf. Caligiore et al., 2008). The system controls only 2 DOFs of the arm working on the plane (one actuated joint is on the shoulder and the other in the elbow). This reflects the fact that, when learning to reach, children tend to use few degrees of freedom (Berthier et al., 2005, 1999). The hand is always kept straight open.

Each of the muscle models (one for each of the two controlled DOFs of the arm) is based on a *Proportional Derivative* controller (PD) that offers a simple way of simulating the spring-like and dumping properties of real muscles that are important for producing stable and smooth reaching movements. The PDs supply the torque to the arm joints in proportion to the difference between the current joint angles and the desired *equilibrium points* (EPs) generated by the model (i.e., the desired shoulder and elbow joint angles, see sec. 3.4). The torque applied to each joint is decreased inversely to the current rate of change (*derivative*) of the joint angle. As shown in Berthier et al. (2005), simple muscle models as these allow reproducing various aspects of real reaching movements.

The environment is a working plane with a simplified *cup* (achored to the table) whose handle can be at either the left, the centre, or the right with respect to the robot. The task requires that the arm learns to touch the handle by starting from random initial positions. When the hand touches the handle, the system gets a reward of one; if the hand touches the body of the cup the system gets a small punishment (-0.2); in all other cases, the system receives a zero reinforcement. Notice that the task is rather challenging for four reasons. First, to reach the handle the model has to generate variable EPs so that the *dynamic arm follows a curved trajectory* (cf. Caligiore et al., 2008, 2010). Second, different positions of the handle require very different sensorimotor mappings, thus making the overall problem for the controller highly unlinear. Third, learning is based exclusively on a very rare scalar value of reinforcement (Berthier et al., 2005). Fourth, the perception of the system (see Sect. 2.1) is very limited: in particular, the controller is informed only on the kinematics (joint angles) of the arm but not on the its dynamics (e.g., changes of joint angles and hand velocity).

3. Architecture and Algorithms

The system (fig. 2) is composed of two main components: an *actor* for controlling actions and a *critic* for evaluating states. Both the actor and the critic have a hierarchical architecture formed by a *gating network* and a number of *experts*, as in the mixture of experts model (Jacobs et al., 1991). With respect to the mixture of experts model the functioning and learning algorithms of all components have been modified for working with a continuous RL system (cf. Baldassarre, 2002).

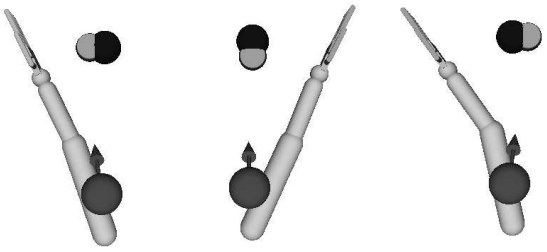


Figure 1: The robotic set-upin the three conditions: handle on the left, centre, and right. The black sphere with the grey appendix represents the body of the cup with its handle. The larger sphere with the arrow represents the eye gazing towards the centre of the handle.

The system gets two types of inputs: (a) the gaze direction of the camera, which indicates the position of the target, and (b) the combined information about the arm posture and the hand-target spatial relationship. Both sources of information are encoded in neural maps with *population codes* (Pouget and Latham, 2003): the closer the current posture to the *preferred posture* of a neural unit, the higher the unit activation. In particular, the camera pan and tilt angles are encoded in a 2D *eye-posture map* formed by 21×21 neural units. This map is activated on the basis of a Gaussian function (height = 1; width = distance between two neighbouring units in the map) centred on the current posture. The arm-posture/hand-target-distance information is encoded in a 3D *arm-posture map*. First, the arm posture is encoded in each of five 21×21-unit maps as done for the eye-posture map. Then, the activation of all units of four of these maps is scaled on the basis of the distance (passed through a Gaussian function) of the hand from the target with respect to a particular direction (i.e, each map is maximally activated when the hand-target distance is maximal, respectively, towards east, north, west, and south). The last of the five maps is maximally activated, again on the basis of a Gaussian function, when the hand-target distance is zero.

Importantly, the information on gaze direction, which is related to the task to be accomplished (where to reach), constitutes the input of the gating networks, whereas the combined information on arm posture and hand-target relation, which is required for producing appropriate control signals, constitutes the input of the experts. The difference in the input that the gating networks and the experts receive depends on the different functions that they play in the

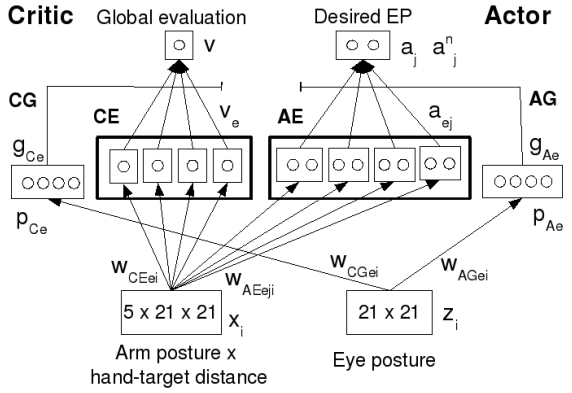


Figure 2: The architecture of the model.

hierarchical system: the gating networks have to decide which expert to use depending on the task, while the experts have to implement the sensorimotor mappings that produce appropriate behaviors. This arrangement seems also to reflect the organization of the basal ganglia, where high-level loops receive perceptual information useful for selecting overall goals (e.g., internal states, object identity), whereas low-level loops receive proprioceptive and visual information that are useful to control action (e.g., information on object shape).

3.1 Functioning of the system

Both the actor and the critic are formed by a gating network and four experts.

Actor gating network. The actor gating network (AG) has four output units (indexed with e) which receive input from the eye-posture map units z_i (see below) via connections with weights w_{AGei}. The output units are ranked in a decreasing order on the basis of their activation potential p_{Ae}, and their activation g_{Ae}, representing the expert *prior responsibility*, is calculated as a function of the resulting ranks k_e ($k_e = 0, 1, 2, 3$):

$$g_{Ae} = b^{-k_e} / \sum_{e=1}^{4} b^{-k_e} \tag{1}$$

where b is a constant that determines the differences in the degree with which each expert contributes to the global action depending on their rank (in the experiments presented below b has been set to 6, so that $g_{Ae} = 0.834, 0.139, 0.023, 0.004$). Differently from the mixture-of-experts way of computing the prior responsibilities, based on a soft-max function (Jacobs et al., 1991), the use of the ranks guarantees that the responsibility of all the experts is always different from zero, even after prolonged training. This implies that although one expert will tend become maximally specialized in encoding one skill, some other experts might learn in "background" the same skill as their responsibility is different from zero. Preliminary tests, not reported here, show that this allows a "latent duplication" of skills that can facilitate the development of new skills from previously acquired ones (a phenomenon which might be related to Piagetian accommodation). This aspect of the model, not further discussed here, will be investigated in future work.

Actor experts. Each actor expert (AE_e) has two output units with sigmoidal activation a_{ei} which encode the control signals to the arm (the two desired joint angles). These output units receive input from the arm-posture map units x_i (see below) via connections with weights w_{AEeji}. The global action a_j (desired EPs) of the actor is computed on the basis of the prior responsibilities of the experts g_{Ae}:

$$a_j = \sum_e g_{Ae} \cdot a_{ej} \tag{2}$$

Critic gating network. The critic gating network (CG) works analogously to the AG, on the basis of the connection weights w_{CGei}, the unit activation potentials p_{Ce}, and the *prior responsibilities* of the critic experts g_{Ce}.

Critic experts. Each critic expert (CE) has a linear output unit v_e encoding the evaluation of the current state and receives input from the arm-posture map units x_i via connections with weights w_{CEei}. The global evaluation v of the critic is computed on the basis of the prior responsibilities of the experts g_{Ce}:

$$v = \sum_e g_{Ce} \cdot v_e \tag{3}$$

3.2 Learning signals

Global TD-error. Couples of successive global evaluations, together with the reward signal r_t, are used to compute the global TD-error (or *surprise*) s_t for reinforcement learning (Sutton and Barto, 1998):

$$s_t = \begin{cases} r_t - v_{t-1} & if\ end\ of\ trial \\ (r_t + \gamma v_t) - v_{t-1} & if\ during\ trial \\ 0 & if\ start\ of\ trial \end{cases} \tag{4}$$

where γ is a discount factor (here $\gamma = 0.99$).

Experts TD-error. The expert TD-error (surprise) signals are calculated as follows:

$$s_{et} = \begin{cases} r_t - v_{et-1} & if\ end\ of\ trial \\ (r_t + \gamma v_{et}) - v_{et-1} & if\ during\ trial \\ 0 & if\ start\ of\ trial \end{cases} \tag{5}$$

In the brain, the error signals s_t and s_{et} might correspond to phasic dopaminergic signals.

Actor experts posterior responsibilities. To train the actor experts and gating network the algorithm computes the *posterior responsibilities* of the

actor experts as follows:

$$h_{Ae} = \frac{c_{Ae} \cdot g_{Ae}}{\sum_e [c_{Ae} \cdot g_{Ae}]} \quad (6)$$

where c_{Ae} is a measure of the *correctness* of the actor expert e, defined as:

$$c_{Ae} = e^{-0.5\left(D\left[\mathbf{a}^n_{t-1}, \mathbf{a}_{et-1}\right]\right)^2} \quad (7)$$

where $D\left[\mathbf{a}^n_{t-1}, \mathbf{a}_{et-1}\right]$ is the Euclidian distance between the two vectors encoding respectively the global action $\mathbf{a}^n_{t-1}$ issued to the muscles model (sec. 3.4) and the action $\mathbf{a}_{et-1}$ computed by expert e.

Note that eq. 7 actually measures the *similarity* of the expert action with the whole-system action, and so favours a non-guided specialisation of the actor experts. Indeed, a true measure of *correctness* should take into consideration not only such similarity but also the quality of the whole-system action based on the surprise, for example as follows: $c_{Ae} = e^{-0.5 \cdot s_t \cdot \left(D\left[\mathbf{a}^n_{t-1}, \mathbf{a}_{et-1}\right]\right)^2}$. An alternative solution would be to use the surprise to modify eq. 10 below as follows: $\Delta w_{AGei} = \eta_{AG} \cdot s_t \cdot (h_{Ae} - g_{Ae}) \cdot z_{it-1}$.

Critic experts posterior responsibilities. The *posterior responsibilities* of the critic experts are computed as follows:

$$h_{Ce} = \frac{c_{Ce} \cdot g_{Ce}}{\sum_e [c_{Ce} \cdot g_{Ce}]} \quad (8)$$

where c_{Ce} is a measure of the *correctness* of the critic expert e defined as:

$$c_{Ce} = e^{-0.5(s_{et})^2} \quad (9)$$

3.3 Learning

Actor gating network learning. The learning of the AG network has been developed in analogy with the mixture of experts model. Intuitively, the learning rule tends to increase the responsibility of an expert if its correctness (i.e., its similarity to the executed action) is higher than average and to decrease it otherwise. Formally, the weights of the AG network are updated as follows:

$$\Delta w_{AGei} = \eta_{AG} \cdot (h_{Ae} - g_{Ae}) \cdot z_{it-1} \quad (10)$$

where η_{AG} is the learning rate (here set to 3.0).

Actor experts learning. The weights of the actor experts are trained on the basis of a TD(λ) learning rule with *replacing eligibility traces* applied to linear function approximators (Sutton and Barto, 1998). In particular, at time t and for the expert e the eligibility trace e_{AEejit} of a connection weight w_{AEeji} is computed. If this eligibility is smaller than the "decayed" old eligibility $e_{AEejit-1}$, the latter is used instead of the former to train the weights:

$$\begin{aligned} & e_{AEejit} = \gamma \cdot \lambda \cdot e_{AEejit-1} \\ & e^b = h_{Ae} \cdot (a^n_{jt} - a_{ejt}) \cdot \dot{a}_{ejt} \cdot x_{it} \\ & if \quad |e_{AEejit}| < \left|e^b\right| \quad then \quad e_{AEejit} = e^b \\ & w_{AEjit} = w_{AEjit-1} + \eta_{AE} \cdot s_t \cdot e_{Ajit-1} \end{aligned} \quad (11)$$

where e^b is a buffer variable, η_{AE} is a learning rate (set to 0.9), and $\dot{a}_{ejt} = a_{ejt}(1 - a_{ejt})$ is the Sigmoid derivative. The rationale of this formula is as follows. By default, the new eligibility e_{AEejit} is equal to the old discounted ($\gamma = 0.99$) and decayed ($\lambda = 0.94$) eligibility $e_{AEejit-1}$ (cf. Sutton and Barto, 1998). Then the potential new eligibility (stored in e^b) is computed and becomes the new actual eligibility if it is higher, in absolute value, than the decayed old eligibility. In either case, the resulting eligibility is used to update the weights (in particular the *previous* eligibility e_{Ajit-1} is used to this purpose together with the global surprise s_t). Importantly, e^b is computed on the basis of the input x_{it}, the expert posterior responsibility h_{Aet} (the update is stronger if the responsibility is higher), and the difference between the executed action a^n_{jt} and the expert output a_{ejt} (the expert action is moved towards the executed action if $s_t > 0$, and away from it if $s_t < 0$).

Critic gating network learning. The weights of the critic gating network are updated as follows:

$$\Delta w_{CGei} = \eta_{CG} \cdot (h_{Ce} - g_{Ce}) \cdot z_{it-1} \quad (12)$$

where η_{CG} is a learning rate (here set to 0.5). Again, the rule has been developed in analogy with the mixture of experts model: the responsibility of an expert is increased if its correctness was higher (i.e., its reward prediction error was smaller) than average, and decreased otherwise.

Critic experts learning. As for the actor, the weights of the critic experts are trained on the basis of replacing eligibility traces. In particular, for expert e the eligibility trace at time t, e_{CEeit}, of a connection weight w_{CEei} is computed on the basis of the input x_{it} and the expert responsibility h_{Cet}. If this eligibility is smaller than the decayed old eligibility $e_{CEeit-1}$ the latter is used instead of the former to train the weight:

$$\begin{aligned} & e_{CEeit} = max\left[\gamma \cdot \lambda \cdot e_{CEeit-1}, h_{Ce} x_{it}\right] \\ & w_{CEeit} = w_{CEeit-1} + \eta_{CE} \cdot s_{et} \cdot e_{CEeit-1} \end{aligned} \quad (13)$$

where λ is the decay coefficient of the eligibility trace (here $\lambda = 0.94$), and η_{CE} is the learning rate (here $\eta_{CE} = 0.06$). Note how, contrary to what done for the actor, the comparison between the old decayed eligibility and the new potential eligibility can be done without considering their absolute values as both values are positive: the sign of change of the weight is given only by surprise s_{et}. Also note that, contrary

to the actor experts, the expert surprise s_{et}, and not the global surprise s_t, is used to update the critic expert weights.

Notice that the learning rates of gates (η_{AG} and η_{CG}) are higher than those of the respective experts (η_{AE} and η_{CE}) as this was found to ease the specialisation of experts (cf. Baldassarre, 2002). Moreover, the learning rate of the actor experts is higher than that of the critic experts as the actor experts have sigmoidal output units (implying a derivative ≤ 0.25 in the learning rule of eq. 11), whereas the critic experts have linear output unit (implying a derivative = 1: eq. 13). The learning rate of the AG is larger than that of the CG as the difference between the prior and postirior responsibilities of the former tend to be smaller than that of the latter (cf. eq. 10 and eq. 12).

3.4 Noise generator

To foster exploration, noise must be added to the (continuous) actions produced by the actor. The techniques used for noise generation when the control is based on torque (e.g., see Doya, 2000) are not well suited when the control is based on desired equilibrium points, as it happens in our system. The reason is that if noise is added to the *desired posture* the system tends to explore only the space around it. To solve this problem, we use a method where noise is generated with respect to the *current posture* $\mathbf{J}_t$ of the arm, so this can "be pulled progressively away" and explore the whole space. Furthermore, the noise has an inertia as the arm inertia would average out a white noise.

The method works as follows (for more details and a study of the effects of this noise, see Caligiore et al., 2010). First, we generate a inertial noise vector $\mathbf{N}_t^b$ on the basis of a two-dimensional random vector $\mathbf{N}^{rand}$ (whose elements are uniformly drawn in $[-1, +1]$):

$$\mathbf{N}_t^b = (1 - \sigma) \cdot \mathbf{N}_{t-1}^n + \sigma \cdot \mathbf{N}^{rand} \qquad (14)$$

where σ ($\sigma = 0.05$) is a time constant determining the inertia of noise. Then, we re-scale the noise vector so that its maximum lenght is m ($m = 10$):

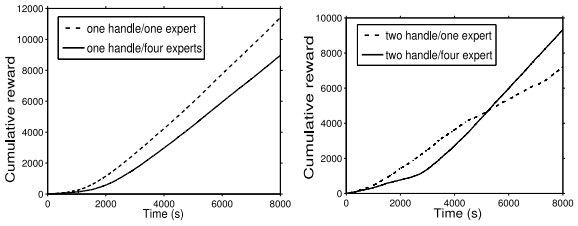
$$\mathbf{N}_t^n = \begin{cases} \mathbf{N}_t^b / \left|\left|\mathbf{N}_t^b\right|\right| & if \quad 1 < \left|\left|\mathbf{N}_t^b\right|\right| \\ \mathbf{N}_t^b & otherwise \end{cases} \qquad (15)$$

$$\mathbf{N}_t = m \cdot \mathbf{N}_t^n \qquad (16)$$

where $\left|\left|\mathbf{N}_t^b\right|\right|$ is the length of the non-scaled noise vector. Finally, we average the noise vector with the equilibrium point vector produced by the actor ($\mathbf{EP}_t$), *expressed with respect to the reference frame centred on the current joint angles* $\boldsymbol{J}_t$ and denoted with EP_t^r), and then we obtain the final motor command $\mathbf{EP}_t^n$ actually issued to the muscle models:

$$\begin{aligned} \mathbf{EP}_t^r &= \mathbf{EP}_t - \mathbf{J}_t \\ \mathbf{EP}_t^{nr} &= (1 - A) \cdot \mathbf{EP}_t^r + A \cdot \mathbf{N}_t \\ \mathbf{EP}_t^n &= \mathbf{EP}_t^{nr} + \mathbf{J}_t \end{aligned} \qquad (17)$$

where A is a parameter, linearly decreased during training from 0.9 to 0.1, which weights the relative importance of noise over the action signal (i.e., which controls the exploration-exploitation ratio).



(a) (b)

Figure 3: Cumulative reward of the one-expert and four-expert systems in the experiments with one (a) and two (b) handle positions.

4. Results

The performance of the hierarchical system with four experts was compared with a non-hierarchical system composed of a single expert. The two systems were compared in three experiments requiring to reach a cup-handle in various conditions: (a) the handle is always positioned on the left; (b) the handle can be either on the left or on the right; (c) the handle can be in one of three positions: on the left, on the right, or in the centre. The last experiment was run only with the four-expert system as the one-expert system could not tackle this task (as it already failed in the easier task with two handles, see below).

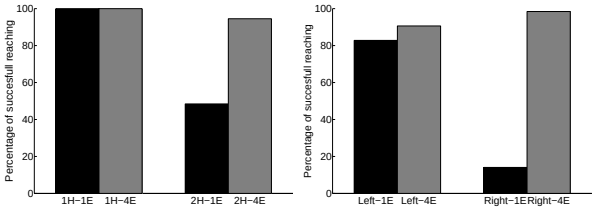
Fig. 3 shows the cumulative reinforcement of the two systems during training in the one-handle and two handles conditions; Table 1 and Fig. 4a show the performance (percentage of successful reaches) in the same two conditions during post-training tests of 64 trials in which the initial position of the arm was set on one of the 8×8 vertexes of a regular grid in the joint space. Different replications of the experiments produced qualitatively similar results.

4.1 Experiment with one handle

Fig. 3 shows that the one-expert system learns faster than the four-expert one. This is due to the fact that, to solve the task, the one-expert system does not need to train the gating networks aside the experts. After some time, however, the four-expert systems catches up, and its performance becomes similar to the one of the one-expert system, as confirmed by the tests reported in Fig. 4a and in Table 1.

Table 1: One-expert and four-expert systems: performance with one, two, and three handles.

Experiment	*1 exp.*	*4 exp.*
One condition, right	100%	100%
Two conditions, right	14.06%	98.44%
Two conditions, left	82.81%	90.62%
Three conditions, right		93.75%
Three conditions, left		90.62%
Three conditions, centre		98.43%



(a) (b)

Figure 4: Test performance of the one-expert (black) and the four experts (gray) systems. (a) Experiments with one (left bars) and two handles (right bars). (b) Experiment with two handles: performance when the handle is on the left (left bars) and when it is on the right (right bars).

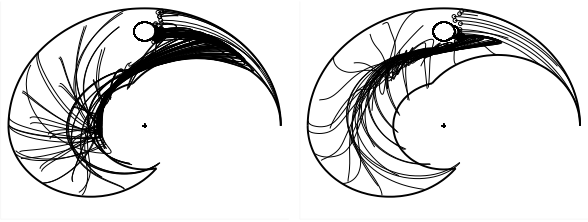
4.2 Experiment with two handles: encoding of skills in different experts

Fig. 3 shows that in the two-handles condition, after an initial transient, the four-expert system starts to outperform the one-expert system. This is even more clearly indicated by the results of the post-training tests (Table 1 and Fig. 4).

Indeed, the one-expert system managed to consistently reach the handle only when this was on the left (performance: 82.81%), while it reached a very poor performance when the handle was on the right (14.06%). This clearly shows the limitations of a standard single reinforcement learning system when dealing with more than one task. On the contrary, the four-expert system succeeds in solving both tasks through the exploitation of *two different experts*, one for each task, for both the critic and the actor. This indicates that the hierarchical system is both capable of discriminating the two tasks at the level of the gating networks and to acquire different skills with different experts. Fig. 5 shows some examples of the trajectories exhibited by the two systems.

4.3 Experiment with three handles: encoding of skills in the same expert

In the experiment with three handle positions, both the actor and the critic of the four-expert system learn to use the same expert for both the left and the central handle while using a different expert for the right handle. The reason of this is that the system discovered that the same movements can be used to reach the handle both when it is on the left and when it is in the centre.



(a) (b)

Figure 5: Hand trajectories of the one-expert (a) and four-expert systems (b) during tests after training in the one-handle condition. See text for details.

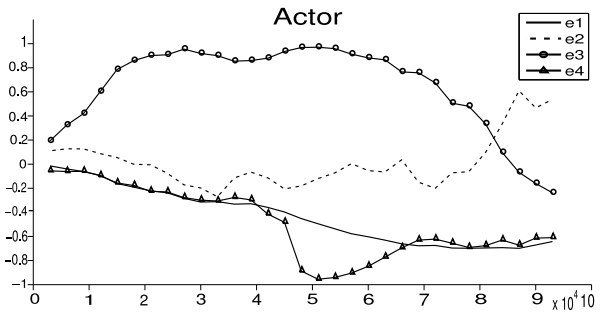
Fig. 6 shows that at the beginning of the experiment with three handle positions both the critic and actor start to use different experts for the three handles. With the progression of learning, however, the actor starts to use the expert allocated for the central handle (second expert) also for reaching the left handle. The critic does exactly the opposite: after a while it starts using the expert allocated for the left handle also when the handle is in the centre. This shows that both the actor and the critic gating networks can learn to allocate experts on the basis of the similarity of the sensorimotor mappings required for solving different tasks.

5. Conclusion

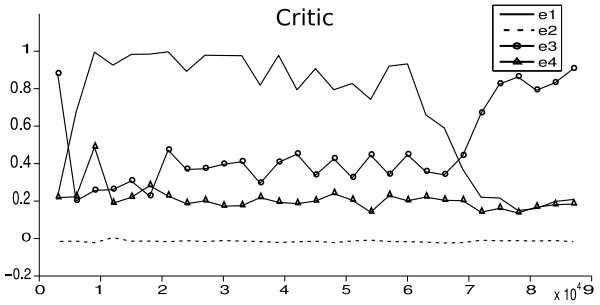
This article presented a hierarchical modular reinforcement leaning system that is able to acquire different skills by using different expert controllers on the basis of (a) the different sensorimotor mappings required by the skills and (b) the computational capability of the experts. The tests showed that the system can autonomously learn to use the same expert for skills that require similar sensorimotor mappings and different experts for skills that require different mappings. Furthermore, thanks to the fact that the system is hierarchical, is based on the neural biologically-plausible actor-critic model, and can work with continuous actions and states, it seems to be particularly well suited for studing developmental processes (e.g., see Berthier et al., 2005). In particular, the results presented here provide preliminary indications that the system can be profitably used to investigate the assimilation/accommodation processes proposed by Piaget.

Acknowledgements

This research was supported by the EU Projects *ROSSI* (contract no. FP7-STREP-216125) and *IM-*



(a)



(b)

Figure 6: Text with three handles, four-expert system: (a) Moving average (1000 steps window) of the activation of the actor gating network output units during learning when the system pursues the left handle. (b) Same data for the critic when the system pursues the central handle.

CLeVeR (contract no. FP7-IP-231722). We thank Anna Borghi for contributing to the initial ideas on the hierarchical aspects of the model.

References

Baldassarre, G. (2002). A modular neural-network model of the basal ganglia's role in learning and selecting motor behaviours. *Journal of Cognitive Systems Research*, 3:5–13.

Barto, A. G. and Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 13(4):341–379.

Berthier, N. E., Clifton, R. K., McCall, D. D., and Robin, D. J. (1999). Proximodistal structure of early reaching in human infants. *Exp Brain Res*, 127:259–269.

Berthier, N. E., Rosenstein, M. T., and Barto, A. G. (2005). Approximate optimal control as a model for motor learning. *Psychol Rev*, 112:329–346.

Caligiore, D., Ferrauto, T., Parisi, D., Accornero, N., Capozza, M., and Baldassarre, G. (2008). Using motor babbling and hebb rules for modeling the development of reaching with obstacles and grasping. In Dillmann, R., Maloney, C., Sandini, G., Asfour, T., Cheng, G., Metta, G., and Ude, A., (Eds.), *Proc. of COGSYS 2008*, Karlsruhe, Germany. Springer.

Caligiore, D., Guglielmelli, E., Parisi, D., and Baldassarre, G. (2010). A reinforcement learning model of reaching integrating kinematic and dynamic control in a simulated arm robot. In *IEEE International Conference on Development and Learning (ICDL2010)*, pages 211–218. IEEE, Piscataway, NJ.

Doya, K. (2000). Reinforcement learning in continuous time and space. *Neural Comput*, 12(1):219–245.

Doya, K., Samejima, K., Katagiri, K.-i., and Kawato, M. (2002). Multiple model-based reinforcement learning. *Neural Computation*, 14(6):1347–1369.

Houk, J. C., Davis, J., and Beiser, D., (Eds.) (1995). *Models of Information Processing in the Basal Ganglia*. The MIT Press, Cambridge, MA.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, 3:79–87.

Konidaris, G. D. and Barto, A. G. (2009). Skill discovery in continuous reinforcement learning domains using skill chaining. In Bengio, Y. e. a., (Ed.), *Advances in Neural Information Processing Systems 22 (NIPS09)*, pages 1015–1023.

Mugan, J. and Kuipers, B. (inpr). Autonomous exploration and the qualitative learner of action and perception, qlap. *IEEE Transactions on Autonomous Mental Development*.

Peters, J. and Schaal, S. (2008). Natural actor-critic. *Neurocomputing*, 71:1180–1190.

Piaget, J. (1953). *The Origins of Intelligence in Children*. Routledge and Kegan Paul, London.

Pouget, A. and Latham, P. E. (2003). Population codes. In Arbib, M. A., (Ed.), *The Handbook of Brain Theory and Neural Networks*. The MIT Press, Cambridge, MA, USA, second edition.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge MA, USA.