A Reinforcement Learning Model of Reaching Integrating Kinematic and Dynamic Control in a Simulated Arm Robot

Daniele Caligiore[1,2], Eugenio Guglielmelli[2], Anna M. Borghi[1], Domenico Parisi[1] and Gianluca Baldassarre[1]
[1]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
Email: {daniele.caligiore, domenico.parisi, gianluca.baldassarre}@istc.cnr.it, annamaria.borghi@unibo.it
[2]Biomedical Robotics and Biomicrosystem Lab, Università Campus Biomedico
Via Álvaro del Portillo 21, I-00128 Roma, Italy
Email: e.guglielmelli@unicampus.it

***Abstract*—Models proposed within the literature of motor control have polarised around two classes of controllers which differ in terms of controlled variables: the Force-Control Models (FCMs), based on dynamic control, and the Equilibrium-Point Models (EPMs), based on kinematic control. This paper proposes a bioinspired model which aims to exploit the strengths of the two classes of models. The model is tested with a 3D physical simulator of a 2DOF-controlled arm robot engaged in a reaching task which requires the production of curved trajectories to be solved. The model is based on an actor-critic reinforcement-learning algorithm which uses neural maps to represent both percepts and actions encoded as joint-angle desired equilibrium points (EPs), and a noise generator suitable for fine tuning the exploration/exploitation ratio. The tests of the model show how it is capable of exploiting the simplicity and speed of learning of EPMs as well as the flexibility of FCMs in generating curved trajectories. Overall, the model represents a first step towards the generation of models which exploit the strengths of both EPMs and FCMs and has the potential of being used as a new tool for investigating phenomena related to the organisation and learning of motor behaviour in organisms.**

I. Introduction

An important issue which has not yet been settled within the literature of motor control is the "controlled variable problem" [1]: does the central nervous system of organisms control movements in terms of dynamic variables (e.g. forces and torques) or kinematic variables (e.g. joint angles and postures)? The two possibilities, often at the core of antagonist theories on the organisation of motor behaviour, have generated a number of different models based on either one of the two assumptions (*Force Control Models*, FCMs, vs. *Equilibrium Point Models*, EPMs). This work contributes to this debate by presenting a model which integrates some of the strengths of FCMs and EPMs. This is done within a *developmental* perspective which investigates not only how motor control *is organised* but also how it *becomes organised* as it is.

An important distinction for the debate is the one between *forward models*, which mimic the input-output causal flow of the motor apparatus [2], and *inverse models*, which generate values of torques/forces starting from kinematics [3] [4]. FCMs often use forward models to generate values of relevant variables in a predictive manner, for example to produce anticipated "simulated" sensory feedbacks in the presence of feedback delays so as to guarantee motion stability [2]. A possible drawback of the use of forward models, and hence of FCMs using them, is that they need to be trained on the basis of their prediction errors [5] while in some cases such errors might not be directly available to organism brains. A further drawback of FCMs is that in some conditions they fail to assure motion stability [6].

EPMs are inspired by the idea that brain does not explicitly compute the necessary joint torques [7], [8]. Rather, it sets desired equilibrium positions for the limbs and on this basis muscles and reflexes generate suitable torques and guarantee stability with their spring-like and dumping properties [9]. These processes allow EPMs to avoid using inverse dynamics calculations so simplifying movement planning [7] [10]. EPMs also offer interesting solutions to the problems of motor redundancy [11]. An important drawback of EPMs is that the generation of suitable EPs might be very difficult to be obtained if the system needs to produce complex movement trajectories [12], for example as those produced on the basis of pattern-generator mechanisms [13] [14]. In this respect, a number of EPMs have been proposed to study the development of reaching skills in infants [15], to analyze kinematically redundant reaching movements [16], or to model the detailed functioning of reaching trajectories formation at the neural level [17]. However, in these studies the models are not capable of performing reaching movements with sophisticated curved trajectories (e.g. reaching trajectories preparing grasp actions or avoiding obstacles). The reason of this is likely that these trajectories require the generation of EPs in an *anticipatory fashion* with respect to the dynamic aspects of

This research was supported by the EU Projects *ROSSI*, contract no. FP7-216125-STREP, and *IM-CLeVeR*, contract no. FP7-IST-231722-IP

movements (inertia, Coriolis forces, etc.), and this is difficult to obtain without forward models.

This paper proposes a reinforcement-learning neural model which represents the first step of a research agenda aiming at producing models capturing the strengths of both FCMs and EPMs. To this purpose, the model is based on the following key bioinspired elements: (a) control in terms of *EPs* and generation of joint torques with *muscle models* [8] [7] [15] [18]; (b) *continuous update of EPs* based on current states (here proprioception) which allows generating curved trajectories in a flexible fashion as in real organisms [19]; (c) an architecture based on *neural maps* encoding information with *population codes* [20]; (d) a biologically-plausible *reinforcement learning* (RL) algorithm [21], used here to mimic the learning of organisms of suitable mappings between proprioception and EPs; (e) a *noise mechanism* which allows fine tuning the exploration/eploitation ratio and can work with continuous actions and limbs with relevant inertia.

The model captures a number of advantages of EPMs and FCMs (see also Sec. IV). As EPMs, it does not need forward models and its stability is guaranteed by the use of muscle models (point "a" above; in this respect, however, we will see that for simplicity the model assumes the availability of a non-delayed proprioceptive input which simulates the effects of the use of a feed-forward model such as those used by FCMs; this eliminates an important source of instability). Moreover, although the model does not use forward models it exhibits interesting "implicit anticipatory properties" [22] [10] which allow it to generate complex curved trajectories. In particular, the possibility of continuously updating the EPs (point "b") allow the RL algorithm and the tunable noise mechanism (points "d" and "e") to progressively *optimise the EPs by taking into consideration all the dynamics aspects of the arm*. Indeed, the system can learn to suitably update EPs to compensate inertial forces, Coriolis forces, and the dynamical interactions between the arm links so as to follow suitable curved trajectories. This is the most important contribute of this work.

In the rest of the paper, Sect. II explains the functioning and learning of the model and presents the simulated robotic setup used to test it, Sect. III-B shows the results of the tests of the model, and finally Sect. IV draws the conclusions.

II. Methods

A. The Simulated Robot and Task

Fig. 1 shows the simulated robot and environment. The environment contains a working plane with a simplified blue "cup" on it having a handle on the right hand side. The cup is solidly anchored to the table. The controller controls only 2DOFs of the arm and this moves on the plane. The hand is always kept open and straight (see Sect. II-A2). 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, otherwise it does not get any reward. Notice that, notwithstanding the minimal number of controlled DOFs, the tasks is rather challenging for at least three reasons. The first is that to reach the cup handle the robot has to generate variable EPs so that the arm follows a curved trajectory: for example, when it starts to move with the hand at the left of the cup, the hand cannot reach the handle along a straight line but has to move around it, similarly to reaching a target with obstacles [13]. The second reason is that the controller has to learn to perform such trajectories on the basis of the rare feedback of the scalar value of reinforcement. This generates difficulties due to the "time and space credit assignment problems" well-known within the reinforcement learning literature [23]. The rare feedback mirrors the conditions in which organisms acquire behaviours by trial-and-error. The third reason is that the perception of the system (see Fig. II-B) is rather limited and the controller is informed only on the kinematics (joint angles) but not on the dynamics of the arm (e.g., changes of joint angles, hand velocity, etc.). This implies that each perceived posture might correspond to several different speeds and movement directions of the arm. In terms of the RL framework, the task involves a *Partially Observable Markov Decision Problem* (POMDP). POMDPs are often computationally intractable to be exactly solved [24]. Velocity was not furnished to the system to keep the whole model as simple as possible to the extent that this did not impair the target results.

The robot is formed by three components: a visual system, a 3D dynamic arm-hand, and a muscle system. These components are now explained in detail.

1) Visual System: The visual system is composed of an "eye" (an RGB camera with a resolution of 630×630 pixels covering a $120°$ pan angle field and a $120°$ tilt angle field) mounted $25cm$ above the shoulder and "leaning forward" $10cm$. The eye movement is controlled by a hard-wired reflex that tends to foveate blue objects (the object to be reached). In particular, the pan and tilt angles, encoded in the vector $\mathbf{pt}$, are changed as follows:

$$\Delta\mathbf{pt} = \frac{(\mathbf{T} - \mathbf{C})}{630} 120° \qquad (1)$$

where $\mathbf{C}$ is a vector with two elements equal to 315, corresponding to the center of the image in pixels, $\mathbf{T}$ is a vector whose two elements are equal to the weighted average of the x-y components of the positions of the target blue pixels in the image (with average weights equal to the activations of the neurons), and 630 and $120°$ are respectively the size of the image in pixels and the size of the two dimensions of the camera field in degrees. Note that due to this reflex the system does not need to learn to guide the eye gaze.

The assumption for which the eye always foveates the target allows the model to use eye proprioception to identify the position of the object relative to the robot body. Although in this work this information is not useful to guide reaching as the object is always set at the same position with respect to the arm, Sect. II-C will show that the system can exploit such information, which does not change during movement, to speed up learning in the initial phases of development. This information will also be used in future work to allow the

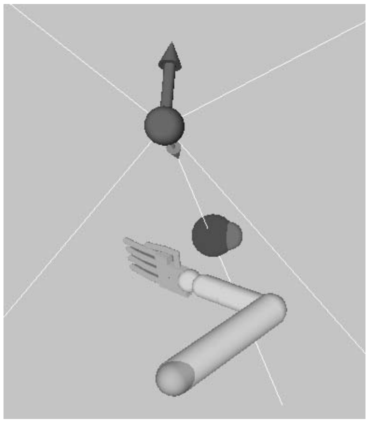

Fig. 1. The robotic setup used to test the model. The sphere with the arrows indicates the position of the eye (thin arrow: current gaze; thick arrow: camera up-vector). The sphere in front of the arm is the cup and the smaller semisphere on its is the cup-handle that the robot has to reach.

system to reach objects located in any position in the working space. The assumption for which the system always foveates the arm target is in line with the current neuroscientific literature suggesting that primates tend to maintain the foveation on the target objects during reaching and that their brain exploits gaze centered reference frames as much as possible to organise sensorimotor coordination (see [1] for a review).

Note that given the used setup, the camera is used in this work only to control the focussing reflex of the camera itself. In turn, this reflex allows the system to compute the gaze direction which is then used as a proprioceptive input for the neural controller. This implies that the image returned by the camera is not used. This image might be useful in future work to allow the system to tackle more complex tasks, for example to visually guide grasping actions.

2) Simulated Dynamic Arm and Hand: The robot simulated arm (2 links/7 DOFs) and hand (21 links/19 DOFs) (see Fig. 1) have the same kinematic and dynamic parameters of the *iCub* robot (http://www.robotcub.org). In particular, the upper arm is a cylinder with radius of $2cm$, length $7.5cm$, and mass $0.575kg$. The forearm is a cylinder with radius of $1.5cm$, length $6.5cm$, and mass $0.625kg$. The wrist is a sphere with radius of $1.5cm$ and mass $0.01kg$. The palm and fingers are formed by shorter and lighter segments. The inertia matrix of each link is computed based on standard formulas for solid bodies.

The arm and hand are simulated with a physical engine software developed at ISTC-CNR. This software is based on the open-source software "OPAL" (Open Physics Abstraction Layer) updated to be able to interface "NEWTON" physical engine (originally OPAL could only be used with the open-source "ODE" physical engine). NEWTON allows the simulation of physical interactions between the solid bodies forming the robot and the environment. The time step used by the physical engine for the integration of the equations was set to $0.01s$.

In the simulations, the hand DOFs are not controlled, that is their EPs are kept at fixed values corresponding to a completely opened hand. Even if not used, the hand is simulated as in the future the model, suitably developed, will be tested with grasping tasks. Moreover, the arm moves on the plane and two of the three DOFs of the shoulder and all the three DOFs of the wrist are kept still (so only one DOF of the shoulder and one DOF of the elbow are controlled, respectively ranging in [0, 180] and [0, 160] degrees). This assumption is in line with the developmental psychology framework within which the model is developed. Experiments with infants have shown that these tend to learn to accomplish the first reaching movements by using only two or three DOFs (cf. [15]). The explanation is that this strategy reduces the complexity of movements and accelerates initial learning [25]. Also, as neurodevelopment proceeds the brain tends to generate intrinsic neural constraints that implement simplified solutions (i.e., using a minimal number of DOFs) to the problem of kinematic redundancy [26]. Here the assumption is also useful to allow the analysis and interpretation of the properties of the system through 2D graphs (see Sect. III-B).

3) Muscle Model: The robotic arm moves on the basis of joint torques generated by simulated muscle models which receive as input the desired arm angles (EPs) from the output of the neural network controller (see Sect. II-B). Similarly to what is done in [15], the model simulates the main properties of muscles, in particular spring-like properties and dumping effects, by using *Proportional Derivative controllers* (PDs) [27] governed by the following equation:

$$\mathbf{T} = \mathbf{K}_P(\mathbf{EP}_n - \mathbf{J}) - \mathbf{K}_D\dot{\mathbf{J}} \tag{2}$$

where $\mathbf{T}$ is the vector of torques applied to the joints, $\mathbf{K}_P$ is a diagonal matrix with elements equal to 800, $(\mathbf{EP}_n - \mathbf{J})$ is the difference vector between the noisy desired equilibrium point issued to the muscles and the current angular joint position, $\mathbf{K}_D$ is a diagonal matrix with elements equal to 40, and $\dot{\mathbf{J}}$ is the vector of current angular speed of joints. As shown in [15], muscle models as simple as the ones used here allow reproducing reaching movements quite accurately.

A further assumption is that the PD action is integrated by a *gravity-compensation mechanism*. This is implemented in a simple fashion by ignoring the effects of gravity on the arm and the hand in the dynamic simulator of the arm-hand. This is a strong simplification as compensation mechanisms are always approximate due to the difficulty of estimating accurate dynamic models. This is a relevant issue for future work if the model will be transferred to real robots. In this respect, interestingly preliminary experiments controlling the 2DOFs arm working on a vertical plane showed that the controller is capable of learning to compensate gravity on the basis of the same "anticipatory properties" illustrated in the next sections.

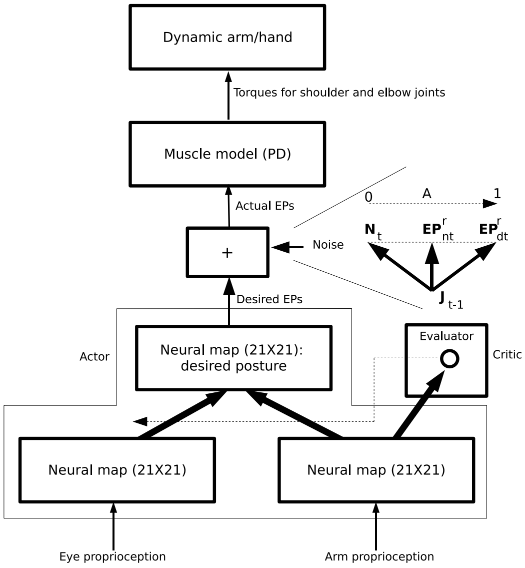

Fig. 2. The architecture of the system. Thin arrows represent information flows, bold arrows represent all-to-all connection weights trained with the RL algorithm. The hatched arrow represents the critic learning signal. The "zoomed" little diagram represents the key elements used to generate the fine-tunable noise used by the RL algorithm, with symbols indicating relevant points in the joint space.

B. The Architecture and Functioning of the Model

The architecture of the system (Fig. 2) is based on a neural implementation of the actor-critic model [23]. From a biological perspective the actor-critic model can be related to basal ganglia brain nuclei and dopamine dynamics [21]. The *actor* is composed of two input 2D neural maps, and one output 2D neural map, each formed by 21×21 neurons. These maps use *population codes* [20] to encode input and output signals similarly to what is done in parietal cortex [28]. The two input maps encode proprioceptive information: respectively the two pan-and-tilt angles of the camera gaze direction, and the two controlled joint angles of the arm. The two maps are activated on the basis of a Gaussian function (maximum value equal to 1 and width equal to the distance between two close units in the neural space) centred on the angles to encode, capturing the assumption that the closer the posture angles to the *preferred angles* of the map units, the higher their activation. The output map of the model, assumed to correspond to motor cortex [18], encodes arm desired angles. The neurons a_j of the actor output map receive the signals from the neurons x_i of the input maps via the connections having weights w_{ji}, and activate on the basis of the positive part of a hyperbolic tangent function. The current desired EP of the arm ($\mathbf{EP}_{dt}$) is *read out* from the population code of the map as a weighed average of the preferred angles of its neurons with weights equal to their activations [20] [17].

Before being sent to muscle models, the desired posture is affected by noise to foster exploration. The technique used here to generate noise has some properties that allows: (a) fine balancing the relative weight of the desired EP and noise; (b) working with continuous actions; (c) tackling the problem for which the inertia of the arm tends to average and cancel out instant noise signals. The design of this technique was triggered by the lack in the literature of a suitable method with such properties (e.g., the method proposed in [29] revealed difficult to be tuned; cf. [30] for an alternative solution). Mathematically, the noisy EP issued to muscles at time t, $\mathbf{EP}_{nt}$, is computed as follows (measure unit expressed in neural space as the distance between two close units):

$$\begin{aligned}
&\mathbf{N}^b_t = (1-\sigma)\cdot\mathbf{N}^n_{t-1} + \sigma\cdot\mathbf{N}^{rand}\\
&\mathbf{N}^n_t = \mathbf{N}^b_t / \left|\left|\mathbf{N}^b_t\right|\right| \quad if \quad 1 < \left|\left|\mathbf{N}^b_t\right|\right| \quad else \quad \mathbf{N}^n_t = \mathbf{N}^b_t\\
&\mathbf{N}_t = \mathbf{N}^n_t \cdot N^{max}\\
&\mathbf{EP}^r_{nt} = A\cdot\mathbf{EP}^r_{dt} + (1-A)\cdot\mathbf{N}_t\\
&\mathbf{EP}_{nt} = \mathbf{EP}^r_{nt} + \mathbf{J}_{t-1}
\end{aligned} \tag{3}$$

where $\mathbf{N}^b_t$ is a buffer vector, σ (set to 0.05) is a parameter which allows progressively updating $\mathbf{N}^b_t$ on the basis of the noise vector $\mathbf{N}^{rand}$ (whose elements are uniformly drawn in [-1, +1]), $\mathbf{N}^n_t$ is a two-element noise vector with size normalised in [0, 1], $\mathbf{N}_t$ is a noise vector with maximum size N^{max} ($N^{max} = 10$) , $\mathbf{EP}^r_{dt}$ is the desired equilibrium point vector produced by the actor but expressed with respect to a reference frame centred on the previous joint angles $\mathbf{J}_{t-1}$, A is a variable changed in [0.1, 0.9], $\mathbf{EP}_{nt}$ is the noisy EP vector issued to the muscle models. The rationale of Eq. 3, sketched in Fig. 2, is as follows. The delay mechanism with which $\mathbf{N}^n_t$ is updated on the basis of $\mathbf{N}^{rand}$ assures that the direction and intensity with which noise "pulls" the arm away form the desired EP changes gradually: this is needed as a white noise without inertia would not allow the arm to explore the environment as the arm physical inertia would average it out. N^{max} allows regulating the maximum exploration range due to such noise. A is the "ability" of the actor which is supposed to increase with learning. In the simulations, A is increased linearly from 0.1 to 0.9 during training so as to progressively increase the exploitation/exploration ration as usually done in RL [23]. In the future, A might be set with a suitable index capturing the actual actor ability.

The *critic* is formed by a neural network (*evaluator*) with a linear output unit producing the evaluation v_t of the currently perceived state, and the computation of the TD-error ("suprise") signal (see below). This network gets as input, via the connection weights w_i, only the signals from the neurons of the arm-proprioception map. The critic uses couples of successive evaluations, together with the reward signal r_t, to compute the *surprise signal* s_t [23]:

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

where γ is a discount factor (γ = 0.99).

C. Reinforcement Learning for Continuous Actions Encoded with Neural Maps

The evaluator weights are trained on the basis of a standard TD(λ) learning rule with "replace eligibility traces" [23]. In particular, at time t the eligibility trace e_{it} of a connection weight w_i is computed on the basis of the signal x_{it} "passing through it", or is set equal to the "decayed" old eligibility e_{it-1} if this is bigger than it. The connection weight is updated according to the old eligibility:

$$e_d = \gamma\lambda e_{it-1} \quad e_{it} = max\left[e_d, x_{it}\right] \quad w_{it} = w_{it-1} + \eta s_t e_{it-1} \tag{5}$$

where e_d is the decayed old eligibility, λ is the decay coefficient of the eligibility ($\lambda = 0.94$), and η is a learning rate ($\eta = 0.06$).

For the training of the actor at time t the eligibility trace e_{jit} of a connection weight w_{ji} is computed on the basis of the signal x_{it} "passing through it", or is set equal to the "decayed" old eligibility e_{jit-1} if this is bigger (in absolute value). The connection weight is updated according to the old eligibility:

$$e_d = \gamma\lambda e_{jit-1}, e_{ps} = (a_{jEPnt} - a_{jt})x_{it}, e_{ns} = a_{jEPnt}a_{jt}x_{it}$$

$$e_{jit} = \begin{cases} e_d & if\ s_t \geq 0\ and\ |e_d| \geq |e_{ps}| \\ e_{ps} & if\ s_t \geq 0\ and\ |e_d| < |e_{ps}| \\ e_d & if\ s_t < 0\ and\ |e_d| \geq |e_{ns}| \\ e_{ns} & if\ s_t < 0\ and\ |e_d| < |e_{ns}| \end{cases}$$

$$w_{jit} = w_{jit-1} + \eta s_t e_{jit-1} \tag{6}$$

where e_d is the decayed old eligibility, e_{ps} is the new eligibility with positive surprise, e_{ns} is the new eligibility with negative surprise, and a_{jEPnt} is a pseudo-activation of neuron a_j computed on the basis of a Gaussian function (with height and width equal to one) of $||\mathbf{EP}_{nt} - \mathbf{EP}_j||$, that is the distance between the EP angles sent to the arm and the preferred angles of the neuron j. The rationale of the formula is that when current surprise s_t is positive using e_{ps} to update the weights implies that they change so that a_{jt-1} gets progressively closer to $a_{jEPnt-1}$: in this way the actor action $\mathbf{EP}_{dt-1}$ gets closer to the noisy actually pursued action $\mathbf{EP}_{nt-1}$. When current surprise is negative, using e_{ns} to update the weights implies that they change so that a_{jt-1} gets progressively closer to zero (but only if $a_{jEPnt-1}$ is above zero, that is in correspondence to the actual EP sent to muscles): in this way the actor action gets a lower probability of being selected.

III. Results

A. Learning to Reach: From Fixed EPs to Variable EPs

The model was trained for 400,000 simulation cycles with the evaluator connection weights set to 0 (implying a 0 initial evaluation) and the actor connection weights set to 0.001 (implying an initial desired posture in a central position within the joint space). Each trial started with a random posture of the arm and the eye gaze direction set at the centre of the working plane (south of the object), and terminated either when the hand touched the cup handle or when 600 cycles elapsed.

Fig. 3a shows the average time spent by the system to reach the target in 64 trials with different conditions of the connection weights. The data show that with untrained connection weights (i.e. by performing random movements) the system takes $1.79s$ on average to reach the target, whereas after the whole training process it takes $0.47s$ on average. Notice how the rather good performance of the untrained-weight condition indicates that the exploration generated by the noise mechanism is rather good. The figure also reports the results of the same test after the system has reached only three times the target. In this case the performance, equal to $1.40s$, is higher than the performance of the untrained-weight condition ($1.79s$). This indicates that the use of the EPs for control allows the system to quickly find an association between the object position in space (signalled by the eye gaze direction) and the posture corresponding to the hand on the target. This "scaffolds" the following development as it allows the system to perform gross reaching movements that lead the hand in proximity of the target (Fig. 3b) and to learn the EPs which produce suitable curved trajectories as those produced by FCMs.

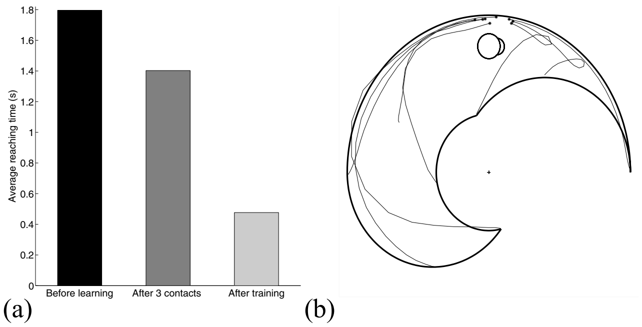

Fig. 3. (a) Average time the system takes to perform 64 reaching movements with the initial positions set on the 8×8 vertexes of a regular grid overlapped with the joint space. The bars of the histogram refer to the outcome of the test run in various conditions after having set to zero the learning rate of RL: before learning; after three successful contacts with the target; after the whole training. (b) Different trajectories (lines) and final ring finger tip positions (circles) after the initial learning (after three contacts with the target) for 9 different starting positions of the hand set on the 3×3 vertexes of a regular grid overlapped with the joint space. The white circle represents the cup and the semi-circle the cup handle. The bold line represents the edge of the working space. Notice that sometimes the trajectories followed by the ring finger cross the boundaries of the working space as the fingers might move away from their equilibrium points (set to fixed values which keep the hand in a straight position) due to inertia forces.

B. Learning to Produce Curved Trajectories

With prolonged training the model develops a good capacity to produce curved trajectories to the handle. Fig. 4 shows that only in 13 trials out of 64 the arm hits the cup before touching the handle. The explanation of the outcome of these 13 trials is that the system can only optimise behaviour while lacking information on the arm velocity. This implies that the best thing RL can do is to seek the best action for each posture which leads, *on average*, to the best consequences corresponding to the various possible dynamical states of the arm for such posture. So, when the hand initial position is far from the target (e.g. in the south-west quadrant of the working space) the arm gains a high speed, collides with the cup, and then performs a movement correction.

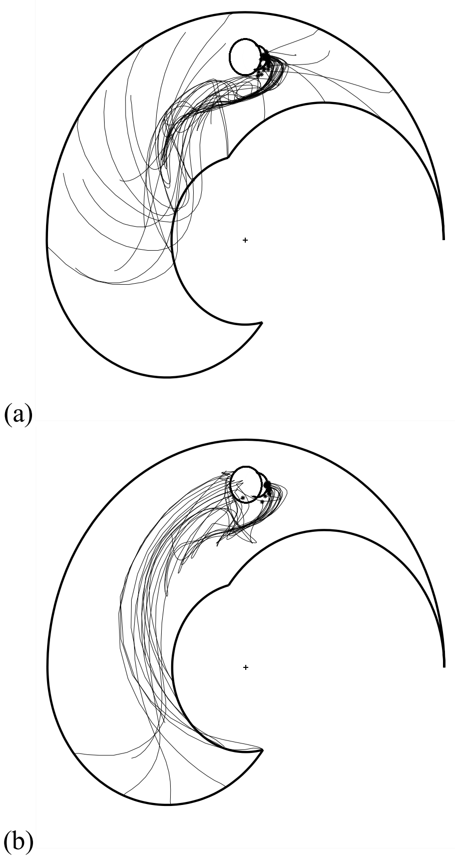

Fig. 4. Trajectories followed by the ring finger tip in 64 trials starting with different hand positions. (a) Trajectories followed in the 51 trials when the hand reaches the handle without colliding the cup. (b) Trajectories followed in the 13 trials when the hand hits the cup before touching the handle.

It is interesting to analyse which variable EPs the system selects to produce the curved trajectories. The EPs selected to perform the 64 movements are indicated with single dots in Fig. 5a. Fig. 5b, obtained by "smoothing" the dots of all trajectories, and direct inspection of the EPs during movement (cf. Fig. 5c), indicate that the EPs tend to group in three clusters: (a) a first cluster (darkest area in Fig. 5b) is used by the system for moving when the hand is situated at the south-west and north-west quadrants of the working space. These EPs allow the system to move towards the cup when the hand is at positions far from it; (b) a second cluster, positioned at south of and close to the cup, is used by the system to avoid colliding with it when the hand is in positions close to it (with the exception of the aforementioned cases when the arm has a high momentum); (c) a third cluster, positioned at north of the the first cluster, is used by the system to "close" the movement on the cup when the hand is at the right of the cup itself (this cluster has a low density of dots, in comparison to the other two clusters, as it is formed by few highly-dynamical EPs before trial termination, see Fig. 5c); Sect. III-D will show that the EPs so generated have interesting anticipatory properties.

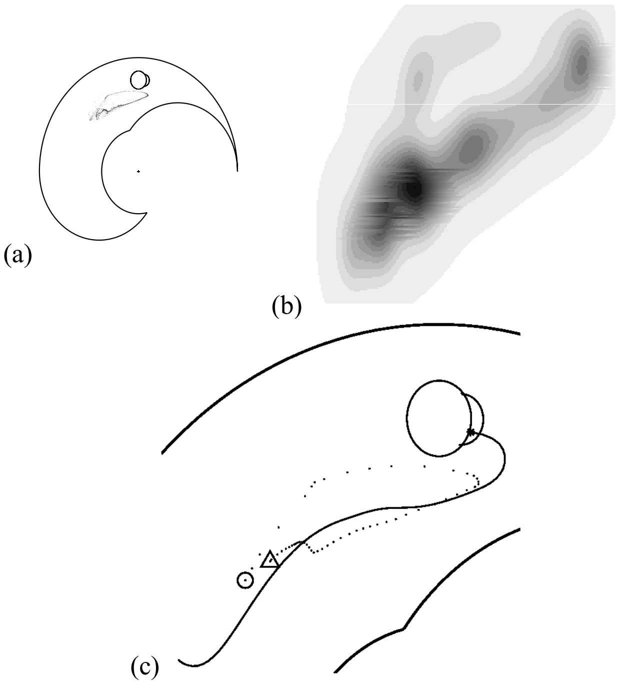

Fig. 5. (a) EPs (represented with a dot corresponding to the x-y position of the ring finger tip) selected by the actor during the execution of the 64 movements indicated in Fig. 4. (b) Density of EPs in the critical portion9 of the working space represented in graph "a". (c) A specific trajectory followed by the arm (continuous line) when the actor selects a particular sequence of EPs (triangle: initial EP; circle: final EP).

Another interesting result related to the system performance is that it learns to optimise trajectories in terms of the part of the hand with which it touches the handle. In this respect, Fig. 6 shows that at the end of the training the arm touches the object with the *tip* of the fingers. In this way the time needed to reach the target is reduced, in particular when the system has to "move around" the cup to reach the handle (recall that reinforcement learning systems automatically tend to reduce the time of the task solutions found due to the fact that they attempt to receive rewards as soon as possible as this reduces the cumulated discounting of the rewards themselves).

C. The Generation of Fine-Tunable Noise in the Neural Space

Fig. 7 shows how the noise mechanism illustrated in Sect. II-B allows to fine-tune the exploration/exploitation balance in the posture space. The graphs of the figure show the postures explored by the the arm when the actor output units are clumped to fixed values corresponding to a desired EP of 90 degrees for both the shoulder and elbow joint angles, and the ability coefficient A is set to 0.1, 0.3, 0.6, and 0.9.

The graphs show that with low levels of ability ($A = 0.1$), corresponding to the initial phases of learning, the arm explores the whole posture space. When ability gradually increases ($A = 0.3, 0.6$), exploration focusses around the

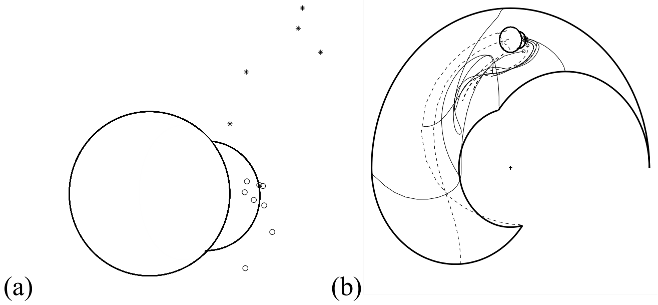

Fig. 6. (a) Ring finger tip final positions when the hand touches the cup handle starting to move from nine different initial positions on vertexes of a 3 × 3 grid overlapped to the working space (trials where the hand hit the cup were discarded). Dots indicate the final tip positions in five trials run before training (in this case training was off) whereas circles indicate the final tip positions in seven trials run after training. (b) Trajectories in the case of trained weights of "a" (dashed lines: trajectories which hit the cup).

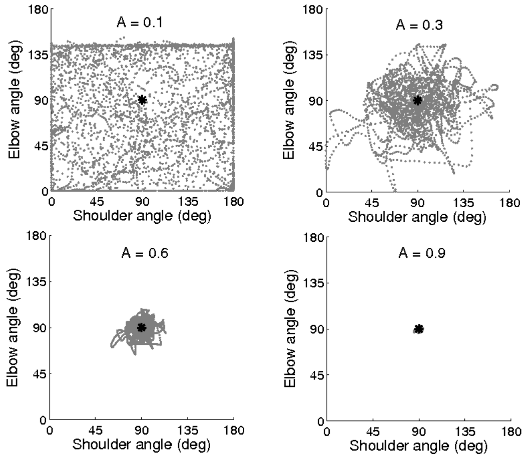

Fig. 7. Exploration of the posture space at various levels of ability A: ring finger tip positions (dots) in 3000 cycles. Dots in the graphs represent the fixed desired arm posture used in the tests, points represent positions of the ring finger tip.

desired EP. This allows the system to refine the posture associated to the arm state perceived through proprioception. When ability reaches very high levels ($A = 0.9$) noise has little effects on movement and the arm can fully exploit the acquired knowledge by performing stable movements.

D. Emergence of Anticipatory Behaviour

Interestingly, direct inspection of the dynamics of the EPs found by the model indicates that they acquired clear anticipatory properties taking into consideration the arm dynamic properties. For example, when the initial posture is set to 90 degrees for both the shoulder and elbow joints, the EPs first move in the second cluster (the one below the cup, see Sect. III-B), so "pulling" the hand towards the right hand side, and then, *before* the arm reaches such position, move in the opposite direction in the third cluster, so first decreasing the speed of the hand which is moving towards the right hand side due to inertia and then "closing" the arm on the object.

Fig. 8 illustrates this quantitatively by reporting the value of the joint angles of the desired EP and the actual joint angles of the arm during such movement. The figure shows that the desired shoulder angle inverts the direction of change just before $0.5s$ whereas the actual shoulder angle follows such inversion about $0.1s$ later. Similarly, the desired elbow angle "closes" on the objects just before $0.4s$ whereas the actual elbow angle performs a similar closure about $0.06s$ later.

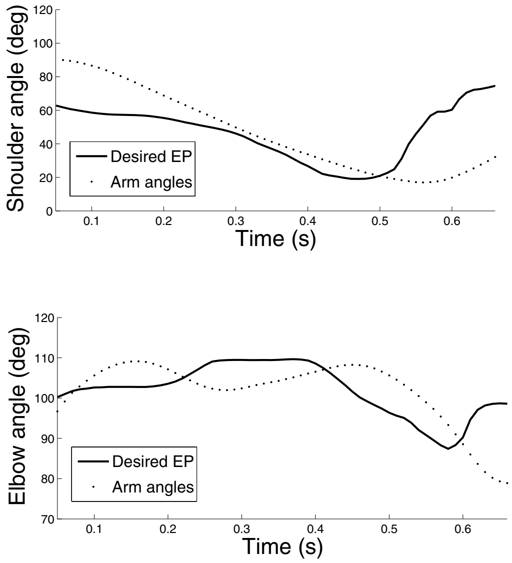

Fig. 8. The EP supplied by neural network (solid line) and the effective joint position for shoulder (top) and elbow (down) during a reaching movement.

IV. Conclusions and Future Work

This work presented a model which integrates some of the advantages of models based on the equilibrium-point (EP) hypothesis and models based on force-based control. In particular, the model controls an arm on the basis of EPs but then uses a reinforcement learning (RL) algorithm to search and set (*at each step*) EPs which guide the arm along complex trajectories. Remarkably, these EPs take into consideration the dynamic aspects of the plant in an anticipatory fashion similar to what is often done by force-based controllers on the basis of forward models. To the best of the authors' knowledge, the model proposed here is the first to use RL as a means to find EPs which generate sophisticated trajectories and which are anticipatory with respect to the dynamics of the controlled plant. The RL algorithm used is also novel in that it is applicable to continuous actions and dynamic systems without the use of pattern generators but on the basis of the particular use of EPs and the specific features of the noise-generation mechanism (for other models tackling these problems see [29], [30]).

The model is also a valuable tool to investigate developmental phenomena, e.g. related to the onset of reaching [31], much in the spirit of [15]. In this respect, for example, the results presented here show that the model tends to produce trajectories on the basis of *clusters* of EPs and this is reminiscent of the possible organisation of children motor behaviour on the basis of the composition of "sub-movements" [32] [25], and in general of organisms motor behaviour on the basis of "motor-primitives" [8]. This topic might deserve further investigations.

One important aspect of the model is that, although it is capable of taking into consideration the dynamical aspects of the controlled plant, it does not control joint stiffness but only (indirectly) joint forces by suitably regulating the desired EPs with respect to the current joint position. In contrast to this, it has been shown that in real organisms stiffness is actively controlled to compensate for disturbances which can potentially cause instabilities [33]. Moreover, a high stiffness is exploited to have stability in the initial phases of learning, when the problem and disturbances are not yet known, whereas a finer control of forces is used with the advancement of learning [34]. These aspects of motor control will be investigated in future work by letting the reinforcement learning model to directly control the gain coefficient $\mathbf{K}_P$ of the PD muscle model (cf. equation 2) aside the desired EPs.

References

[1] R. Shadmehr and S. P. Wise, Eds., *The Computational Neurobiology of Reaching and Pointing*. Cambridge, MA: The MIT Press, 2005.

[2] M. Kawato, "Internal models for motor control and trajectory planning," *Curr Opin Neurobiol*, vol. 9, pp. 718–727, 1999.

[3] N. Hogan, "Mechanical impedance of single- and multi- articular systems," in *Multiple muscle systems. Biomechanics and movement organization*, W. S.-Y. Winters, J. M., Ed. Springer, New York, 1990, pp. 149–164.

[4] F. Nori, G. Sandini, and J. Konczak, "Can imprecise internal motor models explain the ataxic hand trajectories during reaching in young infants?" in *Proceedings of the Ninth International Conference on Epigenetic Robotics (EpiRob2009)*, ser. Lund University Cognitive Studies, L. Canamero, P.-Y. Oudeyer, and C. Balkenius, Eds. Lund: Lund University, 2009, no. 146, pp. 215–216.

[5] M. I. Jordan and D. E. Rumelhart, "Forward models: supervised learning with a distal teacher," *Cognitive Sci*, vol. 16, pp. 307–354, 1992.

[6] N. Bhushan and R. Shadmehr, "Computational nature of human adaptive control during learning of reaching movements in force fields," *Biol Cybern*, vol. 81, pp. 39–60, 1999.

[7] A. G. Feldman, "Once more on the equilibrium-point hypothesis (lambda model) for motor control." *J Motor Behav*, vol. 18, pp. 17–54, 1986.

[8] E. Bizzi, N. Hogan, F. A. Mussa-Ivaldi, and S. Giszter, "Does the nervous system use equilibrium-point control to guide single and multiple joint movements," *Behav Brain Sci*, vol. 15, pp. 603–613, 1992.

[9] J. Won and N. Hogan, "Stability properties of human reaching movements," *Exp Brain Res*, vol. 107, pp. 125–136, 1995.

[10] J. F. Pilon, S. J. De Serres, and A. G. Feldman, "Threshold position control of arm movement with anticipatory increase in grip force," *Exp Brain Res*, vol. 181, pp. 49–67, 2007.

[11] R. Balasubramaniam and A. G. Feldman, "Guiding movements without redundancy problems," in *Coordination Dynamics*, V. K. Jirsa and J. A. S. Kelso, Eds. New York: Springer, 2004.

[12] H. Gomi and M. Kawato, "Equilibrium-point control hypothesis examined by measured arm stiffness during multijoint movement," *Science*, vol. 272, pp. 117–120, 1996.

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

[14] J. Peters and S. Schaal, "Reinforcement learning for parameterized motor primitives," in *IJCNN*. IEEE, 2006, pp. 73–80.

[15] N. E. Berthier, M. T. Rosenstein, and A. G. Barto, "Approximate optimal control as a model for motor learning," *Psychol Rev*, vol. 112, pp. 329–346, 2005.

[16] P. Cesari, T. Shiratori, P. Olivato, and M. Duarte, "Analysis of kinematically redundant reaching movements using the equilibrium-point hypothesis," *Biol Cybern*, vol. 84, pp. 217–226, 2001.

[17] D. Caligiore, D. Parisi, and G. Baldassarre, "Toward an integrated biomimetic model of reaching," in *Proc. of ICDL 2007*, Y. Demiris, B. Scassellati, and D. Mareschal, Eds. London: Imperial College, 2007, pp. E1–6.

[18] T. Aflalo and M. Graziano, "Partial tuning of motor cortex neurons to final posture in a free moving paradigm," *PNAS*, vol. 103, pp. 2909–2914, 2006.

[19] J. Flanagan, D. Ostry, and A. Feldman, "Control of trajectory modifications in target-directed reaching." *J Motor Behav*, vol. 25, pp. 140–152, 1993.

[20] A. Pouget and P. E. Latham, "Population codes," in *The Handbook of Brain Theory and Neural Networks*, 2nd ed., M. A. Arbib, Ed. Cambridge, MA, USA: The MIT Press, 2003.

[21] A. G. Barto, "Adaptive critics and the basal ganglia," in *Models of Information Processing in the Basal Ganglia*, J. Houk, J. Davis, and D. Beiser, Eds. Cambridge MA, USA: The MIT Press, 1995, pp. 215–232.

[22] M. Butz, O. Sigaud, G. Pezzulo, and G. Baldassarre, Eds., *Anticipatory behavior in adaptive learning systems: from brains to individual and social behavior*, ser. LNAI4520. Berlin: Springer-Verlag, 2007.

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

[24] L. Kaelbling, M. Littman, and A. Cassandra, "Planning and acting in partially observable stochastic domains," *Journal of Artificial Intelligence*, vol. 101, pp. 99–134, 1998.

[25] N. E. Berthier, R. K. Clifton, D. D. McCall, and D. J. Robin, "Proximodistal structure of early reaching in human infants." *Exp Brain Res*, vol. 127, pp. 259–269, 1999.

[26] D. Campolo, D. Accoto, F. Taffoni, and E. Guglielmelli, "On the kinematics of human wrist during pointing tasks with application to motor rehabilitation," in *IEEE International Conference on Robotics and Automation (ICRA 2008)*. IEEE, 2008, pp. 1318–1323, pasadena, California, May 19-23.

[27] D. Bullock and S. Grossberg, "Vite and flete: neural modules for trajectory formation and postural control." in *Volitional Action*, W. Hershberger, Ed. Amsterdam: Elsevier., 1989, pp. 253–298.

[28] G. Bosco, R. E. Poppele, and J. Eian, "Reference frames for spinal proprioception limb endpoint based or joint-level based," *J Neurophysiol*, vol. 83, pp. 2931–2945, 2000.

[29] K. Doya, "Reinforcement learning in continuous time and space." *Neural Comput*, vol. 12, no. 1, pp. 219–245, Jan 2000.

[30] J. Peters, S. Vijayakumar, and S. Schaal, "Reinforcement learning for humanoid robotics," in *Third IEEE-RAS International Conference on Humanoid Robots (Humanoids2003)*, 2003, karlsruhe, Germany, Sept.29-30.

[31] J. Konczak and J. Dichgans, "The development toward stereotypic arm kinematics during reaching in the first 3 years of life," *Experimental Brain Research*, vol. 117, no. 2, pp. 346–354, 1997.

[32] J. Kuhtz-Buschbeck, H. Stolze, K. J
"ohnk, A. Boczek-Funcke, and M. Illert, "Development of prehension movements in children: a kinematic study," *Experimental Brain Research*, vol. 122, no. 4, pp. 424–432, 1998.

[33] E. Burdet, R. Osu, D. W. Franklin, T. E. Milner, and M. Kawato, "The central nervous system stabilizes unstable dynamics by learning optimal impedance." *Nature*, vol. 414, pp. 446–449, 2001.

[34] R. Osu, D. W. Franklin, H. Kato, H. Gomi, K. Domen, T. Yoshioka, and M. Kawato, "Short- and long-term changes in joint co-contraction associated with motor learning as revealed from surface emg." *J Neurophysiol*, vol. 88, no. 2, pp. 991–1004, Aug 2002.