A Model of Reaching That Integrates Reinforcement Learning and Population Encoding of Postures*

Dimitri Ognibene, Angelo Rega, and Gianluca Baldassarre

Laboratory of Autonomous Robotics and Artificial Life,
Istituto di Scienze e Tecnologie della Cognizione,
Consiglio Nazionale delle Ricerche (LARAL-ISTC-CNR),
Via San Martino della Battaglia 44, 00185 Roma, Italy
{dimitri.ognibene, angelo.rega,
gianluca.baldassarre}@istc.cnr.it
http://laral.istc.cnr.it/

Abstract. When monkeys tackle novel complex behavioral tasks by trial-and-error they select actions from repertoires of sensorimotor primitives that allow them to search solutions in a space which is coarser than the space of fine movements. Neuroscientific findings suggested that upper-limb sensorimotor primitives might be encoded, in terms of the final goal-postures they pursue, in premotor cortex. A previous work by the authors reproduced these results in a model based on the idea that cortical pathways learn sensorimotor primitives while basal ganglia learn to assemble and trigger them to pursue complex reward-based goals. This paper extends that model in several directions: a) it uses a Kohonen network to create a neural map with population encoding of postural primitives; b) it proposes an actor-critic reinforcement learning algorithm capable of learning to select those primitives in a biologically plausible fashion (i.e., through a dynamic competition between postures); c) it proposes a procedure to pre-train the actor to select promising primitives when tackling novel reinforcement learning tasks. Some tests (obtained with a task used for studying monkeys engaged in learning reaching-action sequences) show that the model is computationally sound and capable of learning to select sensorimotor primitives from the postures' continuous space on the basis of their population encoding.

1 Introduction

This research is motivated by the idea that when humans and monkeys learn to solve complex tasks by trial-and-error they select and execute *sensorimotor primitives* (that is behavioral chunks that tend to achieve whole goals, cf. [2, 6, 7]) that have a coarse granularity with respect to the detailed commands sent to muscles. By using these primitives, they can learn to tackle complex tasks by assembling relatively few "behavioral chucks" instead of a multitude of fine muscular movements that would make the problems' search space huge. The computational advantages of this strategy have been explored in reinforcement learning literature (see [4] for a review; note that

* This research has been supported by the project "MindRACES - From Reactive to Anticipatory Cognitive Embodied Systems", European Commission's grant FP6-511931.

S. Nolfi et al. (Eds.): SAB 2006, LNAI 4095, pp. 381 – 393, 2006.

within this context sensorimotor primitives are called "macro actions" or "options"). This work is part of a research program directed to design, implement and test computational models that not only mimic animal's behaviors organized on sensorimotor-primitive repertoires, but also account for the neuroscientific evidence related to the brain's mechanisms underlying them. With this regards, an increasing amount of empirical evidence is giving specific indications on how vertebrates' brains *encode repertoires* of sensorimotor primitives and *select* and *assemble* them to flexibly produce complex behaviors. For example, it has been shown that when different areas of frogs' spinal cord are electrically stimulated, their lower limbs tend to assume a discrete number of particular postures in space independently of the initial configuration [6]. Moreover, recordings of neurons' activity in *premotor areas* controlling arms in monkeys that freely move in ecological conditions showed that the biggest amount of variance of the neurons' firing rate is explained by the final postures achieved by the limbs [1, 8]. Remarkably, other aspects of movement previously hypothesized to be encoded in premotor cortex, such as direction of movement, hand position, torques, and speed of motion, explained much less or none of the remaining variance.

A general hypothesis on the brain's architecture that might underlie reinforcement learning and behavior based on sensorimotor primitives has been proposed in [10] and has been used for building a modular reinforcement-learning model in [3]. According to this hypothesis sensorimotor primitives are acquired and executed by *cortical pathways* that involve sensory, associative, premotor, and motor cortex. These primitives are then *assembled, selected* and *triggered* to produce reinforcement-based complex behavior by *basal ganglia* (deep nuclei of vertebrates' brain that receive input signals from virtually the whole cortex, send output signals mainly to pre-frontal, premotor and motor cortex [12], and play an important role in chunking and assembling motor primitives in order to accomplish complex reinforcement-based behaviors [7, 8, 11]). This hypothesis has been further investigated in [16] by building a biomimetic model that explicitly incorporates the aforementioned biological evidence reported in [1, 8].

As the model presented shares many features with the model reported in [16], first these features are reviewed and then the main novelties introduced here are highlighted. In both models sensorimotor primitives are neural schemes that allow the system to produce sequences of fine movements that lead the arm to assume particular *final postures*. Both models learn the primitives through a *direct inverse modeling* process [14] based on *spontaneous random movements* performed by the system. The latter aspect of the process is interesting as it is very similar to *motor babbling* observed in infants [15] and might have functions similar to it. In both models, random movements are used for learning to associate limbs' final postures with the movements that led to them. Final postures are represented in a 2D neural map that mimics the function of premotor cortex reported in [1, 8]. Note that such *final postures* can be considered as the *goals* of the corresponding primitives, in fact: (a) the activations of the map's units correlate with the final postures of primitives, but not with other aspects of them (e.g., initial and intermediate postures); (b) the activations take place before the corresponding final-posture states are achieved; (c) the activations drive the system to act in order to get in the states that they encode. The representation of primitives in terms of their goals in the map has the computational advantage of being (almost) local: this eases the selection of them by reinforcement-learning systems (see section 2). Both models assume that basal ganglia select primitives by fueling a

dynamic competition between their representations in the map: the representation that wins the competition triggers the execution of the corresponding primitive. In both models, the functionalities of basal ganglia are reproduced with an actor-critic reinforcement-learning model [23]. This model captures several anatomical and physiological properties of basal ganglia [3, 10, 11]. The dynamic competition between goals is simulated through an *accumulator model* [24]. Accumulator models are among the best behavioral models of decision making and reaction times; moreover, the activation patterns of their units are similar to those of neurons of premotor cortex of monkeys engaged in action selection tasks [21, 22].

The first novelty of the model presented here is that, while in [16] the representations of the sensorimotor primitives' goals in the 2D map were hand coded, they are now developed through a Kohonen network [13] which takes the arm's angles as input. This has the advantage of leading the map's units to cover the space of "legal" postures in a uniform fashion. Moreover, contrary to [16], the model is now capable of representing all possible postures of the arm in the continuous space of postures by representing them through a *population encoding* [18]. To this purpose, the previously used winner-take-all dynamic competition taking places within the accumulator model has been substituted with a many-winner dynamic competition. A second novelty is the proposal of a modified version of the actor-critic reinforcement-learning algorithm capable of selecting postures on the basis of such population encoding (to the best of the authors' knowledge, the learning rule used for training the actor is new). A third novelty is that the system performs a "pre-training" of the actor on the basis of the same motor babbling used for training the sensorimotor primitives. This pre-training allows the actor to learn to associate the *perceived hand's position* with the posture that produces it, and so biases the actor to select sensorimotor primitives that drive the hand on *salient* points in space such as those occupied by objects. This greatly speeds up learning when the system tackles new reinforcement-learning tasks. The whole architecture is tested through a task similar to the one used in [19] to conduct physiological studies in monkeys engaged in reinforcement-learning action-sequence tasks.

The paper is organized as follows. Section 2 illustrates the architecture and functioning of the model, and the task used to test it. Section 3 presents the results of the tests. Section 4 illustrates the strengths of the model, its limitations, and future work.

2 Methods

The Task. The model has been tested with a task similar to the one used by Hikosaka and coworkers [19] to carry out physiological studies of various brain's districts (e.g., frontal cortex, basal ganglia, and cerebellum) of monkeys engaged in learning to perform sequences of reaching actions. In this task a monkey is set in front of a panel containing 16 LED buttons. These buttons are contained in 16 squares organized in a 4×4 grid, each with sides measuring 5 cm (see Fig. 7). The task (see figure Fig. 1) is formed by "hypersets", each composed of five "sets" organized in sequence. In each set, two buttons turn on and the monkey has to press each of them in a precise sequence, which has to be discovered by trial-and-error, in order to obtain a reward. In case of error, the task re-starts from the first set, while in case of success the task

continues with the second set, and so on, until it terminates with the fifth set. Here for simplicity: (a) the test is composed of only one particular hyperset (see Fig. 1) presented to the system several times; (b) the buttons involved in different sets are different; (c) the first LED to be "pressed" in each set is turned off when reached.

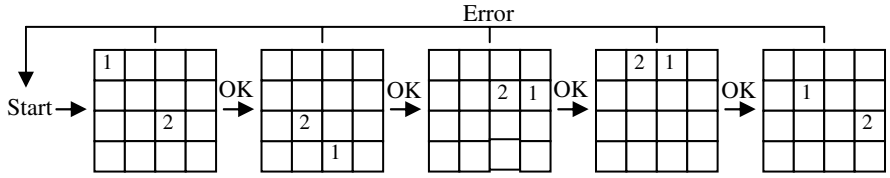

Fig. 1. The "hyperset" of Hikosaka's task used for testing the architecture. Each grid represents a "set": numbers "1" and "2" represent the two LEDs to be reached in sequence within the set.

The system's "body". The system is composed of a two-segment arm that moves on a 2D plane (Fig. 7, left), and a 2D retina. The *retina* is formed by 20×20 units and is supposed to correspond to an "eye" that watches the whole area that the arm can reach from above. The retina's visual field has a size of 40×40 cm and is centered on the arm's shoulder joint (so as to cover the whole area that the arm can reach). The centers of the retina's units are organized in a 20×20 grid that cover to whole visual field. The two segments of the *arm* measure 20 cm each. The arm has two degrees of freedom: the upper arm can move 180° with respect to the system's torso, by pivoting on the shoulder joint, while the forearm can move 180° with respect to the upper arm, by pivoting on the elbow joint (only simple kinematics of the arm were simulated).

The Architecture of the Model. The architecture of the model is shown in Fig. 2. The functioning and learning processes of its components will now be explained in detail (note: the corresponding brain parts will be indicated in *Italics* in brackets).

The *retina*'s units are activated by LEDs. Each LED is simulated as a point with coordinates (c_1, c_2) and when it is on, it activates the retina's units with an activation $x_i \in [0, 1]$ on the basis of Gaussian receptive fields having standard deviation σ (0.75 cm) and centers (c_{1i}, c_{2i}) that correspond to the positions of the units in the visual field:

$$x_i = \exp\left(-\frac{(c_{1i}-c_1)^2+(c_{2i}-c_2)^2}{\sigma^2}\right) \quad (1)$$

The *actor-critic* components are a neural implementation of the actor-critic model [23]. The *actor* (*basal ganglia's matrix,* cf. [10]) is a two-layer feed-forward neural network with 20×20 input units, that correspond to the units of the retina, and 20×20 output units. The output units have a Sigmoid transfer function with activation y_j and each has a topological one-to-one connection (with weights equal to $\upsilon = +1$) with the posture controller's input units. The *critic* (*basal ganglia's striosomes* and *substantia nigra pars-compacta,* cf. [10]) is mainly composed of a neural network ("evaluator") having a linear output unit. At each step t this output unit produces evaluations V_t of perceived states, and the critic uses couples of successive evaluations, together with the reward signal R_t, to compute the *surprise* signal S_t (*dopamine*) (cf. [23]):

$$S_t = (R_t + \gamma V_t) - V_{t-1} \tag{2}$$

where γ is a discount factor ($\gamma = 0.3$). The surprise signal is used for training both the actor and the evaluator (see [10, 23] and the learning algorithms presented below).

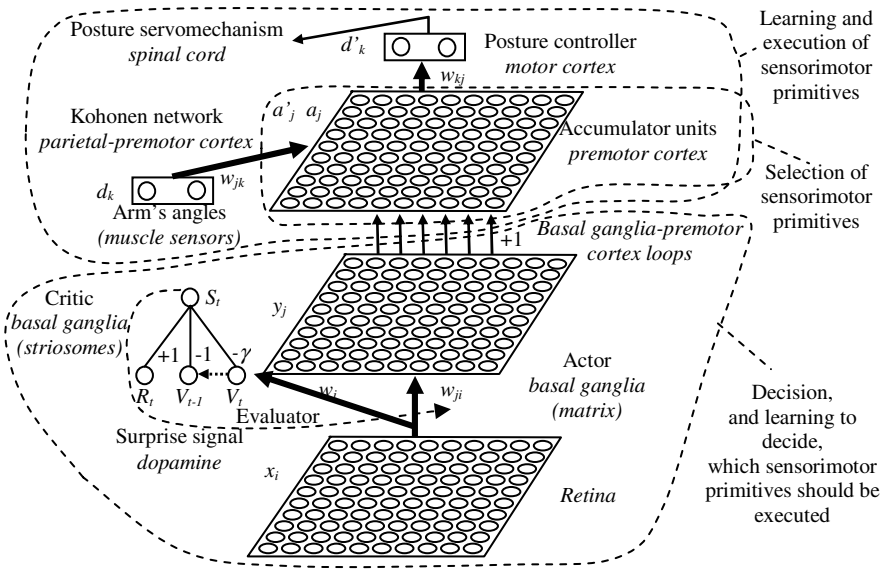

Fig. 2. The neural components of the architecture with the corresponding brain areas in *Italics*. Symbols: grouping: broad functionalities implemented by the architecture's main parts; bold arrows: all-to-all trained connections; thin arrows (only few of them are shown): one-to-one connections (weights = +1); dashed arrow: surprise learning signal; dotted arrow: delay connection; the weights of the critic's one-to-one connections are indicated in the figure.

The *accumulator units* (*premotor cortex*) form a 2D 20×20 map, have all-to-all lateral inhibitions, and have local excitations that decrease with distance on the map. The units engage in a many-winner competition on the basis of the signals ("votes") that they receive from the actor's output units via the one-to-one connections. In particular, they behave as *leaky-integrators* and have an activation a_j as follows:

$$\begin{aligned} a_{j\,t} &= \max\left[\left(a_{j\,t-1} + \frac{dt}{\tau}(D)\right), 0\right] \\ D &= \chi\left(-\delta a_{jt-1} - \iota \sum_{l,l\neq j} a_{lt-1} + \eta \sum_{l,l\neq j} e_{jl} a_{l\,t-1} + \upsilon y_j + \varepsilon_{jt} + \varepsilon_{jc}\right) \end{aligned} \tag{3}$$

where τ is a time constant, corresponding to 1/10s, dt is the integration time step (dt = 0.05 1/10s, so $dt/\tau = 0.05$; a_j is numerically updated every 0.005 s), χ regulates the speed of the dynamics ($\chi = 1$), δ is a decay coefficient ($\delta = 0.1$), ι regulates the all-to-all lateral inhibition ($\iota = 0.15$), η regulates the local lateral excitation ($\eta = 1$), e_k represents the fixed weights of the lateral excitatory connections (e_k is set to 0.4 for

neighboring units along the x/y-axes directions, to 0.2 for neighboring units along the diagonals, and to zero for all other units), ε_{jt} is a noise component that ranges over [-0.1, +0.1] and varies in each cycle, ε_{jc} is a noise component that ranges over [-0.25, +0.25] and is constant for time intervals c randomly drawn from [0, 5] s (ε_{jc} is important for exploration of reinforcement learning as various ε_{jt} tends to sum to zero over many steps). When the activation a_j of one accumulator unit reaches a threshold T (T = 1.9), the total activation of accumulator units is *normalized to 1*, their dynamics is "frozen", and the execution of a reaching sensorimotor primitive is triggered.

The *posture controller* has an input-unit layer corresponding to the accumulator units and two Sigmoid output units, with activation d'_k, that range over [0, 1] (*motor cortex/spinal cord neurons*). The activations of these output units are remapped onto the arms' angles and form the commands issued to the posture servomechanism in terms of arms' desired angles (posture). It is important to notice that these desired angles are generated by the *cluster* of accumulator units that are active at the end of the many-winner competition. This implies that the target of the executed sensorimotor primitive is a *mixture* of the targets "suggested" by all active units: this *population encoding* allows the arm to cover the whole continuous space of postures.

The *posture servomechanism* is a hardwired closed-loop controller (*Golgi tendon-organs*, *muscle-fiber afferents,* and *spinal cord,* cf. [22]) that issues commands to the arm's actuators (*muscles*) on the basis of the desired-posture command received from the posture controller. In practice, this component simply changes the arm's current angles in the direction of the desired angles, with maximum changes of 10 degrees.

Learning Phases. The learning processes take place in two phases, the *childhood phase* (three processes) and the *adulthood phase* (one process). Now we first present an overview of these learning processes and then describe them in detail.

During the childhood phase the system performs motor babbling: in practice the arm randomly varies its joints' angles, with changes $\Delta d'_k$ belonging to [-10, +10] degrees, without violating the joints' constraints. Motor babbling is used for performing three learning processes. The first two processes allow the system to learn to perform sensorimotor primitives, in particular: (a) to train the 2D map of accumulator units, through a Kohonen algorithm [13], to represent the postures perceived by the proprioceptive units d_k (during the childhood phase the proprioceptive units, the accumulator units, and their connections, function as a Kohonen network); (b) to train the posture controller, through a Widrow-Hoff algorithm [20] (the generalized "delta rule"), to return as output the arm's angles corresponding to postures encoded in the Kohonen map. These two training processes lead the whole network formed by the Kohonen network and the posture controller to implement an "auto associative" function (i.e., the arm's angles encoded in the proprioceptive units are returned by the postural controller's output units). This whole network allows the system to recode postures, at the level of accumulator units, in an expanded format suitable to perform actor-critic reinforcement learning (cf. [23]). Notice that suitable *population encodings* at the level of the accumulator units allow the system to select *any posture* in the *continuous space* of postures: this is precisely what the actor-critic components learn to do while solving reinforcement-learning reaching tasks in the adulthood phase.

With the third learning process of the childhood phase the system's actor learns, through a Widrow-Hoff algorithm, to associate the point in space where the retina sees

the arm's "hand" (i.e., the forearm segment's tip) with the activation pattern of the Kohonen map's units corresponding to such point (pattern caused by the arm's perceived angles). With this training, the actor acquires a bias to select sensorimotor primitives that drive the arm's hand to points in space corresponding to the retina's active units. This bias makes reinforcement learning performed during the adulthood phase quite fast notwithstanding the fact that the continuous space of postures is quite large. Note that two simplifying assumptions allow obtaining this result: (a) the retina does not perceive the arm and hand in the adulthood phase; (b) retina's units activated by the hand in the childhood phase are activated by the LEDs in the adulthood phase.

During the adulthood phase the system learns by trial-and-error to accomplish Hikosaka's task. The actor-critic model used to this purpose has been suitably modified to be capable of selecting "actions" represented with population encodings. The four learning processes are now illustrated in detail.

Childhood phase: training of the Kohonen network. During the childhood phase, while the system performs motor babbling, the accumulator units receive input signals from two input units, having activation d_k, that encode the arm's current angles (remapped in [-1, +1]: this information is thought to be returned by proprioceptive sensors located in the muscles, e.g. *Golgi tendon-organs* and *muscle-fiber afferents*, cf. [22]). An extra pseudo input unit is used to perform a "z-normalisation" of the input pattern: this is a normalization that preserves size information [13]. The accumulator units are trained with a Kohonen algorithm [13] that allows them to develop representations of the arm's angles in their weights. The output units give place to a winner-take-all competition: the unit with the highest activation potential activates with 1 ("winning unit"), while the other units activate at levels decreasing with their distance from the winning unit on the basis of a Gaussian function. In particular, the activation a'_j of the unit j and the rule to update its weights w_{jk} are as follows:

$$a'_j = \exp\left[-\frac{h_{fj}^2}{\sigma^2}\right] \qquad w_{jk\,t} = w_{jk\,t-1} + \phi\, a'_j \left(d_k - w_{jk\,t-1}\right) \tag{4}$$

where h_{fj} is the distance on the map between the unit j and the winning unit f ($h_{fj} = 1$ for two contiguous units), σ is the standard deviation of the Gaussian function ($\sigma = 1$), ϕ is a learning coefficient ($\phi = 0.01$). Note that the Kohonen algorithm uses a *winner-take-all* competition to activate the accumulator units instead of the *dynamic competition* reported in equation 3, used in the adulthood phase: indeed, the former tends to lead to an activation of the accumulator units that approximates the steady state activation that the same units would get through the latter (cf. [13]).

Childhood phase: training of the posture controller. The posture controller is trained on the basis of a direct inverse modeling procedure [14] that exploits the random movements $\Delta d'_k$ produced by motor babbling as follows: (a) the arm's angles are perceived and categorized by the Kohonen net; (b) a Widrow-Hoff algorithm ([20], learning rate = 0.3) is used for training the posture controller's weights w_{kj} to associate the Kohonen-map units' activation (input pattern) with the angles d'_k caused by the random movements considered as desired output.

Childhood phase: pre-training of the actor. Through this pre-training, based on a Widrow-Hoff algorithm, the actor's weights w_{ji} are trained to associate the position of

the hand perceived with the retina (input pattern $\boldsymbol{x}$) with the corresponding posture (desired output $\boldsymbol{a}'$) encoded in the Kohonen map (learning rate 0.1).

Adulthood phase: actor-critic's reinforcement learning. During the adulthood phase, the actor-critic component is trained to solve the Hikosaka's task by reinforcement learning. During training, R_t is set to 1 when the arm reaches the two targets of any set of the hyperset in the correct order, and to 0 otherwise. The *evaluator* is trained after the selection and execution of a whole sensorimotor primitive (the primitive terminates when the arm reaches the desired posture selected by the posture controller). In particular its weights w_i are trained, on the basis of a Widrow-Hoff algorithm (learning rate ψ = 0.6) and a *TD-rule* (cf. [23]), as follows:

$$w_{it} = w_{it-1} + \psi S_t x_{it-1} = w_{it-1} + \psi\left(\left(R_t + \gamma V_t\right) - V_{t-1}\right)x_{it-1} \tag{5}$$

Through this learning process, the evaluator's evaluations V_t of the perceived states $\boldsymbol{x}_t$ tend to become higher for states corresponding to postures "closer" to reinforced states, and to form a gradient over the space of postures. The *actor* uses this gradient to learn to select highly rewarding sequences of primitives (cf. [23]). In particular the actor updates its weights w_{ji} with a Widrow-Hoff algorithm (learning rate ζ = 0.6):

$$w_{jit} = w_{jit-1} + \zeta\left(\left(y_{jt-1} + S_t a_{jt-1}\right) - y_{jt-1}\right)\left(y_{jt-1}\left(1 - y_{jt-1}\right)\right)x_{it-1} \tag{6}$$

where $(y_{jt-1}(1-y_{jt-1}))$ is the derivative of the Sigmoid function. The functioning of this learning rule is illustrated in Fig. 3. The rule tends to update only the weights of the units of the "winning cluster" because the activation a_j of other units tends to be zero at the end of the race. The votes of the winning units are decreased or increased in correspondence of respectively positive and negative surprises.

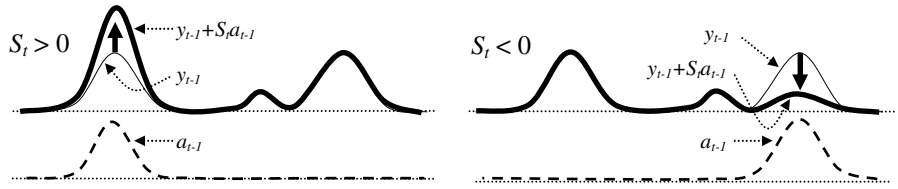

Fig. 3. Effects of the actor's learning rule of equation 6 illustrated with a scheme relative to a 1D layer of actor's output units (horizontal axis). Left: with a surprise $S_t > 0$, the actor's votes y_{t-1} (upper graph), that caused certain accumulator units' final activations a_{t-1} (lower graph), are moved toward the target $y_{t-1}+S_t a_{t-1}$ (upper graph): this causes the votes of the winning cluster of accumulator units to increase (bold arrow) while other votes are not changed. Right: with a surprise $S_t < 0$, actor's votes y_{t-1} are moved toward the target $y_{t-1}+S_t a_{t-1}$: this causes the votes of the winning cluster of accumulator units to decrease, while other votes are not changed.

3 Results

Now we present some tests that prove the computational soundness of the model, illustrate the functioning of its components, and show its capacity to learn sensorimotor primitives, by motor babbling, and to compose sequences of them, by reinforcement learning, on the basis of their population encoding.

During the first training of the childhood phase, the Kohonen network's error (measured as the average over 1,000 cycles of the square of the norm of the difference between the vector of weights and the vector of the input pattern) decreases from 0.411 to 0.034 after 600,000 random arm's movements. After this training the network learns to represent the whole perceived postural space by using its units in a statistically well-distributed fashion (Fig. 4, left graph). This representation is at the basis of the *population encoding* of postures used in the adulthood phase.

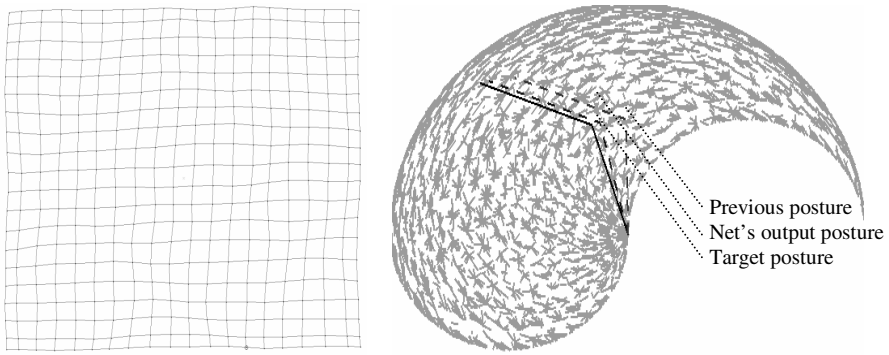

Fig. 4. Left: result of the training of the Kohonen network. Each vertex of the grid represents a node of the Kohonen map, and its x-y coordinates correspond to the node's two weights encoding the arm posture. Right: errors of the posture controller after training, collected while the arm produces several random movements; the graph represents the errors as gray segments plotted between the x-y positions of the hand corresponding to the target actual posture (e.g., black arm) and the position that the hand would have achieved on the basis of the posture controller's output pattern (e.g., dark gray dashed arm; the light gray dashed arm indicates the previous posture assumed by the arm during motor babbling).

During the second training of the childhood phase, the posture controller's error (measured as the average over 1,000 cycles of the distance between the point reached by the arm and the target point) decreases from 8.62 cm to 1.19 cm. Note that this error cannot become very low since the Kohonen network's units are activated on the basis of a Gaussian function *centered on the winner units*, that are in a *finite number*, while the desired output belongs to the whole *continuous* space of arm's postures. Indeed, the right graph of Fig. 4, which shows the residual errors after training, indicates that the hand tends to reach only few specific points corresponding to the vertex of a grid that covers the whole postural space (this grid is explicitly represented in Fig. 5, right graph). In the adulthood phase, this problem is overcome by the population encoding of postures resulting from the accumulator units' activation.

During the third training of the childhood phase, the actor's error (measured as the output units' mean error averaged over 1,000 cycles) decreases from 0.513 to 0.052. This training leads the system formed by the actor, accumulator units, and postural controller to acquire the capacity to perform fine reaching movements in the continuous space of postures even if the accumulator units cover such space at a gross

granularity. This can be illustrated by showing the system a sequence of 100 targets positioned along a circumference having a ray of 10 cm and located near the arm's shoulder (see Fig. 5, right graph). The left graph of Fig. 5, which shows the errors between the targets and the points reached by the hand in the test, indicates that the errors are very small (mean: 3.2 mm). Moreover, and more importantly, the system succeeds in reaching virtually any point in the continuous space of postures even if the accumulator units cover such a space with a gross granularity. This skill depends on the mentioned accumulator units' capacity to represent postures by population encodings.

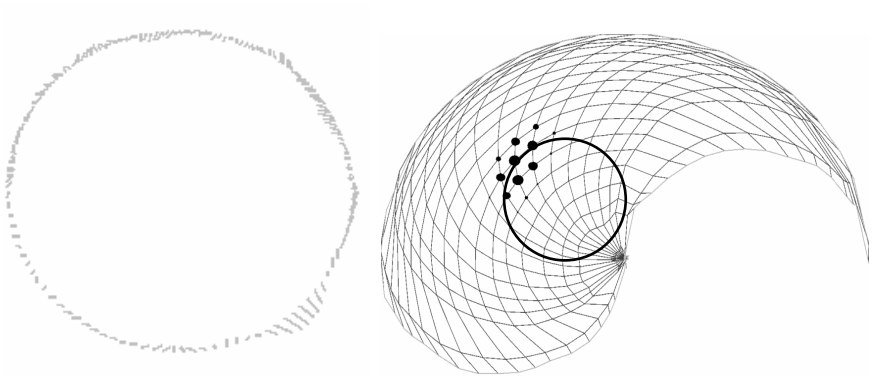

Fig. 5. Left: errors (indicated by the gray segments) between 100 target points positioned on a circumference (shown in the right graph) and the corresponding points reached by the hand. Right: activation (proportional to the size of the full dots) of the actor's output units caused by a target. The positions of the dots and vertexes of the grid plotted in the graph correspond to the positions of the hand related to the "postures" encoded in the accumulator units' weights of the posture controller.

During the adulthood phase, the system is tested with the Hikosaka's task illustrated in section 2. During 120,000 learning cycles, the performance of the system (measured as a 1000-step moving average of rewards) increases from 0.187 to the theoretical maximum of 0.500, when it successfully completes all the five sets of the task in sequence. The results show that the pre-training of the actor gives it a useful bias to reach the targets perceived by the retina. In particular the left graph of Fig. 6, reporting the activations of the actor's output units when the system sees two targets, shows that the units that "vote" for the two possible correct arm's postures form two clusters and have an activation higher than that of other units. The same figure (right graph) shows that the two clusters compete, at the level of the accumulator units, and only one of them "survives" and triggers the corresponding arm's posture when the activation of one of its units reaches the threshold. The left graph of Fig. 7 shows how the arm moves from one target to another, after target postures have been selected, on the basis of the postural servo controller. The same figure (right graph) also shows that the final points reached by the trained arm are quite accurate.

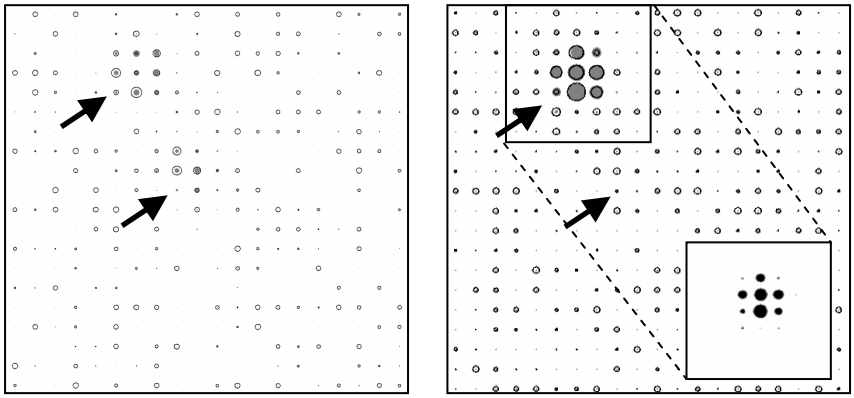

Fig. 6. Left: activations of the actor's output units before adulthood training caused by the perception of two targets in the Hikosaka's task (the area of the gray dots and black circumferences is proportional to the units' activations respectively before and after the addition of noise); the two arrows indicate two clusters of units with activation higher than that of the other units due to the actor's pre-training. Right: activation of the same units after training; notice how one of the two clusters has been strengthened while the other one has disappeared; the activation of the units of the strengthened cluster cause an activation of the accumulator units, at the end of the race, as plotted in the bottom right small graph.

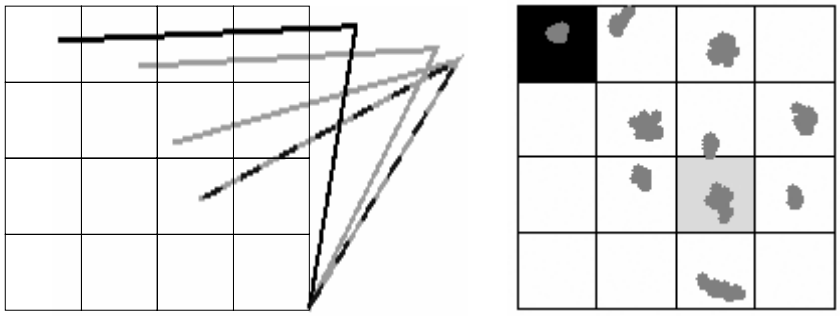

Fig. 7. Left: the trained arm that moves from the first to the second LED of "set 1" of Hikosaka's test under the control of the postural servomechanism (the two LEDs are represented by the black and light gray squares in the right graph). Right: panel with the LEDs, with gray dots indicating the positions reached by the hand of the trained arm in several trials of the hyperset.

4 Conclusions

This paper presented an architecture to solve reaching tasks by reinforcement learning. The architecture is based on the idea, suggested by recent neuroscientific research, according to which monkeys' sensorimotor behavior involving upper-limbs is organized on the basis of a repertoire of sensorimotor primitives that are represented in premotor cortex in terms of the limbs' final postures that they produce. The architecture uses motor babbling to learn sensorimotor primitives, develops a map of units

that represent the corresponding postures on the basis of population encodings (so mimicking premotor cortex), and selects primitives on the basis of a biological-plausible accumulation model. Moreover, it proposes a novel learning rule which allows the actor of the actor-critic component (supposed to correspond to basal ganglia) to learn to select sensorimotor primitives on the basis of the population-encoding of their postural goals. The relevance of these novelties resides in the fact that population-encoding representations are widespread in real brains [18], so it is important to have reinforcement-learning models that can function on the basis of them.

The main limitations of the architecture that will be the starting point for future work. First, tests are needed to verify if the system can scale to arms with redundant degrees of freedom and/or to arms with a number of degrees of freedom higher than the number of the dimensions of the Kohonen network. Second, the Kohonen network functions on the basis of a winner-take-all competition: in the future this will be substituted with the same many-winner competition used while performing reinforcement learning. This improvement is relevant for the biological plausibility of the system. Third, although very detailed, the architecture takes into account only a part of the relevant available neuroscientific empirical evidence. For example, it does not model the different time courses of learning in basal ganglia and prefrontal cortex [17], the role of basal-ganglia direct and indirect pathways [10, 12], the possible separation of selection vs. control pathways [9], and the role of ventral and dorsal portions for appetitive and consummatory behaviors [5].

References

1. Aflalo, T.N., Graziano, M.S.A.: Partial Tuning of Motor Cortex Neurons to Final Posture in a Free-Moving Paradigm. Proceedings of the National Academy of Science 103(8) (2006) 2909-2914
2. Arbib, M.: Visuomotor Coordination: From Neural Nets to Schema Theory. Cognition and Brain Theory 4 (1981) 23-39
3. Baldassarre, G.: A Modular Neural-Network Model of the Basal Ganglia's Role in Learning and Selecting Motor Behaviours. Journal of Cognitive Systems Research 3 (2002) 5-13
4. Barto, A.G., Mahadevan S.: Recent Advances in Hierarchical Reinforcement Learning. Discrete Event Dynamic Systems, 13 (2003) 341-379
5. Girard, B., Filliat, D., Meyer, J.-A., Berthoz, A., and Guillot, A.: Integration of Navigation and Action Selection Functionalities in a Computational Model of Cortico-Basal Ganglia-Thalamo-Cortical Loops. Adaptive Behavior 13(2) (2005) 115-130
6. Giszter, S.F., Mussa-Ivaldi, F.A., Bizzi, E.: Convergent Force Fields Organised in the Frog's Spinal Cord. Journal of Neuroscience 13 (2) (1993) 467-491
7. Graybiel, A. M.: The Basal Ganglia and Chunking of Action Repertoires. Neurobiology of Learning and Memory 70 (1998) 119-136
8. Graziano, M.S., Taylor, C.S., Moore, T.: Complex Movements Evoked by Microstimulation of Precentral Cortex. Neuron 34 (2002) 841-851.
9. Gurney, K., Prescott, T. J., Redgrave, P.: A Computational Model of Action Selection in the Basal Ganglia I. A New Functional Anatomy. Biological Cybernetics, 84 (2001) 401-410.
10. Houk, J.C., Davis, J.L., Beiser, D.G. (eds.): Models of Information Processing in the Basal Ganglia. MIT Press, Cambridge MA (1995)

11. Joel, D.E.E., Niv, Y., Ruppin, E.: Actor-critic Models of the Basal Ganglia: New Anatomical and Computational Perspectives. Neural Networks 15 (2002) 535-547.
12. Kandel, E.R., Schwartz, J.H., Jessell, T.M.: Principles of Neural Science. McGraw-Hill, New York (2000)
13. Kohonen, T.: Self-Organizing Maps. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (2001)
14. Kuperstein, M.: A Neural Model of Adaptive Hand-Eye Coordination for Single Postures. Science 239 (1988) 1308-1311
15. Meltzoff, A. N., Moore, M. K.: Explaining Facial Imitation: A Theoretical Model. Early Development and Parenting 6 (1997) 179-192
16. Ognibene, D., Mannella, F., Pezzulo, G., Baldassarre, G.: Integrating Reinforcement-Learning, Accumulator Models, and Motor-Primitives to Study Action Selection and Reaching in Monkeys. In: Fum, D., Del Missier, F., Stocco, A. (eds.): Proceedings of the 7th International Conference on Cognitive Modelling - ICCM06 (2006) 214-219
17. Pasupathy, A., Miller E.K.: Different Time Courses of Learning-Related Activity In the Prefrontal Cortex and Striatum. Nature 433 (2005) 873-876
18. Pouget A., Lathaam P. E.: Population Codes. In : Arbib, M. A. (ed.): The Handbook of Brain Theory and Neural Networks. MIT Press, Cambridge MA (2003) 893-897
19. Rand, M.K., Hikosaka, O., Miyachi, S., Lu, X., Miyashita, K.: Characteristics of a Long-Term Procedural Skill in the Monkey. Experimental Brain Research 118 (1998) 293-297
20. Widrow, B., Hoff, M.E.: Adaptive Switching Circuits. IRE WESCON Convention Record, Part 4 (1960) 96-104
21. Schall, J.D.: Neural Basis of Deciding, Choosing and Acting. Nature Reviews Neuroscience 2 (2001) 33-42
22. Shadmehr, R., Wise, S.: The Computational Neurobiology of Reaching and Pointing. MIT Press, Cambridge MA (2005)
23. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge MA (1998)
24. Usher, M., McClelland, J.L.: On the Time Course of Perceptual Choice: The Leaky Competing Accumulator Model. Psychological Review 108 (2001) 550-592.