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 reinforcement learning (RL) architecture which has a number of features that make it suitable to investigate these phenomena. The model instantiates a mixture of expert architecture within a neural-network actor-critic system trained with the TD(λ) RL algorithm. The "responsibility signals" provided by the gating network are used both to weight the outputs of the multiple "expert" controllers and to modulate their learning. The model is tested in a simulated dynamic 2D robotic arm which autonomously learns to reach a target in (up to) three different conditions. The results show that the model is able to train same or different experts to solve the task(s) in the various conditions depending on the similarity of the sensorimotor mappings they require.

1. Introduction

During development children acquire a complex repertoire of skills by interacting with the environment. In particular, they are capable of learning to do many related things and to execute them in various contexts. Although social interactions are fundamental for human development, individual processes have at least a comparable importance, in particular those based on trial-and-error learning.

This research had the goal to develop a bioinspired hierarchical and (softly) modular reinforcement-learning model useful to study these individual processes. In particular, the model allows to study the brain processes for which when an organism learns many different skills, it can store the related information in the same neural structures when the sensorimotor mappings related to the skills to be learned are similar, so to enhance generalisation and fast learning, and in different brain structures when they are substantially different, so to avoid catastrophic interference. These mechanisms might for example be the neural correlate behind the assimilation and accommodation processes proposed by Piaget (1953) to explain children development. In particular, *assimilation*, which implies that a goal is accomplished on the basis of previously acquired skills, might involve the reuse of the same neural structures, whereas *accommodation*, which implies that a goal is pursued by developing a new skill, might involve the formation of rather new neural representations.

1.1 The biology of skill acquisition

Neuroscience suggests that *basal ganglia* are among the main brain systems underlying the acquisition of multiple skills (Houk et al., 1995). They seem to underly trial-and-error learning processes and action selection (also in an exploratory/random fashion when the system faces a new situation). They are formed by a first type of structure, the *matriosomes*, which might encode *actions* at various levels of abstraction, and a second type of structure, the *striosomes*, capable of responding to rewards and cues predicting them Striosomes are connected to areas (e.g., the *substantia nigra pars compacta*) responsible for producing learning signals based on the neuromodulator *dopamine*, leading to update the synapses of matriosomes and striosomes themselves.

The basal ganglia also have a *hierarchical* structure based on (partially) segregated loops linked to different cortical areas. These loops encode, for example, motor actions (e.g., the loops with *motor* and *premotor cortex*), or contest and goals (e.g., loops with *prefrontal cortex*). Loops seem to be characterized by a (soft) modularity, possibly encoding differ-

ent 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 The constraints used to build the model

Given the aforementioned goal of this research, the model presented here was developed with these constraints in mind: (a) using RL (Sutton and Barto, 1998), and not supervised learning, as our goal is to study skill acquisition based on individual trial-and-error learning processes; (b) being capable of autonomously deciding if encoding skills in the same or different neural structures depending on their similarities; (c) using neural-networks (linear function appproximation) to ease finding relations with brain structures and processes; (d) using RL *actor-critic* models (Sutton and Barto, 1998) as these are among the most biologically plausibility RL models; in particular, the *actor* component of the model plays a function similar to the basal ganglia matriosomes, the *critic* plays a function similar to the basal ganglia striosomes, and the *TD-error* learning signal has a dynamics similar to that of phasic dopamine during learning (Houk et al., 1995); (e) having a hierarchical macro-architecture, similarly to basal ganglia, capable of suitably deciding which part of the system should encode which skills based on their similarities/differences; (f) being capable of functioning within an embodied system (here a simulated robot) interacting with a world with *continuous states* through *continuous actions*. Note that, given the constraints "a-d", the features "e" and "f" render the model rather 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, some models have been developed to work 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 subtasks from a "final" task. However, most of them assume discrete states and action spaces. Konidaris and Barto (2009) and Mugan and Kuipers (inpr) have proposed two hierarchical systems to build options in continuous spaces. The first system is based on the idea of forming new skills in chain on the basis of the "initiation set" (set of states from which a skill can be successfully accomplished) of other skills. The second (QLAP) learns models and uses them to learn to discretise states represented by continuous variables, and to build actions which 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 expert/options. Furthermore, they have some non-neural aspects that might reveal difficult to be mapped to brain processes.

Doya et al. (2002) have developed a Multiple Model-Based Reinforcement Learning system (MMRL) capable of performing autonomous task decomposition in continuous state-action spaces. The model is based on several experts each formed by a controller and a forward model. This system allows performing task decomposition when the non-observability of the world can be disambiguated only by acting on it (e.g., lifting a new object can reveal its weight). Although very interesting, the system performs task decomposition based on the dynamical characteristics of the sensorimotor mapping space, and not on the capability of each module to learn or not certain skills.

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 which can tackle tasks requiring continuous actions: in particular it can control a dynamic simulated robotic arm that learns to reach the handle of a cup with different orientations.

In the rest of the paper, sec. 2 presents the simulated robot and environment, sec. 3 presents the model, sec. 4 presents the results of the tests,and 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 *fix-*

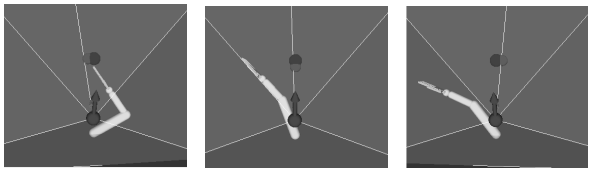


Figure 1: The robotic setup and the three conditions: handle on the left of the cup, at the centre facing the robot and, and on the right of the cup.

ation reflex focusing on the barycenter of the pixels having the colour of the target (Caligiore et al., 2008). The model controls only 2DOFs of the arm working on the plane. This reflects the fact that children use few degrees of freedom when learning to reach (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) which offers a simple way of simulating the spring-like and dumping properties of real muscles and of producing stable reaching movements. The PDs supply the torque to the arm joints in *proportion* to the difference between the arm desired *equilibrium points* (EPs) generated by the model (i.e., the desired shoulder and elbow joint angles, see sec. 3.4), and the current joint angles. 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", solidly anchored to it, having a handle at either the left, centre, or right position with respect to the robot. The task requires that the arm learns to touch the cup handle with the hand starting to move from random initial positions. When this happens, the system gets a reward of one. If the hand touches parts of the cup different from the handle it receives a small punishment (-0.2). In all other cases it receives a zero reinforcement. Notice that the tasks is rather challenging for four reasons. First, to reach the cup handle the model has to generate variable EPs so that the arm follows a *curved trajectory with a dynamic plant* (cf. Caligiore et al., 2008). Second, the target changes position in space renders the sensorimotor mapping highly unlinear. Third, the controller has to learn on the basis of the rare scalar value of reinforcement (Berthier et al., 2005). Last, the perception of the system (see Sect. 2.1) is rather limited and the controller is informed only on the kinematics (joint angles) but not on the arm dynamics (changes of joint angles, hand velocity, etc.).

3. Model Architecture and Algorithms

The architecture of the model is built on the following ideas (see fig. 2). First, it is based on two components, an *actor* for controlling action and a *critic* for evaluating actions. Second, each of the actor and critic has a hierarchical architecture formed by one *gating network* and number of *experts*, following the idea of the mixture of experts model (Jacobs et al., 1991). The functioning of the gating networks and the critic experts is as in the mixture of expert model. The functioning of the actor experts has been modified to implement actions within continuous RL. Last, the learning algorithms of all components are novel and have been modified to implement continuous RL (cf. Baldassarre (2002)).

The system gets two types of inputs: (a) the gaze direction of the camera, which indicates the position of the target (the image of the camera is only used to guide the hardwired fixation reflex, see sec. 2.1); (b) the combined information about the arm posture and the hand-target distance. These two sources of information are encoded in neural maps on the basis of the *population code* hypothesis Pouget and Latham (2003) for which the closer the posture (angles) to the *preferred posture* of a neural unit, the higher its 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 (maximum value equal to 1, width equal to the distance between two close units in the map) centred on the angles to encode. The arm-posture/hand-target-distance information is encoded in a 3D *arm-posture map*. First, the arm posture (angles) 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 towards a particular direction (i.e, each map is maximally activated when the hand-target distance is maximally towards east, or north, or west, or 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 different information sent to the gating networks and to the experts reflects the the fact that the gating networks should make high-level decisions on overall goals, whereas the experts should implement the detail actions to pursue them (cf. also Jacobs et al. (1991)). This also reflects basal ganglia organisation where high-level loops receive perceptual information useful for selecting overall goals (e.g., internal states, object identity), whereas low-level loops receive visual information useful to control action (e.g., object shape) and proprioception.

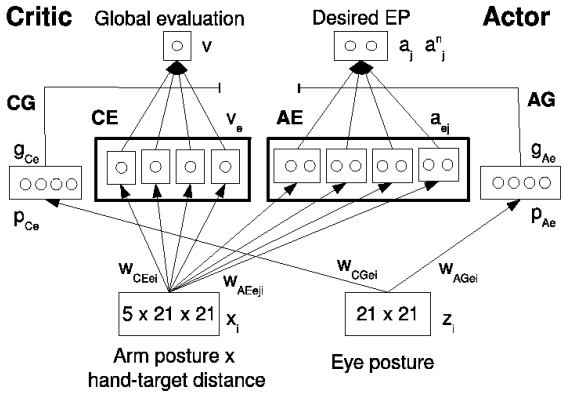


Figure 2: The architecture of the model.

3.1 Functioning of actor and critic

The actor is formed by a gating network and four experts.

Actor gating network This network (AG) has four output units indexed with e with activation potential p_{Ae}. These units receive input from the eye-posture map units z_i (see below) via connections with weights w_{AGei}. The activation g_{Ae} of the units encodes the *prior responsibility* of the actor experts and is computed on the basis of a many-winner competition (see below). To this purpose, the units are ranked in a decreasing order based on p_{Ae} and then they are activated on the basis of the resulting ranks k_e ($k_e = 0, 1, 2, 3$) as follows:

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

where e ($e = 1, 2, 3, 4$) is the number of experts and b is a coefficient used to set the propability that each expert contributes to the global action computation (b is set to 6, so $g_{Ae} = 0.834, 0.139, 0.023, 0.004$). Differently from the mixture-of-experts way of computing the prior responsibilities, based on a softmax 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 might become maximally specialized in encoding a skill, some other expert could learn in "background" the same skill as its responsibility is different from zero. Preliminary tests, not reported here, show that this allows a "latent duplication" of skills which could allow to develop new skills starting from previous ones (this might be the neural correlate of Piagetian accomodation). This aspect of the model, not further discussed here, will be investigated in future work.

Actor experts Each actor expert (AE) encodes actions (the two arm angles) with two output units j having Sigmoidal activation a_{ej}. These receive input signals 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}:

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

Also the critic is formed by a gating network and four experts.

Critic gating network This network (CG) works as the one of the actor on the basis of the connection weights w_{CGei}, the unit activation potential p_{Ce}, and the *prior responsibility* 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

Critic 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):

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

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

Experts TD-error The expert TD-error (surprise) signals are instead:

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

In the brain, the error signals s_t and s_{et} might correspond to 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:

$$h_{Ae} = \frac{c_{Ae} \cdot g_{Ae}}{\sum_e [c_{Ae} \cdot g_{Ae}]} \tag{6}$$

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

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

where $D\left[\mathbf{a}^n_{t-1}, \mathbf{a}_{et-1}\right]$ is the Euclidian distance between the two vectors encoding respectively the past

action of the actor expert $\mathbf{a}_{et-1}$ and the global actually-executed action $\mathbf{a}^n_{t-1}$ affected by noise (issued to muscles, see sec. 3.4).

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 of actor and critic

Actor gating network learning The weights of the actor gating 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 a learning rate set to 3.0. Technically the rule follows the ideas proposed by the mixture of experts model to update the gating network output (i.e., the experts' responsibilities). Intuitively, here the rule implies that the responsibility of an expert is increased if its correctness was higher (i.e., its action closer to the overall actually-executed noisy action) than other experts.

Actor experts learning The weights of the actor experts are trained on the basis of a TD(λ) learning rule with *replace 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 weight:

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

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 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 signal x_{it} affecting the weight to which the eligibility refers to, the expert posterior responsibility h_{Aet} (this implies that the update is stronger if this is higher), and the difference between the global noisy answer a^n_{jt} and the expert output a_{ejt} (this implies that the expert action is moved towards the noisy executed action, if $s_t > 0$, or 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 set to 0.5. Again, the rule follow the ideas proposed by the mixture of experts model. Intuitively, here the rule implies that the responsibility of an expert is increased if its correctness was higher (i.e., its future-reward prediction error smaller) than other experts.

Critic experts learning The weights of the critic experts are also trained on the basis of "replace eligibility traces". In particular, at time t and for the expert e the eligibility trace e_{CEeit} of a connection weight w_{CEei} is computed on the basis of the signal x_{it} and the expert responsibility h_{et}. Similarly to what done for the actor, 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:

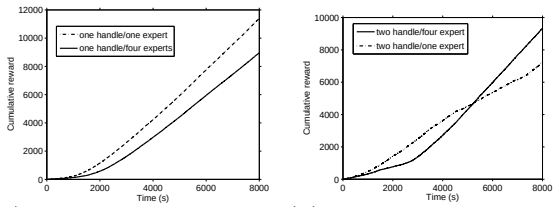
$$\begin{aligned} & e_{CEeit} = max\left[\gamma \cdot \lambda \cdot e_{CEeit-1}, h_{Ce} x_{it}\right] \\ & w_{CEeit} = w_{CEeit-1} + \eta_{CE} \cdot s_{et} \cdot e_{CEeit-1} \end{aligned} \quad (13)$$

where λ is the decay coefficient of the eligibility ($\lambda = 0.94$), and η_{CE} is a learning rate ($\eta = 0.06$). 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: indeed, the sign of change of the weight is given 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}) were set to values smaller than those of the learning rates 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 related to the actor experts is higher than the one related the critic experts as the actor experts have sigmoid 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 in the learning rule of eq. 13). The learning rate related to actor gating network is larger than the one related the critic gating network as the former tends to have a difference between the posterior and prior responsibilities much smaller than the latter (cf. eq. 10 and eq. 12).

Table 1: Performance of the one-expert and four-expert systems in the three experiments with one, two, and three handle positions.

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 3: (a) Cumulated reward of the one-expert and four-expert models for the experiment with one handle position. (b) Cumulated reward of the one-expert and four-expert models for the experiment with two handle positions.

3.4 Noise generator

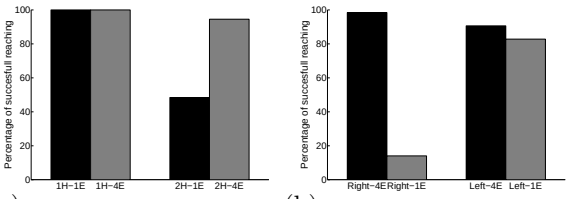
To foster exploration, we used a technique of noise generation which is more easily tunable than the method proposed in Doya (2000) for continuous reinforcement learning, and allows taking into account the fact that the inertia of the arm tends to average out white noise (see Caligiore et al., inpr for more details on this technique; cf. Peters and Schaal, 2008 for alternative methods).

In what follows, the global motor command produced by the actor $\mathbf{a}$, is mapped to the desired angles (*equilibrium point*) sent to muscles, denoted with $\mathbf{EP}$, the *noisy* global actor motor command, $\mathbf{a}^n$, is mapped into the noisy equilibrium point issued to the muscle models, denoted with $\mathbf{EP}^n$, and the current joint angles are denoted with $\mathbf{J}_t$.

Mathematically, the noisy EP issued to muscle models at time t, $\mathbf{EP}_{nt}$, is computed as follows (measure unit expressed in neural space as the distance between two close units):

$$\begin{aligned}
&\mathbf{N}_t^b = (1-\sigma)\cdot\mathbf{N}_{t-1}^n + \sigma\cdot\mathbf{N}^{rand}\\
&\mathbf{N}_t^n = \mathbf{N}_t^b / \left|\left|\mathbf{N}_t^b\right|\right| \quad if \quad 1 < \left|\left|\mathbf{N}_t^b\right|\right| \quad else \quad \mathbf{N}_t^n = \mathbf{N}_t^b\\
&\mathbf{N}_t = \mathbf{N}_t^n \cdot N^{max} \qquad (14)\\
&\mathbf{EP}_t^{nr} = A\cdot\mathbf{EP}_t^r + (1-A)\cdot\mathbf{N}_t\\
&\mathbf{EP}_t^n = \mathbf{EP}_t^{nr} + \mathbf{J}_{t-1}
\end{aligned}$$

where $\mathbf{N}_t^b$ is a buffer vector, σ (set to 0.05) is a parameter which allows progressively updating $\mathbf{N}_t^b$ on the basis of the noise vector $\mathbf{N}^{rand}$ (whose elements are uniformly drawn in [-1, +1]), $\mathbf{N}_t^n$ is a two-element noise vector with size normalised in [0, 1], $\mathbf{N}_t$ is a noise vector with maximum size N^{max} ($N^{max} = 10$), $\mathbf{EP}_t^r$ is the desired equilibrium point vector produced by the actor but expressed with respect to a reference frame centred on (*relative* to) the previous joint angles $\mathbf{J}_{t-1}$, A is a variable changed in [0.1, 0.9], $\mathbf{EP}_t^n$ is the noisy EP vector issued to the muscle models. Briefly, the rationale of eq. 14 is that the delay mechanism with which $\mathbf{N}_t^n$ is updated on the basis of $\mathbf{N}^{rand}$ assures that the direction and intensity with which noise "pulls" the arm away form the current posture $\mathbf{J}_{t-1}$ changes gradually, so solving the problem of inertia averaging noise out. N^{max} allows regulating the maximum exploration range due to such noise. A is the "ability" of the actor increased linearly from 0.1 to 0.9 during training.



(a) (b)

Figure 4: (a) Y-axis: performance of the four-expert model (black bars) and the one-expert model (gray bars) in the experiment with one handle (two bars at the left) and two handles (two bars at the right: average for the two handle positions). (b) Y-axis: performance of the four-expert model (black bars) and the one-expert model (gray bars) in the experiment with two handles, measured separaterly for the handle at the left (two bars at the right) and at the right (two bars at the left).

4. Results

The performance of the hierarchical model with four experts was compared with one model having only one expert and representing a system with linear computational capabilities. The two systems were tested in three experiments requiring to reach a cup-handle in various conditions (see Fig. 1): (c) an experiment requiring to reach the cup-handle positioned only in one position: at the left of the robot. (b) a similar experiment requiring to reach the cup-handle in two positions: at the left or at the right of the robot; (a) a similar experiment requiring to reach to the cup-handle in three positions: the handle either at the left, or at the right, or in front of the robot. This experiment was run only with the four-expert model as the one-expert model could not tackle this task (as shown by the fact that it failed the easier task with two handles, see below).

(a) (b)

Figure 5: (a) Trajectories followed by the one-expert model to reach the handle at the right, in the experiment with one handle, in 64 trials when the initial position of the hand is set on the 8×8 vertexes of a regular grid overlapped with the joint space. (b) Trajectories followed by the four-expert model to reach the handle at the right, in the experiment with one handle, with the same initial 64 hand positions.

Fig. 3 shows the performance during training of the two models measured as the cumulated reward in the two experiments with one and two handle positions. Table 1 and Fig. 4a show the results of a test on the performance of the two systems after they were trained in these two experiments. This test measured the percentage of times, out of 64, when the hand reached the target with the hand initial position set on the 8×8 vertexes of a regular grid overlapped with the joint space. Note that other experiments run with different random-number generator seeds had qualitatively similar results.

4.1 Experiment with one handle

Fig. 3 shows that the one-expert model learns faster than the four-expert model. This is due to the fact that it does not need to train the gating networks before training the specific critic/actor experts to solve the task. The performance of the four-expert model, however, after some time becomes similar to the performance of the one-expert model, as indicated by the fact that the derivative of the two curves in Fig. 3 become the same, and as confirmed by the test reported in Fig. 4a and in Table 1. The experiments also show that the four-expert model learns to solve the task using only one expert.

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

Fig. 3 shows that in the two handle experiment after an initial transient the four-expert model outperforms the four-expert model. Its superiority is also indicated by the results of the test on the after-learning performance reported in Table 1 and Fig. 4. Data reported in Table 1 and Fig. 4 also indicate that the one-expert model focussed its resources on reaching the left handle (performance: 82.81%) and had a poorer performance with the right handle (14.06%). This indicates that the task is non-linear and so cannot be fully solved by the one-expert model formed by a critic and an actor based only on one linear function approximator, even if the system is forced to adopt this solution by the lack of other neural resources. On the contrary, the four-expert model succeeds to solve the task by employing *two different experts* for both the critic and the actor. This indicates that the four-expert model is both capable of discriminating the two conditions at the level of the gating networks and to exploit each of the two experts to solve the task in one particular condition. Fig. 5 shows some examples of trajectories exhibited by the two models.

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

In the experiment with three handle positions, the actor and critic of the four-expert model both learn to use the same expert for the left and central hanlde and a different expert for the right handle. The reason of this is that the arm needs to execute very similar movements to reach both the left and central handle (in fact if the straight hand moves towards the cup it can touch both handles).

Interestingly, 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 used for the central handle also to reach the left handle (second expert), and the evaluator does the opposite (third expert). This shows that the actor and critic gating networks have the capacity to use the same expert if the conditions where the task is pursued require similar sensorimotor mappings.

5. Conclusions and future work

This article presented a hierarchical modular reinforcement leaning model that when acquires different skills is capable of assigning responsibilities to different expert controllers on the basis of: (a) the different sensorimotor mappings required by the skills; (b) the computational capability of experts. The tests show how the model is capable of autonomously learn to use only one expert for same or similar skills, and more experts for relatively different skills. Thanks to this property, and the fact that the model is hierarchical, is based on a neural biologically-plausible RL model (actor-critic model), and can work with continuous actions and states, it can be used to study developmental processes in future work (e.g., see Berthier et al., 2005). In particular, the results presented here give preliminary indications that the model can indeed be used to investigate the assimi-

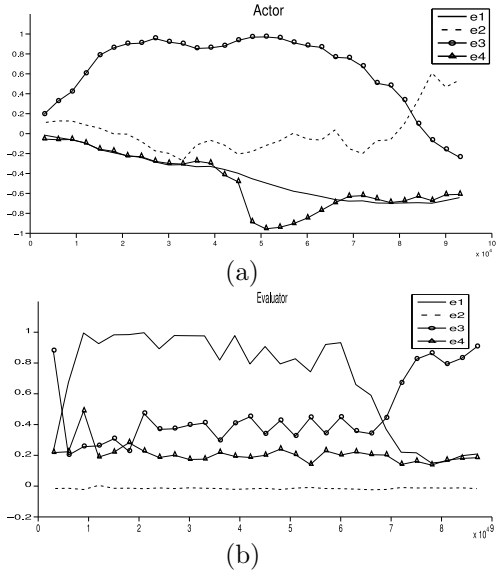


Figure 6: (a) Moving average (1000 steps window) of the activation of the actor selector output units (y axis) of the four-expert model during learning (x-axis), when the system pursues the left handle. (b) Same data for the evaluator when the system pursues the central handle.

lation/accomodation processes proposed by Piaget.

Acknowledgements

This research was supported by the EU Projects *ROSSI*, contract no. FP7-STREP-216125, and *IM-CLeVeR*, contract no. FP7-IP-231722. We thank Prof. 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., Borghi, A. M., Parisi, D., and Baldassarre, G. (inpr). A reinforcement learning model of reaching integrating kinematic and dynamic control in a simulated arm robot. In *Proceedings of ICDL 2010*.

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 MITT 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.