Biologically inspired sensory motor control of a 2-link robotic arm actuated by McKibben muscles

Sofiane Ouanezar, Frédéric Jean, Bertrand Tondu, Marc A. Maier, Christian Darlot
and Selim Eskiizmirliler

Abstract— This study focuses on biomimetic sensory motor control of a robotic arm. We have developed a command circuit that was mathematically deduced from physical and mathematical constraints describing the function of cerebellar pathways. The control circuit contains an internal predictive model of the direct biomechanical function of the limb placed in a closed loop, so that the circuit computes an approximate inverse function. The structure of the model resembles the anatomic connectivity of the cerebellar pathways. In this paper, we present an application of this model to the control of a 2-link robotic arm actuated by four single-joint McKibben muscles and report the results obtained by simulation and real-time learning of 2 degrees of freedom pointing movements.

I. INTRODUCTION

Studying movement control models may allow for a better understanding of human sensorimotor control systems, and may provide a systematic way for designing devices that drive robots. Reciprocally, analysis of robot control systems may help understanding the relationship between neural structure and function in human sensorimotor control, and may help evaluating the plausibility of the underlying control model. The cerebellum is known to be involved in the control and learning of smooth, fast and fine-tuned movements. Since biological feedback loops are too slow to drive fast movements, they cannot be regulated in a closed loop by the use of sensory signals, but must be driven in an open loop mode by motor commands taking into account the dynamical and geometrical characteristics of the limb to be moved. One hypothesis is that the Cerebellum computes inverse dynamics solutions. However the calculation of the inverse model is an ill-posed problem for two major reasons. First, a direct function is deterministic, whereas an inverse function is not. Secondly, an inverse problem does not have a general solution. In local space, various studies corroborate the hypothesis that the cerebellum calculates approximate inverse functions of limb biomechanics [1], [2], [3], [4], [5], [6], [7], [8], [9].

The second hypothesis proposes that the cerebellum computes sensorimotor predictions [10], [11], [12] using internal forward models, i.e. the direct functions of limb biomechanics. These two opposing views can be unified in a single model (see detail in previous papers [13], [14], [15], [16], [17], [18]). Our approach is based on mathematical and physical principles, which determine fast open-loop movements. We suppose that the principal problem to be solved is the computation of an approximate inverse function of the biomechanics of the arm.

In order to obtain a correct movement, the control signal sent by the motoneurons to the muscles must be "tuned" such that the target $\theta^D$ and accomplished $\theta^A$ movements are equal. A schematic representation of this constraint is given in Fig.1 where $H$ represents the biomechanical function of the limb and $H^{-1}$ its inverse function (Fig.1-A).

The explicit computation of the ill-posed inverse function can be easily by-passed by placing 2 parallel loops in an internal feedback loop (Fig.1-B): a positive loop which transmits the signal without any change and a negative loop which passes through the internal forward model of the direct biophysical function of the limb $H$ denoted by $H^*$ in order to predict the movement resulting from the motor control.

In our view, the component $H^*(\alpha)$ represents the cerebellar cortex which receives, via mossy fibers, a large amount of sensory signals and also receives efferent copies of motor signals, and then treats both signals to produce the Purkinje cell activity. This activity, noted $P$ (in Fig.2), codes for a dynamic signal which anticipates the movement resulting from motor commands after the transmission delay (not presented

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig1.png}
\caption{(A) The model satisfying the imposed constraint $\theta^D = \theta^A$ by using the inverse function $H^{-1}$. (B) The proposed model simply using the estimation of the direct biomechanical function to predict its approximate inverse.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig2.png}
\caption{(A) The model satisfying the imposed constraint $\theta^D = \theta^A$ by using the inverse function $H^{-1}$. (B) The proposed model simply using the estimation of the direct biomechanical function to predict its approximate inverse.}
\end{figure}
in the figure) in the motor pathways. Immediately placed downstream $H^*(\alpha)$ (in Fig.2) there is a summing element, where the positive loop is added to the $H^*(\alpha)$ output signal $P$, producing the Q signal. The summing element represents a group of neurons of the cerebellar nucleus. Note that the same feedback signal reaches the predictor and the summing element, as is the case for mossy fiber activity that reaches the cerebellar cortex and the cerebellar nuclei.

According to this theory, anticipation of sensory and motor signals would be the main function of the cerebellar cortex, whereas the computation of an approximate inverse function would be the role of the entire cerebellar circuit. In the following we will focus on the application of this approach to the control of a two-link arm performing pointing movements in the horizontal plane.

## II. MATERIAL AND METHODS

Our experimental apparatus is a 2 degrees of freedom (dof) robotic arm actuated by four artificial McKibben muscles, a simplified replication of the human arm (Fig.3). Two antagonistic McKibben muscles drive each joint. They are connected by means of a chain driving a sprocket, and the difference between their forces produces a torque. Pressures are independently set by servo-valves driven by a PC equipped with a digital/analog converter. The joint angles $\theta_1$ and $\theta_2$ are measured by the potentiometers mounted on the sprocket rotation axis. The robotic arm is also equipped with force sensors mounted on each muscle tendon.

### A. Horizontal planar arm model

The dynamics equation of a planar two-link robotic arm is described by the well known Euler-Lagrange equation:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} = \tau(\theta, \dot{\theta}, p)$$  \hspace{1cm} (1)

where $I$ is the moment of inertia around the joint, $L$ is the link length, $L_g$ is the gravity center of each link, $M_i$ is the link mass, $C(\theta, \dot{\theta})$ is the matrix of coriolis forces and interaction forces:

$$C(\theta, \dot{\theta}) = M_2 L_2 L_{g2} \sin \theta_2 \begin{pmatrix} -2\dot{\theta}_2 & -\dot{\theta}_2 \\ \dot{\theta}_1 & 0 \end{pmatrix}$$  \hspace{1cm} (2)

and $\tau(\theta, \dot{\theta}, p)$ is the joint torque generated by antagonist muscles, where $p$ designates the vector of the control pressures. We will note by $\tau_{i}, i=1,2$ the two components of the vector $\tau$ related to the forces $F_{i1}$ and $F_{i2}$ generated by the actuator muscles according to the actuator working principle depicted in figure 4 (see [19] for details).

### TABLE I. ARM PARAMETERS

<table>
<thead>
<tr>
<th>Link 1</th>
<th>Link 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(Kg)$</td>
<td>$L(m)$</td>
</tr>
<tr>
<td>4.58</td>
<td>0.4</td>
</tr>
<tr>
<td>0.67</td>
<td>0.6</td>
</tr>
</tbody>
</table>

In Eq.1 $\theta = [\theta_1, \theta_2]^T$ is the joint angle vector, $\dot{\theta} = d\theta/dt$, $\ddot{\theta} = d\dot{\theta}/dt^2$. $M(\theta)$ is the inertial matrix:

$$M(\theta) = \begin{pmatrix} I_1 + I_2 + M_2 L_1^2 & I_2 \\ +2M_2 L_1 L_{g2} \cos \theta_2 & +M_2 L_1^2 L_{g2} \cos \theta_2 \\ I_2 + M_2 L_1^2 L_{g2} \cos \theta_2 & I_2 \end{pmatrix}$$  \hspace{1cm} (2)

For pulley radius $R_i$ of actuator $i$, we get, for $i = 1, 2$ :

$$\tau_i = R_i \cdot (F_{i1} - F_{i2})$$  \hspace{1cm} (4)

and hence the general expression linking the two-component torque vector $\tau$ to the four-component vector $F$, where $\varepsilon$ designates the muscle linear contraction vector (see further), becomes:

$$\tau(\theta, \dot{\theta}, u) = A \cdot F(\varepsilon, \dot{\varepsilon}, p)$$  \hspace{1cm} (5)

![Fig. 2. A circuit allowing for an anatomical interpretation which is functionally equivalent to the general circuit shown in figure 1](image)

![Fig. 3. (A) Schematic representation of the 2 dof robotic arm (B) Connectivity of four muscles providing an antagonist activation scheme for two robot segments](image)

![Fig. 4. Working principle for antagonist muscle actuators. (a) Initial state, (b) Activated state.](image)
Where $A$ describes the moment arms for every muscle.

$$A = \begin{pmatrix} R_1 & -R_1 & 0 & 0 \\ 0 & 0 & R_2 & -R_2 \end{pmatrix} \quad (6)$$

With $R_i$ = the muscles moment arm.

B. Modelling the muscle

The McKibben artificial muscle is a pneumatic actuator whose properties include a very high force to weight ratio. Such an artificial muscle consisting of a braided sheath, according to a double-helix weaving, surrounding a rubber inner tube is characterized by its initial length $l_0$, initial radius $r_0$, and initial braid angle $\alpha_0$ which is defined as the angle between the muscle axis and each thread of the braided sheath before expansion. When the inner tube is pressurized the muscle inflates, its radius increases, its braid angle changes and it contracts (Fig.4). The muscle can be modelled as a spring and a parallel damper. The force of the muscle $F_M$ is correlated to the applied air pressure $p$, the contraction $\varepsilon$ and the derivative of the contraction $\dot{\varepsilon}$ of the muscle [19].

$$F_M(\varepsilon, \dot{\varepsilon}, p) = F_{spring}(\varepsilon, p) + F_{damper}(\dot{\varepsilon}, p) \quad (7)$$

The force $F_{spring}$ can be expressed as a function of the total air pressure $p$ and the contraction ratio $\varepsilon$:

$$F_{spring}(\varepsilon, p) = (\pi r_0^2)p(\alpha(1 - k\varepsilon)^2 - b), 0 \leq \varepsilon \leq \varepsilon_{max}$$
$$\varepsilon = (l_0 - l)/l_0, a = 3/tan^2(\alpha_0), b = 1/sin^2(\alpha_0)$$

Tondu and Lopez proposed a linear approximation of this static force model between the initial state ($\varepsilon = 0, F_{spring} = F_{max}$) and the final state ($\varepsilon = \varepsilon_{max}, F_{spring} = 0$) in the following form:

$$F_{spring}(\varepsilon, p) = F_{max}(1 - \varepsilon/\varepsilon_{max})$$
$$\varepsilon_{max} = 1 - \sqrt{b/a}$$

$$F_{max} = (\pi r_0^2)p[a - b]$$

for $\varepsilon = 0$ \quad (9)

The force $F_{damper}$ is described by:

$$F_{damper} = -B\varepsilon/dt \quad (10)$$

where $B$ is the muscle viscosity.

The parameters of the McKibben muscles used in this study are summarized in Table II.

<table>
<thead>
<tr>
<th>Muscles</th>
<th>$l_0[cm]$</th>
<th>$r_0[cm]$</th>
<th>$\alpha_0[deg]$</th>
<th>$\varepsilon_0$</th>
<th>$\varepsilon_{max}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Muscle 1 &amp; 2</td>
<td>0.23</td>
<td>1.4</td>
<td>23</td>
<td>0.15</td>
<td>0.3</td>
</tr>
<tr>
<td>Muscle 3 &amp; 4</td>
<td>0.3</td>
<td>0.7</td>
<td>23</td>
<td>0.1</td>
<td>0.3</td>
</tr>
</tbody>
</table>

C. Controlling the robotic arm

The robotic arm shown in Fig.3 is used to replace the peripheral element denoted by $H$ in the figures 1 and 2 in the aim of extending the validation of our cerebellar-like control model to 2 dof. Previous application of this model was limited to 1 dof [15]. The control strategy we will describe in the following is composed of two steps:

- the trajectory planning,
- and the motor control.

1) Trajectory Planning: The aim of trajectory planning is to enable a robotic arm to move from an initial position to a target position. For an arm with 2 dof this corresponds to a simple pointing task with imposed joint angles. However, it has been observed that human pointing movements are characterized by smooth trajectories. Thus, the generation of smooth trajectories is often realized by means of an optimization process for defined cost functions. Rosenbaum [20] discussed the coordinated movements of arm and trunk using optimization criteria defined in the joint space. A minimum jerk model is one possible example of optimization and thus of the trajectory planning.

$$\min \left( \int_0^T \left( \frac{d^3\theta_1}{dt^3} \right)^2 + \left( \frac{d^3\theta_2}{dt^3} \right)^2 \right) dt \quad (11)$$

The solution of minimization can be obtained by an optimal control strategy which can be stated as follows:

Given the system

$$X = f(X, t) = \begin{pmatrix} d\theta_1/dt = \dot{\theta}_1 \\
\theta_1/dt = \theta_1 \\
\theta_2/dt = \theta_2 \\
\theta_2/dt = u_2 \end{pmatrix} \quad (12)$$

where the vector $[\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2]^T$ represents the state $X$ of the system find the optimal control $(u_1, u_2)$ which minimizes the cost function (Eq.11) among the possible trajectories (Eq.12). The solution of this minimization problem is an application of the minimum principle of Pontryagin [21].

The Hamiltonian is defined as:

$$H(x, t) = q_{11}\dot{\theta}_1 + q_{12}\dot{\theta}_1 + q_{13}u_1 + q_{21}\dot{\theta}_2 + q_{22}\dot{\theta}_2 + q_{23}u_2 - 1/2(u_1^2 + u_2^2) \quad (13)$$

$Q_1 = [q_{11}, q_{12}, q_{13}]^T$ and $Q_2 = [q_{21}, q_{22}, q_{23}]^T$ are the vectors of adjoint variables. The necessary conditions for the existence of a minimum are:

$$\begin{pmatrix} q_{11} = 0 \\
q_{12} = -q_{11} \\
q_{13} = -q_{12} \end{pmatrix} \quad (14)$$

With $Q_1$ = [q_{21}, q_{22}, q_{23}]^T and $Q_2$ = [q_{11}, q_{12}, q_{13}]^T.
and the necessary conditions on the control variables are:

\[
\begin{align*}
\frac{dH}{du_1} &= q_{13} - u_1 = 0 \\
\frac{dH}{du_2} &= q_{23} - u_2 = 0
\end{align*}
\]

(15)

so:

\[
\begin{align*}
\theta^{(3)}_1 &= q_{13} \iff \theta^{(4)}_1 = q_{13} = -q_{12} \iff \theta^{(5)}_1 = q_{11} \iff \theta^{(6)}_1 = 0 \\
\theta^{(3)}_2 &= q_{23} \iff \theta^{(4)}_2 = q_{23} = -q_{22} \iff \theta^{(5)}_2 = q_{21} \iff \theta^{(6)}_2 = 0
\end{align*}
\]

(16)

The general solution defining a trajectory is a fifth order polynomial:

\[
\begin{align*}
\theta_1(t) &= a_{10} + a_{11} t + a_{12} t^2 + a_{13} t^3 + a_{14} t^4 + a_{15} t^5 \\
\theta_2(t) &= a_{20} + a_{21} t + a_{22} t^2 + a_{23} t^3 + a_{24} t^4 + a_{25} t^5
\end{align*}
\]

(17)

The (Eq.17) allows joint velocities to follow a bell-shaped smooth velocity profile as shown in Figure 5.

Fig. 5. Position and velocity profiles.

2) Motor control: The physical model of the 2 dof robotic arm described in (Eq.2 and 3) can be expressed as follows:

\[
M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} = \tau(\theta, \dot{\theta}, p)
\]

\[
\begin{pmatrix}
G_1(\theta, \dot{\theta}, \ddot{\theta}) \\
G_2(\theta, \dot{\theta}, \ddot{\theta})
\end{pmatrix} = A \cdot 
\begin{pmatrix}
g_{11}(\theta, \dot{\theta}, \ddot{\theta}, p_{11}) \\
g_{12}(\theta, \dot{\theta}, \ddot{\theta}, p_{12}) \\
g_{21}(\theta, \dot{\theta}, \ddot{\theta}, p_{21}) \\
g_{22}(\theta, \dot{\theta}, \ddot{\theta}, p_{22})
\end{pmatrix}
\]

(18)

Since muscles and movable segments are two distinct physical elements, having distinct biomechanical functions, two successive inversion steps have been chosen to first control each segment of the arm, and second, to control the target muscle forces. The biomechanical function of the muscles, including rigidity and viscosity, is referred to as g, and the mechanical function of the limb, including the matrix of inertial moments and interaction forces, is referred to as G. Equation 18 can be schematically represented by the right side of the dotted line on figure 6. Moreover, in figure 7, \(G^*\) and \(g^*\) represent respectively the estimated internal forward models of mechanical (e.g inertial, coriolis forces) and biomechanical features (e.g viscosity and stiffness of the muscle). Each of these forward models is embedded within internal feedback loops, thus allowing the inverse dynamic computation for both mechanical \(G^{-1}\) and biomechanical \(g^{-1}\) functions.

Fig. 6. Forward model of the robotic 2 dof arm.

Fig. 7. Adaptation of the proposed model to the control of 2-link planar arm. G and g represent, respectively, the forward transfer functions of the mechanical and biomechanical features of the actual physical system. CC: Cerebellar Cortex; CN: Cerebellar Nucleus; RN: Red Nucleus; P and Q are respectively the signal issued from Purkinje cell and from the Cerebellar Nucleus for the computation of \(G^*\) and \(g^*\).
If the number of active muscles across a joint exceeds the number of available equilibrium equations, the muscle forces needed for equilibrium cannot be calculated algebraically. In order to resolve this problem, we assume that muscles act together and are recruited in order to minimize an objective function (for each joint). We used a pseudo-inverse method to efficiently compute muscle forces that generate desired joint accelerations at each time step of the forward dynamic model.

\[
\tau(\dot{\theta}, \theta^*, u) = A \cdot F(\varepsilon, \dot{\varepsilon}, p) \Rightarrow F(\varepsilon, \dot{\varepsilon}, p) = A^+ \cdot \tau(\dot{\theta}, \theta^*, u)
\] (19)

D. Learning

Artificial neural networks (ANNs) were used to implement each internal model. The predictive ANNs noted \( G^* _i \) (\( i = 1 \) or \( 2 \)) are expected to learn internal models of the biomechanics of the limb segments, and to anticipate the velocity resulting from the muscle forces. The predictive ANNs noted \( g^* _{ij} \) (\( i \) and \( j \) from 1 to 2) are expected to learn internal models of the biomechanics of the muscles, to anticipate the muscle forces resulting from the motor commands). The architecture of the network is a blueprint of the connectivity of the cerebellar cortex [5], [7], [12], [13]. Using formal neurons, we modeled the three principal types of neurons of the cerebellar cortex: granular cells, Golgi cells and Purkinje cells (Fig. 8).

Forward models can be easily learned using supervised learning based on comparing the predicted and the actual outcome of a motor command [9]. Here, the learning, which takes place in the dendrites of the Purkinje cells, is modelled by adaptive synaptic weights located in the cerebellar cortex. Based on the gradient descent learning strategy, the learning rule is given by:

\[
\delta v_j = -\epsilon \cdot \text{error} \cdot eGr_j
\] (20)

where the error is the teaching signal, assumed to be issued from the Inferior Olive (Fig. 9) and calculated from the differences between target and actual signals in velocity (to \( G^* _i \)) and force (to \( g^* _{ij} \)). \( \epsilon \) is a learning rate (small and positive), and \( eGr_j \) is a synaptic eligibility [16], that is calculated by:

\[
\tau \cdot \frac{deGr_j}{dt} = -eGr_j + Gr_j
\] (21)

III. SIMULATION AND REAL TIME LEARNING RESULTS

The learning of 2 dof pointing movements, involving our sensory motor control scheme, has been performed both in simulation and in the experimental (real-time) mode. In the simulation learning mode, the peripheral system, denoted by \( H \) in the general model (Fig. 1 and 2) and by \( G, g \) pairs in the fully developed circuits (Fig. 7 and 9), is represented by the model of the experimental site as described in section II.C.2. In the experimental mode, the robotic arm is directly used to provide real-time movement and sensory feedback. In both cases the same 48 points (i.e. blue points in Figure (10)) were used to construct the learning space (\( W_L \)). They were selected to lie within the real workspace (\( W_R \)) of the arm (depicted in gray color in the same figure).

![Fig. 9. Connectivity of the artificial neural network assumed to represent the cerebellar cortex allowing the computation of \( H^* \). Pij: synaptic weight related to the glomeruli cells, Fgr: transfer function of the granule cells, Gr: signal from the granule cells, Vj: synaptic weight linked to the Purkinje cell, Fpu: transfer function of the Purkinje cell, Fgo: transfer function of the Golgi cell.](image)

![Fig. 10. Positions of the learning and the performance test points on the robot workspace. Grey area: Robot workspace; blue circles: 48 points of the learning workspace; red stars: 10 test points within the learning space; black triangles: 10 test points outside of the learning space.](image)
neural networks used to estimate the direct functions of the artificial muscles (g), as well as that of the arm dynamics (G) as described in section II.D, have all the same architecture copied from the connectivity of the cerebellar cortex. The whole architecture consisted of 33 Granular, 33 glomerules and 6 Purkinje cells. The learning curves obtained over 200 iterations are shown in figure 11 both for the experimental (Fig. 11.A) and simulation mode (Fig. 11.B). The error saturated in both cases after about 100 iterations.

Fig. 11. Learning curves obtained over 48 points belonging to the W1,i. A) Learning curve obtained in real-time learning B) Learning curve obtained in simulation learning mode

In order to validate the proposed model we have conducted two different types of experiments. First, we tested the pointing performance of the robotic arm driven by the model trained in experimental mode, and compared it to the results obtained with a PID controller. A further revised model, including a sensory feedback (FB) during execution in the form of a reflex, i.e. a sensory feedback sent directly to the alpha motoneurons, was also tested. Only the difference between target and actual joint values i.e. the error in joint space, was taken into account. This error was directly applied to the final motor commands i.e. to the servo valves controlling the air pressure of the muscles.

Figures 12, 13 and 14 represent the results of this test for a given configuration (θ1=θ2=30°) obtained by (i) the model without feedback, (ii) the model with the feedback, and (iii) by the PID controller.

Fig. 12. Robot Performance for θ1=θ2=30° with the weights adjusted during the experimental mode without the sensory feedback of joint angle value. A) Target (red) and accomplished (blue) shoulder angles B) Target (red) and accomplished (blue) elbow angles C) Shoulder muscles servo-valve outputs D) Elbow muscles servo-valve outputs

Fig. 13. Robot performance for θ1=θ2=30° with the weights adjusted during the experimental mode with the contribution of the sensory feedback of joint angle value. A) Target (red) and accomplished (blue) shoulder angles B) Target (red) and accomplished (blue) elbow angles C) Shoulder muscles servo-valve outputs D) Elbow muscles servo-valve outputs

Fig. 14. Robot performance with the classical PID controller for each joint with Kp1=3, Ki1=0.01, Kd1=10 for the first (shoulder) joint and Kp2=2, Ki2=0.01, Kd2=1 for the second (elbow) joint. A) Target (red) and accomplished (blue) shoulder angles B) Target (red) and accomplished (blue) elbow angles C) Shoulder muscles servo-valve outputs D) Elbow muscles servo-valve outputs

The red lines represent the target and the blue lines the accomplished movement. The performance was measured for both joint angles θ1 and θ2. The servo-valve outputs corresponding to the realized movement are also provided. The controller employing the sensory feedback in the form of a reflex provides the best performance, compared to the classical PID controller and to the controller without sensory feedback.

The second test consisted in measuring the capacity of the model to generalize pointing movements. Performances in extrapolation and in interpolation were measured on the 10 test points shown in figure 10 which lie either outside or within the learning space. During the experimentations the robotic arm was driven by the controllers trained both in the simulation and in experimental mode. The average end point errors obtained by the three types of controllers are reported.
The average pointing errors in meter calculated on the test points which belong either outside or within the learning space. WLI=Set of test points inside the learning workspace, WLo=Set of test points outside the learning workspace, FB=Feedback.

<table>
<thead>
<tr>
<th>Type of test</th>
<th>End Point Error [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real Time Learning on $W_{L,i}$</td>
<td>0.048</td>
</tr>
<tr>
<td>Real Time Learning on $W_{L,i}$ with FB</td>
<td>0.0147</td>
</tr>
<tr>
<td>Simulation Learning on $W_{L,i}$</td>
<td>0.066</td>
</tr>
<tr>
<td>Simulation Learning on $W_{L,i}$ with FB</td>
<td>0.0167</td>
</tr>
<tr>
<td>PID on $W_{i}$</td>
<td>0.0861</td>
</tr>
<tr>
<td>Real Time Learning on $W_{L,o}$</td>
<td>0.0378</td>
</tr>
<tr>
<td>Real Time Learning on $W_{L,o}$ with FB</td>
<td>0.0118</td>
</tr>
<tr>
<td>Simulation Learning on $W_{L,o}$</td>
<td>0.0677</td>
</tr>
<tr>
<td>Simulation Learning on $W_{L,o}$ with FB</td>
<td>0.017</td>
</tr>
<tr>
<td>PID on $W_{L,o}$(instable)</td>
<td>0.3</td>
</tr>
</tbody>
</table>

The important results reported in Table III is twofold. On the one hand, they provide a numerical proof for the higher performance of the real-time learning over simulation learning. On the other hand, they show the robustness (i.e. the generalizing capacity) of the proposed biomimetic controller for reaching points outside those of the learning workspace, for which the PID controller (with the parameters given in figure 14) performed much worse and was sometimes unstable.

IV. CONCLUSIONS AND FUTURE WORK

This paper describes a model of cerebellar pathways that can be used to control an anthropomorphic robot arm. The results show that an architecture, which includes ANNs modelling the cerebellar pathways, can successfully learn the internal representation of the robot arm. After learning, such a cerebellar-like circuit computes an approximate inverse model of the limb biomechanics based on sensori-motor predictions and controls 2 dof robot pointing movements. We are currently working on the improvement of our convergence criteria. Our next project is to add internal feedbacks to the model, which will not be limited to the position errors but will include also a force feedback, and to test its performance with movements in the vertical plane, as well as against random perturbations during movement execution.

REFERENCES