Cerebellarlike Corrective Model Inference Engine for Manipulation Tasks

Niceto Rafael Luque, Jesús Alberto Garrido, Richard Rafael Carrillo, Olivier J.-M. D. Coenen, and Eduardo Ros

Abstract—This paper presents how a simple cerebellumlike architecture can infer corrective models in the framework of a control task when manipulating objects that significantly affect the dynamics model of the system. The main motivation of this paper is to evaluate a simplified bio-mimetic approach in the framework of a manipulation task. More concretely, the paper focuses on how the model inference process takes place within a feedforward control loop based on the cerebellar structure and on how these internal models are built up by means of biologically plausible synaptic adaptation mechanisms. This kind of investigation may provide clues on how biology achieves accurate control of non-stiff-joint robot with low-power actuators which involve controlling systems with high inertial components. This paper studies how a basic temporal-correlation kernel including long-term depression (LTD) and a constant long-term potentiation (LTP) at parallel fiber-Purkinje cell synapses can effectively infer corrective models. We evaluate how this spike-timing-dependent plasticity correlates sensorimotor activity arriving through the parallel fibers with teaching signals (dependent on error estimates) arriving through the climbing fibers from the inferior olive. This paper addresses the study of how these LTD and LTP components need to be well balanced with each other to achieve accurate learning. This is of interest to evaluate the relevant role of homeostatic mechanisms in biological systems where adaptation occurs in a distributed manner. Furthermore, we illustrate how the temporal-correlation kernel can also work in the presence of transmission delays in sensorimotor pathways. We use a cerebellumlike spiking neural network which stores the corrective models as well-structured weight patterns distributed among the parallel fibers to Purkinje cell connections.

Index Terms—Adaptive, biological control system, cerebellum, learning, plasticity, robot, simulation, spiking neuron.

I. INTRODUCTION

CONTROLLING fast non-stiff-joint robots accurately with low-power actuators is a difficult task which involves high inertia. Biological systems are, in fact, non-stiff-joint "plants"

N. R. Luque, J. A. Garrido, and E. Ros are with the Department of Computer Architecture and Technology, University of Granada, 18071 Granada, Spain (e-mail: nluque@atc.ugr.es; jgarrido@atc.ugr.es; eros@atc.ugr.es).

R. R. Carrillo is with the Department of Computer Architecture and Electronics, University of Almeria, 04120 Almería, Spain (e-mail: rcarrillo@atc.ugr.es).

O. J.-M. D. Coenen is with the Intelligent Systems Research Center, University of Ulster, Derry, BT48 7JL, Northern Ireland (e-mail: olivier@oliviercoenen.com).

driven with relatively low-power actuators. However, in this case, control schemes require building accurate kinematic and dynamic models (dynamic models would not be required in the case of very stiff joint robots with inappreciable inertia). Even if the basic dynamics model is very accurate, manipulating tools and objects will affect this base model. This will lead to significant distortions along the desired movements, affecting the final accuracy. Therefore, these systems require adaptive modules for tuning the corrective models to specific object or tool manipulation. This challenge has been smartly solved by the biological systems by using the cerebellum as a force, stiffness, and timing control machine in every human movement. The cerebellar cortex performs a broad role in different key cognitive functions [1]. Three different layers constitute the cerebellar cortex—the molecular layer, the Purkinje layer, and finally, the granular layer. The cerebellar cortex seems to be well structured into microzones [2] related to a specific somatotopic organization in sensor and actuator areas. The human cerebellum involves about 10 000 000 Purkinje cells receiving excitatory inputs from parallel fibers (150 000 excitatory synapses at each Purkinje cell). Each parallel fiber synapses on about 200 Purkinje cells; these parallel fibers are granule cell axons. These granule cells are excited by mossy fibers (with afferent connections from the spinal cord, with sensory and motor estimates). Each Purkinje cell receives further excitatory synapses from one single climbing fiber. This connection is so strong that the activity from a single climbing fiber can drive the Purkinje cell to fire [3]. These spikes from the Purkinje cells generated by climbing fibers are called complex spikes, while the Purkinje cell spikes generated by the activity received from the parallel fibers are called simple spikes. Basket cells, being activated by parallel fiber afferents, can inhibit Purkinje cells. Finally, Golgi cells receive input from parallel fibers, mossy fibers, and climbing fibers, and inhibit granule cells. The output of a Purkinje cell is an inhibitory signal to the deep cerebellar nuclei [3] (Fig. 1). Granule cells and Purkinje cells play an important role in pattern recognition [4]. We can assume that the granular layer adaptation mechanism is essentially unsupervised [5] toward enhancing information transmission. In this layer, an efficient recoding of mossy fiber activity takes place, improving the learning capability in subsequent stages (granular cell-Purkinje cell synapse). The cerebellum seems to play a crucial role in model inference within manipulation tasks but the way this is supported by actual network topologies, cells, and adaptation properties is an open issue.

We have addressed the study of how this model inference task can be achieved in a local and distributed manner with a basic cerebellumlike architecture based on spiking neurons. Furthermore, we evaluate how spike-timing-dependent
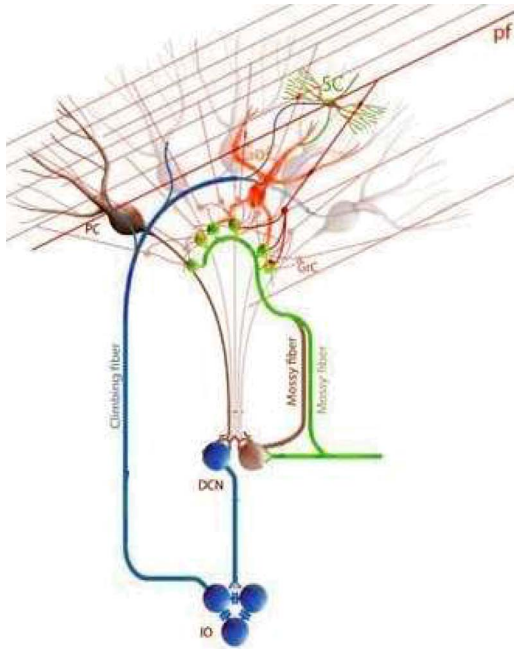
Fig. 1. **Scheme of the cerebellum organization** [6]. This scheme shows the most relevant connections within a cerebellar module. The cerebellar module presents different connections communicating different circuit elements in closed loops. Mossy fibers contact granule cells (GrC) and DCN cells which, in turn, receive inhibition from the same common set of Purkinje cells (PC). Moreover, the IO cells project climbing fibers that contact PC which also are projected to DCN cells.

plasticity (STDP) provides an efficient learning rule for this task. We do this by using a simple temporal-correlation kernel [long-term depression (LTD)] and a constant compensating long-term potentiation term (LTP) as the adaptation mechanism at the parallel fiber (PF)-Purkinje cell (PC) synapses. We explore how the LTP and LTD components of this learning rule need to be well balanced to achieve an acceptable performance. Although different systems that potentially compensate transmission delays have been proposed [7], [8]; in this paper, we explicitly avoid compensating them. The correlation kernel is able to correlate sensorimotor activity with error estimates without explicitly taking into account the transmission delays. This inferred model is therefore trajectory-specific. By means of a certain correlation kernel, the effect of several input spikes on plasticity is accumulated in a reduced number of variables without the necessity of storing spike times. This makes this correlation kernel computationally efficient for event-driven processing engines, as the one used in this paper, EDLUT [9]. In this paper, we explicitly evaluate how these corrective models are structured in a distributed manner among different synapses in the PF-PC connection space. The possibility of monitoring this spatio-temporal learned weight pattern represents a powerful tool to interpret how models are inferred to enhance the accuracy in a control task. We evaluate how this learning engine with specific (fixed gain) LTP and correlation-based LTD components can infer different corrective dynamic models corresponding to the manipulation of objects of different masses.

Control schemes of biological systems must cope with significant sensorimotor delays (100 ms approximately) [10]–[12]. Furthermore, actuators are very efficient but have a limited

power and have to deal with viscoelastic elements. In order to deal with all these issues, biology has evolved efficient "model inference engines" to facilitate adaptive and accurate control of arms and hands [13]–[15]. A wide range of studies have proven the crucial role of the cerebellum in delivering accurate correcting motor actions to achieve high-precision movements even when manipulating tools or objects (whose mass or moment of inertia significantly affects the base dynamics models of the arm-hand) [15]–[17]. For this purpose, the cerebellum structure needs to infer the dynamics model of the tool or object under manipulation [18] and store it in a structured way that allows an efficient retrieval of corrective actions when manipulating this item. There are scientific evidences of synaptic plasticity at different sites of the cerebellum and the sensorimotor pathway. The synaptic connection between PFs and PCs seems to have a significant impact on the role of inferring models of sensorimotor correlations for delivering accurate corrective commands during control tasks in most cerebellar models [19]–[21]. Furthermore, the adaptation at this site seems to be driven by the activity coming from the inferior olive (IO) and by the way this activity correlates with the activity received through the PFs.

Within a cerebellarlike cell-based structure, the corrective model is inferred in a distributed way among synapses. Furthermore, this scheme based on distributed cell populations allows several models to be inferred in a non-destructive way by selecting a specific population each time.

The main goal of this paper is the study of how an adaptive cerebellumlike module embedded in the control loop can build up corrective models to compensate deviations in the target trajectory when the dynamics of the controlled plant (arm-hand-object in the case of a human operator) are altered due to manipulation of heavy objects (whose mass significantly affects the basic model dynamics). We address the study of how this corrective model is inferred through a biologically plausible local adaptation mechanism. To better illustrate this issue, we have simplified the cerebellum architecture.

Through this simple cerebellar structure, we have monitored how the weight's space adapts to a distributed stable model that depends on the basic network topology, the target trajectory, and model deviations.

The IO is an important paracerebellar center whose functional role is still an open issue [5], [6], [22]–[25]. Different research groups have studied its potential role in delivering a teaching signal during accurate movements [26]–[29]. The IO is the only source of cerebellar climbing fibers (CFs) which target the Purkinje cells (PC). Each PC receives a single CF which massively connects with this single neuron strongly driving its activity. When a spike of the IO reaches its target PC, the Purkinje cell fires a complex spike. Each CF connects approximately with ten PCs. Nevertheless, the IO fires at a very low frequency (between 1–10 Hz, average 1 Hz) and therefore, the amount of spikes coming from the CFs is almost negligible compared to the activity of the PCs generated by the parallel fibers (simple spikes) [30]–[33].

Neurophysiologic studies have revealed that there are many adaptation mechanisms at the cerebellum. Each of them may have a specific purpose (segmentation, maximization of information transference, correlation of sensorimotor signals, etc.) [34], [35]. In particular, the activity of the IO has a strong

impact on the PF-PC synaptic adaptation [36]. The adaptation of these synapses mediated by this activity seems to play a crucial role in correlating the sensorimotor activity with a "teaching signal" (arriving from the IO) [19], [20], [37], [38]. This teaching signal can be seen as an "intentional signal" that highlights, in time domain, the accuracy of the movement that is being performed. As proposed in [12], [39], this signal may be related to the error during a movement. But since the IO is only capable of very low-frequency output spikes (typically, output activity between 1 and 10 Hz), it does not encode the error quantity accurately in only one movement repetition, but rather provides a progressive estimate. Therefore, during repetitions of movements, its statistical representation may reproduce the error evolution more accurately [28], [40], [41] and thus, it can be a useful guide toward efficient error reduction to achieve accurate movements.

II. MATERIALS AND METHODS

For extensive spiking network simulations, we have further developed and used an advanced event-driven simulator based on LookUp Tables [9], [42], [43]. EDLUT is an open-source tool [42], [43] which allows the user to compile the response of a predefined cell model (whose dynamics are driven by a set of differential equations) into lookup tables. Then, complex network simulations can be performed without requiring an intense numerical analysis. In this research, as a first approximation, neurons were evolved versions of leaky integrate-and-fire neuron models with the synapses represented as input-driven conductances.

For the experimental work, we have used a biomorphic robot plant, a simulated LWR (lightweight robot). This robot has been developed at DLR [44]. The LWR's arms are of specific interest for machine-human interactions in unstructured environments. In these scenarios, the use of low-power actuators prevents potential damage on humans in case of malfunctioning. Although a real impact on robotic applications is beyond the scope of this paper, the target application scenario of this robotic robot based on non-stiff low-power actuators shares certain characteristics with the daily manipulation tasks performed by humans. Therefore, we considered this robotic platform an appropriate tool for validating the cerebellar-based model inference engine under study.

For the sake of simplicity, in our simulations, we use a simulator of this robot in which we have fixed some joints to reduce the number of actual joints to three, limiting the number of degrees of freedom to three.

A. Training Trajectory

The described cerebellar model has been tested in a smooth pursuit task [45]–[47]. A target (desired target movement) moves along a repeated trajectory, which is composed of vertical and horizontal sinusoidal components. The target movement describes the "eight-shape" trajectories illustrated in Fig. 2, whose equations, in angular coordinates, are given by the following expressions (1). We have evaluated the learning capability performing a goal movement along this target trajectory.
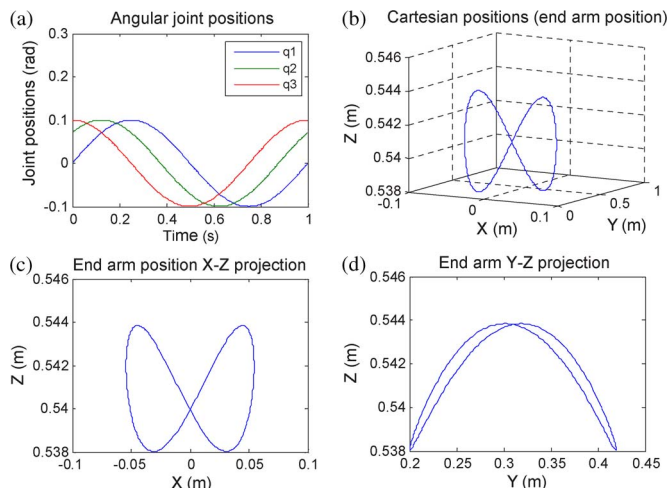

Fig. 2. **Three joint periodic trajectories describing eight-shape movements in joint coordinates. This trajectory implies movements of three joints.** (a) Cartesian coordinates of the eight-like trajectory. (b) 3-D view of the eight-like trajectory. (c) X- and Z-axes representation of this target trajectory. (d) Y and Z-axes representation of the eight-like target trajectory.

Each joint movement in our task is defined by q_1, q_2, and q_3, respectively,

$$
\begin{aligned}
q1 &= A_1 \sin(\pi t) + C_1 \\
q2 &= A_2 \sin(\pi t + \theta) + C_2 \\
q3 &= A_2 \sin(\pi t + 2\theta) + C_3.
\end{aligned} \tag{1}
$$

This trajectory with the three joints which are moving following sine shapes is shown in Fig. 2. We chose fast movements (1 s for the whole target trajectory) to study how inertial components (when manipulating objects) are inferred at the cerebellar structure. Slow movements would hide changes in the dynamics of the arm+object model, since they would not have significant impact when performing very slow movements.

Though for the sake of simplicity, we have used a single eight-like trajectory in each trial, consecutive eight-like trajectories have also been tested leading to similar results (provided that the corrective torque values do not get saturated along the global trajectory).

B. Control Loop. Interfacing the Cerebellum Model With a Simulated Robot

Some studies indicate that the brain may plan and learn to plan the optimal trajectory in intrinsic coordinates [14], [48]–[50]. The central nervous system is able to execute three major tasks—the desired trajectory computation in visual coordinates, the task-space coordinates translation into body coordinates, and finally, the motor command generation. In order to deal with variations of the dynamics of the operator arm, we have adopted an feedback error learning scheme [51] in conjunction with a crude inverse dynamic model. In this scheme, the association cortex provides the motor cortex with the desired trajectory in body coordinates, where the motor command is calculated using an inverse dynamic arm model. On one hand, the spinocerebellum—magnocellular red nucleus system provides an internal neural accurate model of the dynamics of the musculoskeletal system which is learned with practice by sensing the result of the movement. On the other hand, the
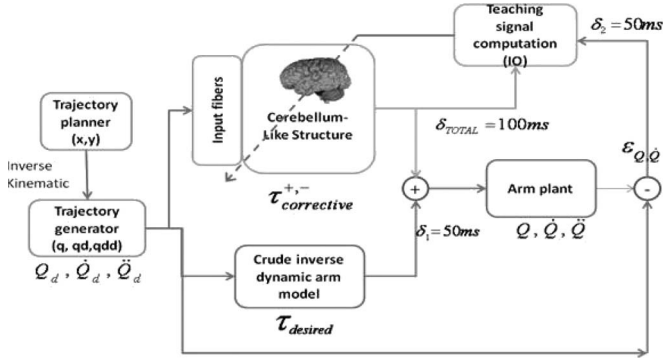
Fig. 3. Control loop. The adaptive module (cerebellarlike structure) contributes to the actual torques being received by the "crude inverse dynamics robot model" to enhance the accuracy of the movement.

cerebrocerebellum—parvocellular red nucleus system provides a crude internal neural model of the inverse dynamics of the musculoskeletal system which is acquired while monitoring the desired trajectory [51]. The crude inverse dynamic model works neck to neck with the dynamical model by updating the motor command by predicting a possible error in the movement. As it is illustrated in Fig. 3, the cerebellar pathways follow a feedforward architecture, in which only information about sensory consequences of incorrect commands can be obtained (i.e., the difference between actual and desired joint positions of the arm). The natural error signal for learning motor commands is the difference between actual and correct commands; this implies, for example, that if M muscles control N sensor dimensions involved in a task, then N-sensory errors must be converted into M-motor errors ($M \times N$ complexity). How to use this sensor information to drive motor learning is the so-called distal error problem [46], [52]. In order to overcome this motor error problem, (the cerebellum in our scheme provides torque corrections) the implemented spiking cerebellum used an adaptation mechanism described in Section II-D which can correlate the actual and desired states toward the generation of accurate corrective motor commands.

In our model, the cerebellum receives well-structured inputs encoding the planned trajectory. We assume that the errors occurred during the movement are encoded at the IO and transferred (at low firing rates) to the cerebellum through the climbing fibers.

We have built a module to translate a small set of signals (encoding the arm's desired state) into a sparse cell-based spike-timing representation (spatio-temporal population coding). This module has been implemented using a set of input fibers with specific receptive fields covering the working range of the different desired state variables (position and velocity of the different joints). In this way, the robot [analog domain consisting of trajectory planer, trajectory generator, crude inverse dynamic arm model, and arm plant (Fig. 3)] has been interfaced with the spiking cerebellar model (spiking domain).

In our control loop, the desired states (positions and velocities) that follow a certain trajectory are obtained from an inverse kinematic model computed by other brain areas [48] and then, they are translated into joint coordinates. These desired arm states are used at each time step by a crude inverse arm dynamics model to compute crude torque commands which are added to the cerebellum corrective torques. This control loop is illustrated in Fig. 3.

Fig. 3 illustrates how the trajectory planner module delivers desired positions and velocities for a target trajectory. The kinematics module translates the trajectory Cartesian coordinates into joint coordinates. The "crude inverse dynamics arm model" calculates the target torque in each joint which are necessary to roughly follow the target trajectory. But this crude arm model does not take into account modifications in the dynamics model due to object manipulation. Thus, if only these torque values are considered, the actual trajectory may significantly differ from the desired one. The adaptive cerebellar component aims at building corrective models to compensate these deviations, for instance, when manipulating objects.

In Fig. 3, the adaptive cerebellarlike structure delivers corrective actions that are added to compensate deviations in the base dynamics plant model when manipulating objects. In this feedforward control loop, the cerebellum receives a teaching error-dependent signal and the desired arm state so as to produce effective corrective commands. Total torque is delayed (on account of the biological motor pathways) and supplied to the robot plant δ_{total}. The difference between the actual robot trajectory and the desired one is also delayed $\delta_{1,2}$ and used by the teaching signal computation module to calculate the IO activity that is supplied to the cerebellum as a teaching input signal (for the computation of the cerebellar synaptic weighs). Using this control loop architecture, an accurate explicit model of the musculoskeletal arm inverse dynamics is not necessary. The cerebellum can infer corrective models tuned to different tools which may affect the dynamics of the plant (arm+object).

C. Cerebellum Model

The proposed cerebellarlike architecture, organized in cerebellar microzones [2] (somatotopic arrangement), tries to capture some cerebellum's functional and topological features [3], [53]. This cerebellum model consists of the following layers: (Fig. 4)

- Input layer (120 cells). This layer represents a simplification of the mossy and granular layers of the cerebellum and drives PCs and cells of the deep cerebellar nuclei (DCN). The goal of this simplification is to facilitate the study of how the sensorimotor corrective models are stored in adapted weights at the PF-PC connections. This input layer has been divided into six groups of 20-grouped cells which carry the desired joint velocity and position information (these desired position and velocity coordinates can be thought as efferent copies of the motor commands or "motor intention"); for the proprioceptive encoding, three groups of cells encode the desired joint positions (one group per joint) and the other three encode the desired joint velocities. The analog position and velocity transformation into the fiber spike activity is carried out by using overlapping radial basis functions (RBF) (Fig. 5) [54] as receptive fields of the input-variable space, see (2) (joint-specific angular position)

$$I_{\text{mossy}_i} = e_i^{\frac{(\text{input variable} - \mu_i)^2}{2\sigma^2}} \qquad 0 < i < n$$
where = size of mossy group, $\qquad$ (2.A)
where the mossy behavior is given by :
$$\tau_{m_i} \frac{dv_i}{dt} = -v_i(t) + R_i I_{\text{mossy}_i} \qquad (2.B)$$
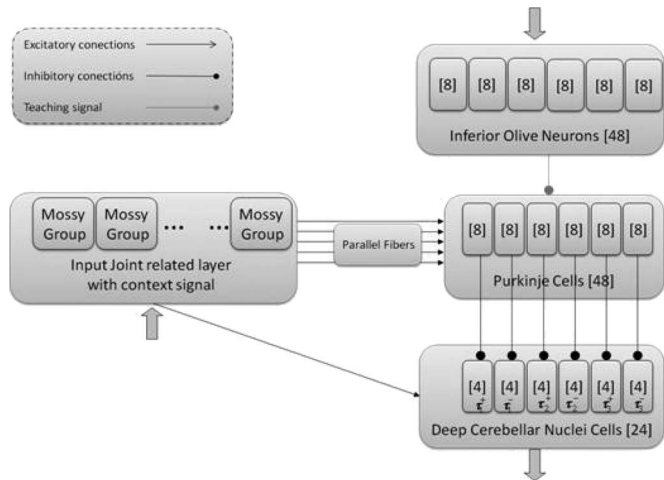
Fig. 4. **Cerebellum model diagram**. Inputs encoding the desired position and velocity (arm state) are sent (upward arrow) through the input layer which represents a simplification of the mossy fibers and granular layer. Inputs encoding the error are sent (upper downward arrow) through the inferior olive (IO). Outputs are provided by the deep cerebellar nuclei (DCN) (lower downward arrow). The DCN collects activity from the input layer (excitatory inputs which provide DCN with a basal activity when an input stimulus is presented) and the Purkinje cells (inhibitory inputs). The DCN activity represents the corrective torque generated by the cerebellum. This output activity is transformed into a proper analog torque signal by means of a buffer in which the DCN activity is accumulated. This activity buffer is used to compute an analog average value that acts as a corrective torque.
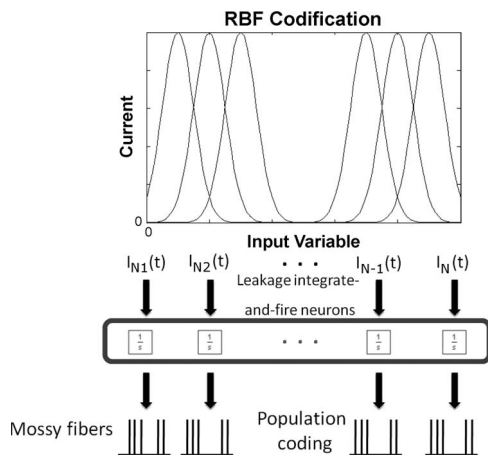

Fig. 5. **Encoding of cerebellar input signals**. Translation from joint-related analog variables (angular positions and velocities) into spike trains is carried out using overlapping RBFs as receptive fields in the analog domain. One-dimensional values are transformed into multidimensional current vectors (one for each RBF). Each current value is integrated using an integrate-and-fire (I & F) neuron model and determines the output activity of an input cell of the cerebellum model.

where μ_i is the mean and σ the standard deviation of i RBF. Related to the cell dynamics, τ_{m_i} is the resting time constant, v_i the membrane potential, I_{mossy_i} the input current, and R_i is related to the resting conductance of the membrane. For the sake of simplicity, in our model, we have not included a more detailed cellular structure (Golgi cells, interneurons, mossy fibers, etc.). We have adopted well-structured noise-free patterns to encode sensorimotor signals to partially embed potential roles typically performed in the granular layer [6], [55] (such as noise

reduction, pattern separation, etc.). Parallel fibers are the output of this layer.

- Inferior olive cells (IO) (48 cells). This layer consists of six groups of eight cells. It translates the error signals into teaching spikes to the Purkinje cells. The IO output carries the teaching signal used for supervised learning (see STDP section).

- Purkinje cells (PC) (48 cells). They are divided into six groups of eight cells. Each input cell sends spikes through excitatory connections to PCs, which receive teaching signals from the IO. The PF-PC synaptic conductances are set to an initial average value (15 nS) at the beginning of the simulation and are modified by the learning mechanism during the training process.

- Cells of the DCN (24 cells). The cerebellum model output is generated by six groups of these cells (two groups of four cells per joint) whose activity provides corrective torques to the specified arm commands. The corrective torque of each joint is encoded by a couple of these antagonist groups, being one group dedicated to compensate positive errors and the other one to compensate negative errors. Each neuron group in the DCN receives excitation from every input layer cell and inhibition from the two corresponding PCs. In this way, the PC-DCN-IO sub circuit is organized in six microstructures (Fig. 4), three for positive joint corrections (one per joint) and three for negative joint corrections (one per joint).

We have used leaky integrate-and-fire (I&F) neurons with synapses modeled as variable conductances to simulate Purkinje cells and DCN cells. These models are a modified version of the spike response model [56]. These synaptic conductance responses were modeled as decaying exponential functions triggered by input spikes as stated by (3.A)–(3.C). Thus, these neuron models account for synaptic conductance changes (driven by pre-synaptic activity) rather than simply for current flows, providing an improved description over more basic I&F models. Table I contains the neuron model parameters of the Purkinje cells and DCN cells

$$g_{\text{exc}}(t) = \begin{cases} 0, & t < t_0 \\ g_{\text{exc}}(t_0) \cdot e^{-\frac{t-t_0}{\tau_{\text{exc}}}}, & t \geq t_0 \end{cases} \quad (3.\text{A})$$

$$g_{\text{inh}}(t) = \begin{cases} 0, & t < t_0 \\ g_{\text{inh}}(t_0) \cdot e^{-\frac{t-t_0}{\tau_{\text{inh}}}}, & t \geq t_0 \end{cases} \quad (3.\text{B})$$

$$C_m \frac{dV_m}{dt} = g_{\text{exc}}(t)(E_{\text{exc}} - V_m) + g_{\text{inh}}(t)(E_{\text{inh}} - V_m)$$
$$+ G_{\text{rest}}(E_{\text{rest}} - V_m) \quad (3.\text{C})$$

where g_{exc} and g_{inh} represent the excitatory and inhibitory synaptic conductance (time constant) of the neuron. τ_{exc} and τ_{inh} represent the time constants of the excitatory and inhibitory synapses, respectively. Synaptic inputs through several synapses of the same type can simply be recursively summed when updating the total conductance if they have the same time constants, as indicated in (4). Membrane potential (V_m) is defined through (3.C) depending on the different reverse potentials and synaptic conductances

$$g_{\text{exc(post-spike)}}(t) = G_{\text{exc},j} + g_{\text{exc(pre-spike)}}(t) \quad (4)$$

$G_{\text{exc},j}$ is the weight of synapse j; a similar relation holds for inhibitory synapses.

TABLE I
NEURON MODEL PARAMETERS FOR THE SIMULATIONS [57]–[61]. IN THE TABLE, nS STANDS FOR NANOSIEMENS AND syn STANDS FOR SYNAPSES

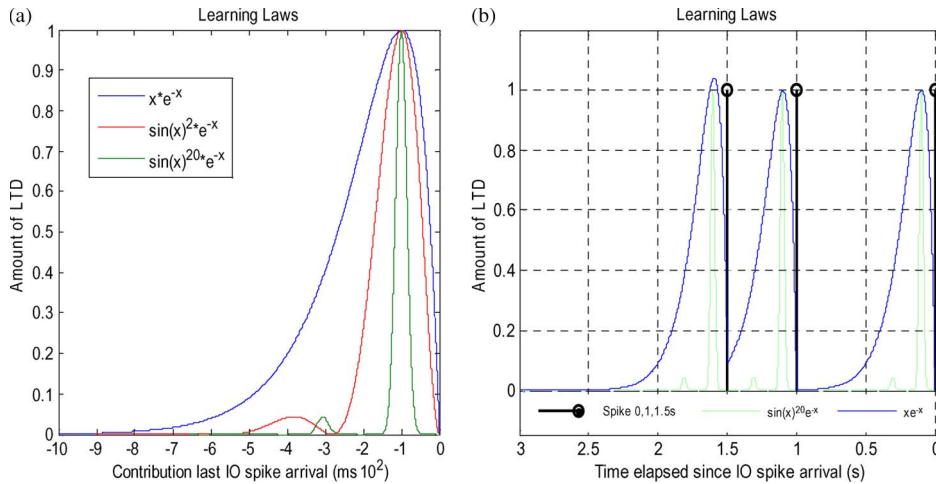|	Purkinje cells	DCN cells
Refractory period	2ms	1ms
Membrane capacitance	500pF	2pF
Total excitatory peak conductance	1.3nS·175000syn.·10%	1nS·7syn.
Total inhibitory peak conductance	3nS·150syn.	30nS·1syn.
Firing threshold	-52mV	-40mV
Resting potential	-70mV	-70mV
Resting conductance	16nS	0.2nS
Mem. pot. time constant (τ_m)	20ms to 30ms	10ms
Exc. syn. time constant (τ_{exc})	1.2ms	0.5ms
Inh. syn. time constant (τ_{ihn})	9.3ms	10ms

Fig. 6. **LTD integral kernel**. (a) Representation of a basic integral kernel $(x \cdot e^{-x})$ which has a rather wide peak that makes PC synaptic weights to decrease more prominently and a more complex integral kernel $(\sin(x)^{20}) \cdot e^{-x}$ which has a sharper peak. (b) This plot shows the amount of LTD at a particular synapse depending on the IO spike arrival time elapsed since the PF spikes for different integral kernels. The figure includes a comparison between the basic integral kernel $(x \cdot e^{-x})$ and a more complex integral kernel $(\sin(x)^{20}) \cdot e^{-x}$ which has a peak 100 ms after the input spike. The PC receives three spikes through a particular CF at times 0.0, 1, and 1.5 s.

D. STDP

The studied cerebellar model only includes synaptic plasticity at the PF-PC connections. The changes of the synaptic efficacy for each connection are driven by pre-synaptic activity (STDP) and are instantaneous.

In our model, since there are delays in the transmission of joint torque values and joint position measurements, the trajectory error measurements (which are used to calculate the teaching signal) reach the cerebellum with a 100-ms delay. This means that the learning mechanism must learn to provide corrective torque predictions.

This plasticity has been implemented including LTD and LTP mechanisms in the following way:

a) LTD produces a synaptic efficacy decrease when a spike from the IO reaches a PC, as indicated in (6.A). The amount of efficacy which decreases depends on the previous activity arrived through the PF (input of the cerebellar model). This previous activity is convolved with an integral kernel as defined by (5). This mainly takes into account those PF spikes which arrived 100 ms before the IO spike (see Fig. 6). This correction is facilitated by a time-logged "eligibility trace" [45], [47], [62], [63], which takes into account the past activity of the afferent

PF. This trace aims to calculate the correspondence in time between spikes from IO (error-related activity) and the previous activity of the PF which is supposed to have provoked this error signal. The eligibility trace idea stems from experimental evidence showing that a spike in the climbing fiber afferent to a Purkinje cell is more likely to depress a PF-PC synapse if the corresponding PF has been firing between 50 and 150 ms before the IO spike (through CF) arrives at the PC [45], [47]

$$k(t) = e^{-(t-t_{\text{postsynapticspike}})} \sin(t - t_{\text{postsynapticspike}})^{20}. \quad (5)$$

b) LTP produces a fixed increase in synaptic efficacy each time a spike arrives through a PF to the corresponding PC as defined by (6.B). With this mechanism, we capture how an LTD process, according to neurophysiologists studies [64], can be inverted when the PF stimulation is followed by spikes from the IO or by a strong depression of the Purkinje cell membrane potential.

The strength of these two mechanisms needs to be tuned to complement and compensate each other. These biological LTP-LTD properties at PF-PC synapses have been tried to be emulated in different fields, i.e., in the adaptive filter [65] theory by

using the heterosynaptic covariance learning rule of Sejnowski [66] or in the adaptive control theory by using the least-mean-square learning rule [67]. Different alternative temporal kernels are shown in Fig. 6. The sharper the integral kernel peak is, the more precise the learning becomes. On the other hand, this leads us to a slower synaptic weight adaptation. However, LTP can lead the weight recruitment to be compensated by future IO activity. This situation drives us to faster synaptic weight saturation where LTP can hardly carry out the weight recruitment for future IO activity. After the main peak in the correlation kernel, a second marginal bump can be seen as a consequence of the mathematical model used for modeling the correlation engines. The chosen mathematical models of the kernel allow accumulative computation in an event-driven engine, avoiding the necessity of integrating the whole correlation kernel each time a new spike arrives. Therefore, these correlation models are computationally efficient in the framework of an event-driven simulation scheme, such as EDLUT [9], but they suffer this second marginal peak that can be considered noise in the weight integration engine.

This is indicated in the following (6):

$$LTD, \forall i, \Delta w_i = - \int_{-\infty}^{IO\text{spike}} k(t - t_{IO\text{spike}}) \delta_{GR\text{spike}-i}(t) dt \tag{6.A}$$

$$LTP, \Delta w_i = \alpha. \tag{6.B}$$

E. Teaching Signal of the Inferior Olive

The crude inverse dynamics controller generates motor torque values for a rough control, but the long delays in the control loop prevent the online correction of the trajectory in a fast reaching task using a classical controller with a continuous feedback. In the studied control model, the trajectory error is used to calculate the teaching signal. This teaching signal follows (7)

$$\epsilon_{\text{delayed}_i} = K_{pi} \cdot \epsilon_{\text{position}_i} + K_{vi} \cdot \epsilon_{\text{velocity}_i}$$
$$i = 1, 2, 3, \ldots \text{joint}$$
$$\epsilon_{\text{position}_i} = (q_{i\text{desired}} - q_{i\text{real}}) \left[(t + t_{\text{pred}})_i - t_i \right]$$
$$\epsilon_{\text{velocity}_i} = (\dot{q}_{i\text{desired}} - \dot{q}_{i\text{real}}) \left[(t + t_{\text{pred}})_i - t_i \right] \tag{7}$$

where $K_{pi} \cdot \epsilon_{\text{position}_i}$ represents the product of a constant value (gain) at each joint K_{pi} and the position error in this joint [difference between desired joint position and actual joint position $(q_{i\text{desired}} - q_{i\text{real}})$].

$K_{vi} \cdot \epsilon_{\text{velocity}_i}$ represents the product between a constant value (gain) at each K_{vi} joint and the velocity error in this joint [difference between desired joint velocity and actual joint velocity $(\dot{q}_{i\text{desired}} - \dot{q}_{i\text{real}})$].

The IO neurons synapse onto the PCs and contribute to drive the plasticity of PF-PC synapses. These neurons, however, fire at very low rates (less than 10 Hz), which appears problematic to capture the high-frequency information of the error signal of the task being learned. This apparent difficulty may be solved by their irregular or chaotic firing [13], [41], [68]. This is a very important property, which has the beneficial consequence of statistically sampling the entire range of the error signal over multiple trials (see below). Here, we implemented this irregular

firing using a Poisson model [69] for spike generation. The weight adaptation was driven by the activity generated by the IO, which encoded the teaching signal into a low-frequency probabilistic spike train (from 0 to 10 Hz, average 1 Hz) [5], [41].

We modeled the IO cell responses with probabilistic Poisson process. Given the normalized error signal $\varepsilon(t)$ and a random number $\eta(t)$ between 0 and 1, the cell fired a spike if $\varepsilon(t) > \eta(t)$; otherwise, it remained silent [47]. In this way, on one hand, a single spike reported accurately timed information regarding the instantaneous error; and on the other hand, the probabilistic spike sampling of the error ensured that the whole error region was accurately represented over trials with the cell firing almost ten spikes per second. Hence, the error evolution is accurately sampled even at a low frequency [12]. This firing behavior is similar to the ones obtained in physiological recordings [41].

LTD and LTP play complementary roles in the model inference process. The LTP implemented at the PF-PC synapses was a non-associative weight increase triggered by each input cell spike [64]. The LTD was an associative weight decrease triggered by spikes from the inferior olive [26], [27]. This model of LTD uses a temporal kernel, shown in Fig. 6, which correlates each spike from the IO with the past activity of the parallel fiber [10], [45], [70]. Correlation-based LTD allows the adjustment of specific PF-PC connections to reduce the error according to the IO activity. When IO spikes are received, the synaptic weights of the PF-PC connections are reduced according to the temporal-correlation kernel and to the activity received through the PF. In this way, we reduce the probability of production of simple spikes by PC due to the activity coming from the PFs through these specific connections. Therefore, the IO effectively modulates the spatio-temporal corrective spike patterns. In this model, a learning state in the cerebellum (PF-PC weights) can be seen as a bidimensional function which relates each PF and PC combination with their corresponding synaptic weight [Fig. 7(c)].

Physiologically, the time matching of the desired and actual joint states can be understood by the fact that the trajectory error would be detected at the level of the spinal cord through a direct drive from the gamma motoneurons to the spinal cord [71].

III. SIMULATIONS AND RESULTS

We have carried out several simulations to study different issues: a) How LTD and LTP need to be balanced to optimize the adaptation performance; b) how the temporal-correlation kernel (integral kernel) works even in the presence of sensorimotor delays; and c) how the same learning mechanism can adapt the system to compensate different deviations in the basic model dynamics (due to manipulating objects of different weights).

A. LTD Versus LTP Trade-Off

At the beginning of the learning process (before the connection weights are adjusted), the spikes received from the input fibers excite the DCN cells, producing a "bias correction" term on the motor commands. The role of the cerebellar PF-PC-DCN loop is to specifically inhibit this bias term according to a spatio-temporal pattern that is inferred during
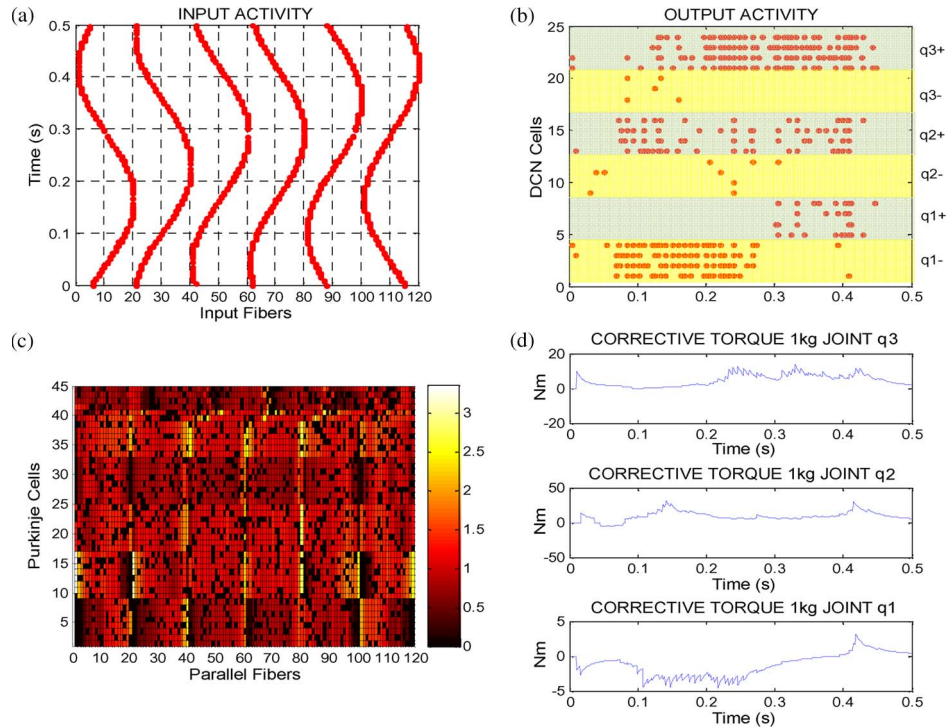
Fig. 7. **Cerebellum state after 300 trajectory learning iterations**. (a) Input activity at the cerebellum. The input layer produces a set of spikes after the transformation from analog domain into spike domain These spikes are transmitted directly by PF. This activity (desired positions and velocities) keeps on constant during all iterations. (b) DCN output activity generated by those synaptic weights. Error corrections are accomplished by changes in the activity of PCs that, in turn, influence the activity of the DCN [72], which afterward is translated into analog torque correction signals. Each group of four DCN cells encodes the positive or negative part of a joint corrective torque. The more activity the positive/negative group has, the higher/lower corresponding corrective torque is generated. (c) PC-PF synaptic weight representation. In x-axis, we can see the source cells (PFs). In y-axis, target cells (PCs). Dark colors represent lower synaptic weights, thus, the corresponding DCN cells are more active. We can see six well-defined rows, each row represents weights related with the positive and negative torque output of the three joints (q_3, q_2, and q_1), and six well-defined columns (related with the input activity of the PF corresponding to the desired position and velocity for the three joints). (d) Output torque after analog transformation from the DCN output spikes. These corrective torque curves have a profile strongly related with the number of DCN cells assigned per joint; thus, increasing the quantity of DCN per joint will generate a smoother corrective profile.

movement executions and to further compensate other deviations generated by the manipulation of different objects or other elements, affecting the dynamics of the initial "arm plant" (without object under manipulation). The PF-PC-DCN loop transmits an activity pattern which is adapted taking into account the teaching signals provided by the IO (described in the previous section).

In the first simulations, the arm is manipulating a 1-kg mass object. This mass significantly affects the dynamics of the arm+object. Therefore, the actual trajectory (without corrective support) deviates significantly from the target trajectory. We have studied how the cerebellar module compensates this deviation building a corrective model.

Fig. 7 illustrates how the corrective model is acquired through learning and structured in distributed synaptic weight patterns. When the arm moves along a target trajectory, different input cell populations are activated. They produce a temporal signature of the desired movement. Meanwhile, the IO continuously transfers trajectory error estimates (teaching signals) which are correlated with the input signature. In Fig. 7, the system adaptation capability is monitored. This helps to interpret how the corrective model is continuously structured. Similar monitoring experiments in much simpler scenarios and smaller scale cell areas are being conducted in neurophysiologic studies [6] to characterize the adaptation capability of neurophysiologic systems at different neural sites.

When manipulating heavy objects which do not properly fit the basic plant model, the followed trajectory drifts from the desired one before learning. This deviation is more prominent when the desired trajectory changes direction [see Fig. 7(a)] due to the arm's inertia. After learning, the cerebellum output counteracts this inertia, generating higher torques during these changes of the desired trajectory direction [see Fig. 7(d)]. The weight matrix learned by the cerebellum reflects the moments when higher corrective torque values are supplied. By looking at Fig. 7(b) and (d), we can see that the higher corrective torque is produced when the desired trajectory joint coordinates change direction. This occurs in the peaks of the sine waves describing the desired trajectory and corresponds to the activation of the higher and lower input fibers of each block [left and right side of the six weight columns of Fig. 7(c)]. To generate a high corrective torque, the cerebellum must unbalance the magnitude of the positive and negative parts of the joint corrective output [q+ and q− in Fig. 7(b)] which is calculated from the activity of the DCN cells. These DCN cells are grouped by joints. A higher activity affecting positive corrections in a joint produces higher corrective torque. Since PCs inhibit DCN cells, a low PC activity is required for a high DCN activity and vice versa. To obtain a low PC activity, low PF-PC weights are required, which corresponds to small dark squares in Fig. 7(c). Small light squares correspond to high values of the weights. Looking at both sides of the six

weight columns of Fig. 7(c), we can observe how the weight values alternate between high and low in adjacent rows which alternately encode the weights corresponding to the positive and negative parts of each joint corrective torque.

During the learning process, the corrective model is captured in the PF-PC connections. In this way, the movements become more accurate, the error decreases and therefore, also the activity of the IO is reduced. This allows the learned models to become stable once the error reaches appropriate values.

The learning performance is characterized by using four estimates calculated from the mean absolute error (MAE) curve [73]. For the calculation of the MAE of a trajectory execution, we have considered the addition of the error in radians produced by each joint independently.

1) accuracy gain (estimates the error reduction rate comparing the accuracy before and after learning). This estimate helps to interpret the adaptation capability of the cerebellum when manipulating different objects, since the initial MAE for each of these manipulated objects may be different

$$\text{Accuracy Gain} = MAE_{\text{initial}}$$
$$- \left(\frac{1}{n} \sum_{i=0}^{n} MAE_{(\text{final}-i)} \right) \quad n = 30; \quad (8)$$

2) final error (average error over the last 30 trials)

$$\text{Final Error} = \left(\frac{1}{n} \sum_{i=0}^{n} MAE_{(\text{final}-i)} \right) \quad n = 30; \quad (9)$$

3) final error stability (average of standard deviation over the last 30 movement trials)

$$\text{Final Error Stability}$$
$$= \frac{1}{n} \sum_{i=0}^{n} \left(\sigma \left(MAE_{(\text{final}-i)} \right) \right) \quad n = 30; \quad (10)$$

4) error convergence speed (number of samples to reach the final error average)

$$\text{Error Convergence Speed} = j; \text{ where}$$
$$MAE_j \leq \left(\frac{1}{n} \sum_{i=0}^{n} MAE_{(\text{final}-i)} \right) \quad 0 < j \leq \text{final}. \quad (11)$$

We have carried out 70 simulations of a complete training process, where each training process consists of 400 trajectory executions and each trajectory execution is carried out in 1-s simulation time (i.e., the whole system is executed 28 000 times). During each of these training processes, the obtained error in each trajectory execution decreases until it reaches a final stable value. The obtained MAE of a single complete training process is shown in Fig. 8(a). We have tested this learning process with different LTD and LTP components to evaluate how they affect the adaptation capability of the system. From each of these training processes (with different LTD and LTP values), we obtain the performance estimates defined above (accuracy gain, final error, final error stability, and error convergence speed). These performance estimates characterize the adaptation mechanism capability.

As it is shown in Fig. 8(b) and (c), both LTP and LTD must be compensated. Low LTD values combined with high LTP values cause high weight saturation. This can be seen in Fig. 8(c), in which 3-D final normalized error values of the first figure are represented in a high flat surface corresponding to high errors. We also have a flat surface close to zero in Fig. 8(c) (3-D final normalized error stability figure); the cerebellum output is totally saturated. Therefore, when LTP-LTD tradeoff is unbalanced (LTP dominating LTD), the system adaptation capability is low, leading to high error estimators and useless high stability. On the other hand, when high LTD values are combined with low LTP values, this causes low weight saturation. In Fig. 8(c) 3-D plots, we see a good final average error and a good accuracy gain and convergence speed but very unstable output. This is also indicated by the error variance figure estimates which are high in this LTD-LTP area. A compensated LTD-LTP setting drives us to a high-accuracy gain and also, to a low and stable final error with high convergence speed. For instance, if our LTD choice is 0.075, our LTP must be lower than 0.015 to achieve a proper stable learning mechanism. In all the following simulations, we have fixed the LTD and LTP parameters to these values. Therefore, we illustrate how different model deviations (by different object manipulations) can be compensated with a fixed and balanced temporal-correlation kernel and how this correction loop works even in the presence of different sensorimotor transmission delays.

B. Learning Temporal-Correlation Kernel Allows Corrective Model Inference Even in the Presence of Sensorimotor Delays

The cerebellumlike structure previously described works even with sensorimotor delays by means of the temporal-correlation kernel which determines the amount of LTD to be applied. This is summarized in Fig. 9. The results (in Fig. 9) have been obtained after performing four simulations (each one for different delay setups) of 400 trajectory executions each. On the other hand, this temporal-correlation kernel remains robust not only with different unbalanced delays but also with a non-perfect matching between sensorimotor delays and the temporal correlation kernel peak, as it is shown in Fig. 10. These results have been obtained after performing five simulations (each one for a different time deviation) of 400 trajectory executions each.

This robustness is achieved because the scheme is using desired coordinates (positions and velocities) which remain stable across different trials. Nevertheless, with delays mismatching (between learning kernel inherent time shift and sensorimotor delays) over 70 ms, this scheme becomes unstable.

C. Learning Different Dynamic Models

The presented cerebellum microstructure and the long-term plasticity, side by side, facilitate internal model inference. The cerebellum model adapts itself to infer a new model by using error signals which are obtained when manipulating this new object. We study the ability of the cerebellar architecture to infer different corrective models for dynamics changes on a base manipulator model.

Under normal conditions, without adding any extra mass to the end of the effector (arm), the crude inverse dynamics
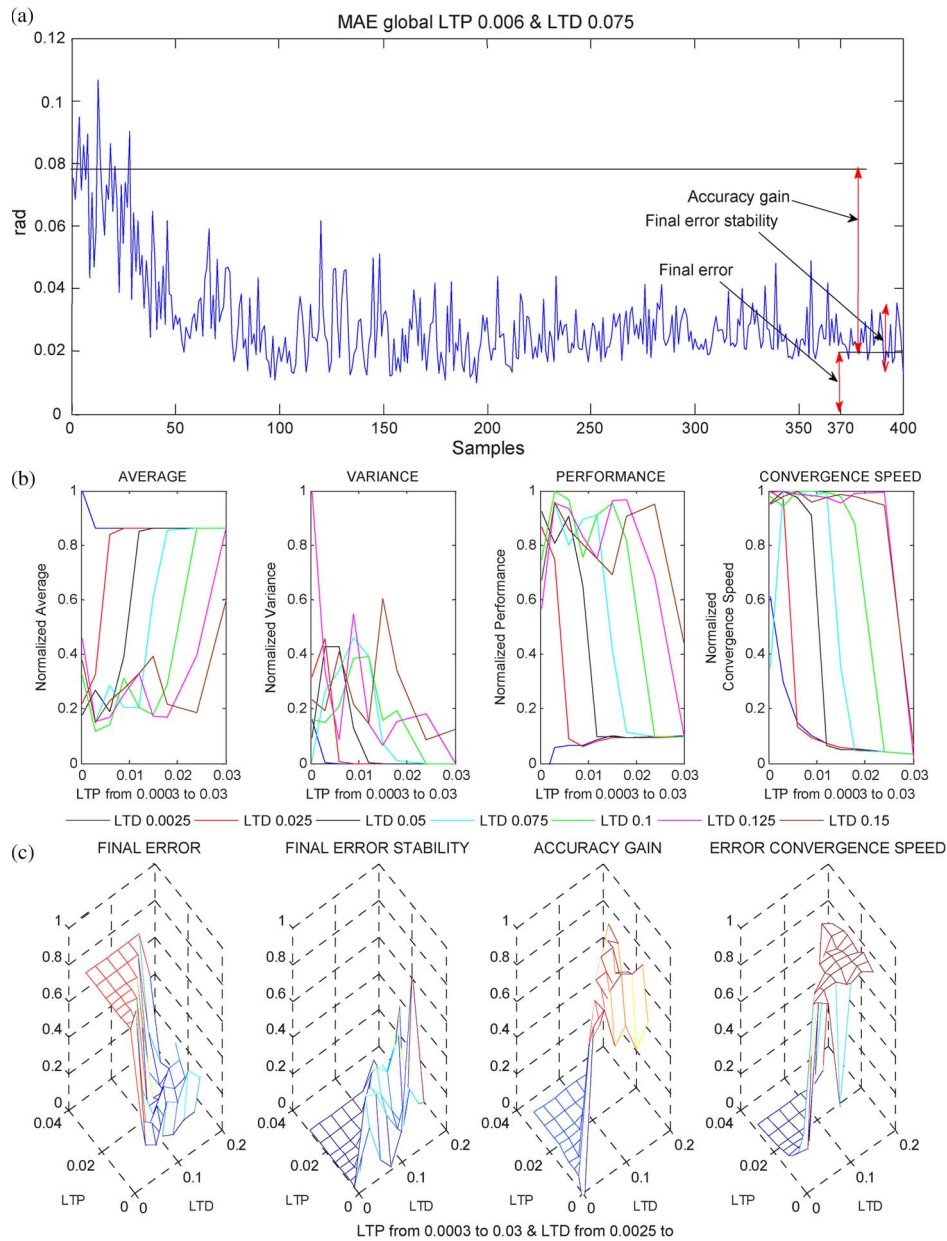
Fig. 8. **Learning characterization**. The error taken into account in this learning characterization is a global addition of the absolute joint errors in position from each link of our robot plant. (a) During the learning process, the movement error decreases along several iterative executions of trials of an eight-like trajectory benchmark. We evaluate the learning performance using four estimates extracted from the MAE curve: 1) accuracy gain; 2) final error; 3) final error stability; and 4) error convergence speed. (b) Using these four estimators, we can evaluate how LTP and LTD affect the learning process. We have conducted multiple simulations with different LTD-LTP tradeoffs to characterize the learning behavior. The goal of an appropriate learning process is to achieve a high-accuracy gain and a low and stable final error.

model calculates rough motor commands to control the arm plant. In contrast, under altered dynamics conditions, the motor commands are inaccurate to compensate for the new undergone forces (inertia, etc.), and this leads to distortions in the performed trajectories. During repeated trials, the cerebellum learns to supply the corrective motor commands when the arm plant model dynamics differs from the initial one. These corrective motor commands are added to the normal-condition motor commands. Then, improved trajectories are obtained as the learning process goes on. The cerebellum gradually builds up internal models by experience and uses them in combination with the crude inverse dynamics controller. This cerebellum adaptation is assumed to involve changes in the synaptic effi-

cacy of neurons constituting the inverse dynamics model [74], as it is shown in our simulation results (Fig. 11).

The performance results of the followed trajectory have been evaluated during 400 trajectory executions manipulating different objects attached at the end of the last segment of the arm of 0.5, 1, 1.5, and 2 kg. Fig. 11 illustrates the performed trajectory for each simulation with an object of a different mass. Fig. 12 shows how the cerebellar model is able to learn/infer the corrective dynamics model for the different objects. The error curves of Fig. 12(a) (where each sample represents the error along one eight-like trajectory) show how the control loop with the adaptive cerebellar module is able to significantly reduce the error during the training process. Fig. 11 shows that
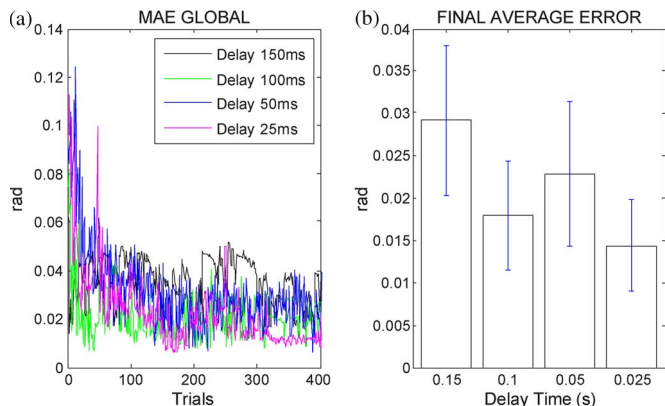
Fig. 9. **Temporal-correlation kernel for different sensorimotor delays (delays from 25 to 150 ms have been tested)**. We have adjusted the correlation kernel peak position to match (see Fig. 6) the sensorimotor delays of the control loop illustrated in Fig. 3. As it is shown, the delay value does not affect to a large extent the obtained performance. The final average error is nearly constant in these different simulations.
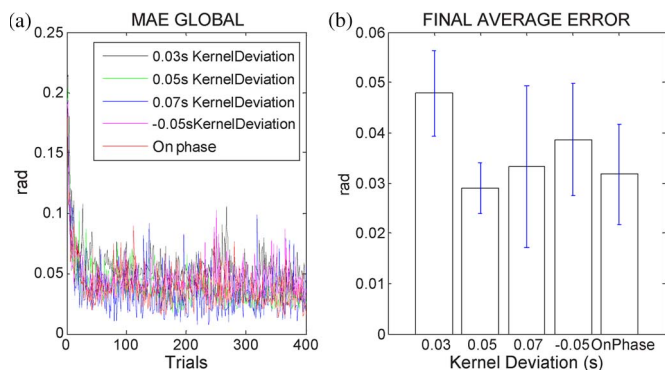

Fig. 10. **Temporal-correlation kernel behavior with different deviations between sensorimotor delays and the kernel peak (deviation from 50 to 70 ms have been tested)**. We have evaluated different deviations between the correlation kernel peak position (see Fig. 6) and the sensorimotor delays of the control loop illustrated in Fig. 3. As it is shown, despite the kernel peak does not exactly match with sensorimotor delays, the cerebellum still works and the final average error keeps on constant. The cerebellum is able to correlate the delayed sinusoidal inputs and the non-in-phase peak kernel.

manipulating heavier objects means that the starting error is higher, since the arm dynamics differ from the original one to a larger extent. Therefore, the cerebellum learns to supply higher corrective torques, which makes a bigger difference between the initial and final error. This makes the accuracy gain estimate higher than in the other cases. On the other hand, for improving the global accuracy gain, higher forces have to be counteracted to follow the desired trajectory.

IV. DISCUSSION

This paper focuses on studying how a cerebellarlike adaptive module, operating together with a crude inverse dynamics model, can effectively provide corrective torque to compensate deviations in the dynamics of a base plant model (due to object manipulation). This is relevant to understand how the cerebellar structure embedded in a biologically plausible control loop can infer internal corrective models when manipulating objects which affect the base dynamics model of the arm. The spiking neural cerebellum connected to a biomorphic robot

plant represents a tool to study how the cerebellar structure and learning kernels (including time shifts for compensating sensorimotor delays) provide adaptation mechanisms to infer dynamics correction models toward accurate object manipulation. Concretely, we have evaluated how a temporal-correlation kernel driving an error-related LTD and a compensatory LTP component (complementing each other) can achieve effective adaptation of the corrective cerebellar output. We have shown how the temporal-correlation kernel can work even in the presence of sensorimotor delays. However, considering the results obtained for several sensorimotor delays, we can state that the desired trajectory must be coded using a univocal population coding in each time step, that is, the codification of the desired position/velocity during the trajectory must be different for each point of the trajectory. And thus, as our cerebellar structure can adaptively generate any suitable output for each trajectory-point codification, the delay of the sensorimotor pathways is not remarkably relevant, even if this delay does not match the intrinsic compensatory delay of the learning integration kernel.

In this simple cerebellarlike structure, we have shown how the representation of the cerebellar weight matrix corresponding to the PF-PC connections can be interpreted in terms of the generated corrective torque (which, in turn, is a direct consequence of this representation). This allows us to study the performance of this corrective model storage and how the changes of the arm dynamics (manipulating different object) are inferred on different synaptic weight patterns.

We have also shown how LTD and LTP need to be balanced with each other to achieve high performance adaptation capabilities. We have studied the behavior of these two complementary adaptation mechanisms. We have evaluated how the learning behaves when they are balanced and also when they are in value ranges in which one of them dominates saturating the adaptation capability of the learning rule. We have evaluated how well-balanced LTD and LTP components lead to an effective reduction of error in manipulation tasks with objects which significantly affect the dynamics of the base arm plant.

We have used a simplified version of the cerebellum to focus on the way that the cerebellar corrective models are stored and structured in neural population weights. This is of interest to inform neurophysiologic research teams to drive attention to potential footprints of inferred models within the PF-PC connections.

As future work, we will study how to dynamically optimize the LTD-LTP integration kernel instead of a single, stable, and balanced LTD-LTP kernel, we will evaluate the capability of improving the adaptation mechanism, shifting this balance to acquire the corrective models faster and then, decrease the plasticity once an acceptable performance is reached. This approach can optimize the learning capability of the system.

We will also develop further real-time interfaces between analog signals and spiking neurons (between the robot and the EDLUT simulator) to perform simulations with real robots and new cerebellar architectures working in a manipulation task scenario in which granular layer, Golgi cells, and stellate cells will be included. This will be addressed in a starting EU project (REALNET).

The neuron models, cerebellar models, and adaptation mechanisms will be available at the EDLUT simulator site to facilitate the reproduction of the presented work.
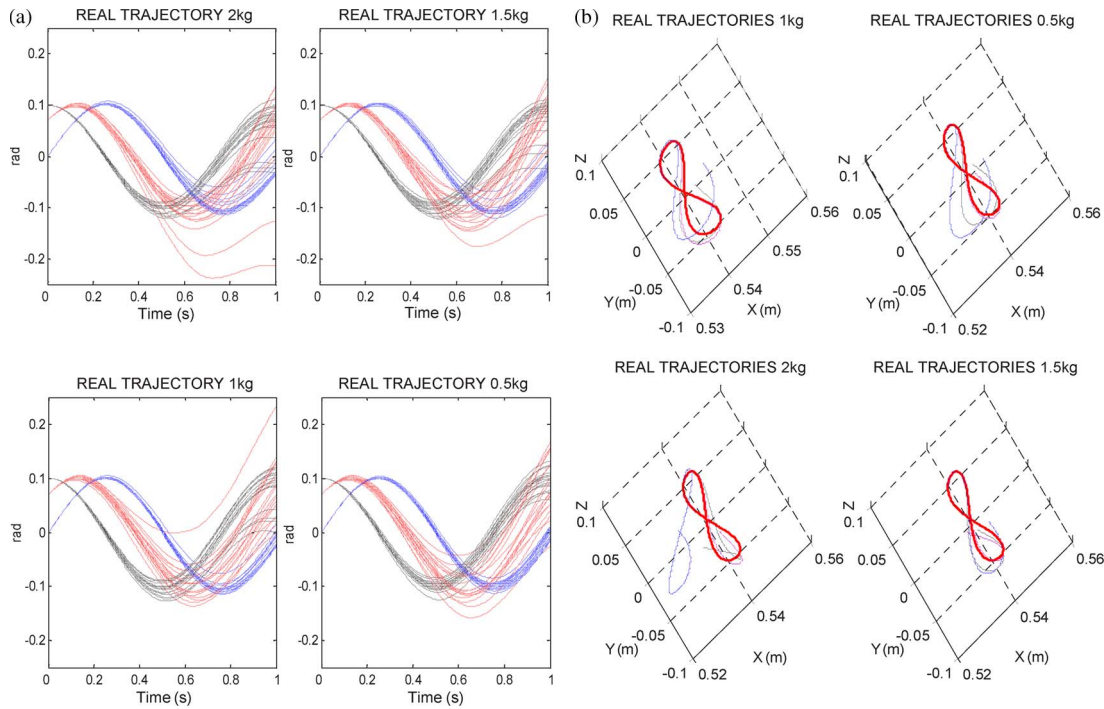
Fig. 11. **Learning the corrective models for the eight-like target trajectory when manipulating objects with different masses (2, 1.5, 1, and 0.5 kg)**. (a) three-joint value representation for the performed trajectory. The three joints are shown. The followed trajectory is shown each 25 trials during a 400-trial complete learning process. (b) 2-D representation of the performed trajectory (Desired trajectory in red; in blue. initial trial; in black, trial number 200; and in cyan, final trial). **Improvement in %: 0.5 kg** 200-trial 40.4% 400-trial 49%; **1 kg** 200-trial 64.6% 400-trial 64.5%; **1.5 kg** 200-trial 72.5% 400-trial 74.4%; **2 kg** 200-trial 78.6% 400-trial 79.3%. **Stability improvement in %** (average std over 0–30 trials/17–200 trials/370–400 trials). **0.5 kg** 170–200-trials 82% compared to initial 0–30-trials, 370–400-trials 60.1% comparing to initial 0–30-trials. **1 kg** 170–200-trials 49.6% compared to initial 0–30 trials, 370–400-trials 42.1% compared initial 0–30-trials. **1.5 kg** 170–200 trials 46.1% compared to initial 0–30 trials, 370–400-trials 26.4% compared to initial 0–30-trials. **2 kg** 170–200 trials 26.4% compared to initial 0–30-trials, 370–400 trials 25.1% compared to initial 0–30 trials.
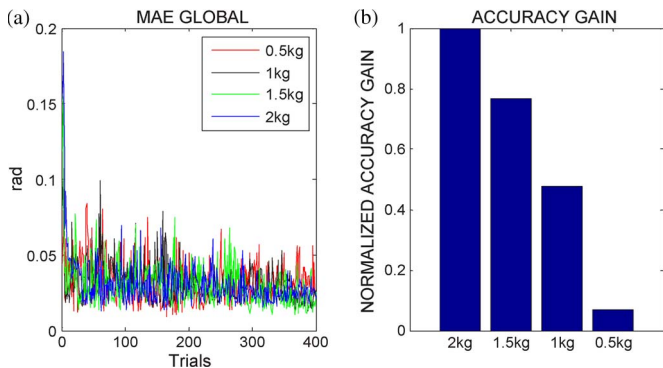

Fig. 12. **Learning performance when manipulating different objects (0.5, 1, 1.5, and 2 kg) during 400-trial learning processes**. (a) MAE evolution. Learning occurs on a continuous basis providing incremental adaptability throughout the simulation time. (b) Accuracy gain.

REFERENCES

[1] H. C. Leiner, A. L. Leiner, and R. S. Dow, "Cognitive and language functions of the human cerebellum," *Trends Neurosci.*, vol. 16, no. 11, pp. 444–447, Nov. 1993.

[2] P. Dean, J. Porrill, C. F. Ekerot, and H. Jörntel, "The cerebellar microcircuit as an adaptive filter: Experimental and computational evidence," *Nat. Rev. Neurosci.*, vol. 11, no. 1, pp. 30–43, Jan. 2010.

[3] E. R. Kandel, J. H. Schwartz, and T. M. Jessell, *Principles of Neural Science*. New York: McGraw-Hill, 2000, ch. 42.

[4] D. Philipona and O. J.-M. D. Coenen, "Model of granular layer encoding in the cerebellum," *Neurocomputing*, vol. 58–60, pp. 575–580, Jun. 2004.

[5] N. Schweighofer, K. Doya, H. Fukai, J. V. Chiron, T. Furukawa, and M. Kawato, "Chaos may enhance information transmission in the inferior olive," *Proc. Nat. Acad. Sci.*, vol. 101, no. 13, pp. 4655–4660, Mar. 2004.

[6] E. D'Angelo, S. K. Koekkoek, P. Lombardo, S. Solinas, E. Ros, J. Garrido, M. Schonewille, and C. I. De Zeeuw, "Timing in the cerebellum: Oscillations and resonance in the granular layer," *Neuroscience*, vol. 162, no. 3, pp. 805–815, Sep. 2009.

[7] S. G. Massaquioi and J. J. E. Stoline, "The intermediate cerebellum may function as a wave-variable processor," *Neurosci. Lett.*, vol. 215, no. 1, pp. 60–64, Aug. 1996.

[8] V. Steuber, D. Willshaw, and A. Van Ooyen, "Generation of time delays. Simplified models of intracellular signaling in cerebellar Purkinje cells network," *Comput. Neural Syst.*, vol. 17, no. 2, pp. 173–191, Jun. 2006.

[9] E. Ros, R. Carrillo, E. M. Ortigosa, B. Barbour, and R. Agís, "Event-driven simulation scheme for spiking neural networks using lookup tables to characterize neuronal dynamics," *Neural Comput.*, vol. 18, no. 12, pp. 2959–2993, Dec. 2006.

[10] J. L. Raymond and S. G. Lisberger, "Neural learning rules for the vestibulo-ocular reflex," *J. Neurosci.*, vol. 18, no. 21, pp. 9112–9129, Nov. 1998.

[11] H. Voicu and M. D. Mauk, "Parametric analysis of cerebellar LTD in eyelid conditioning," *Neurocomput. Comput. Neurosci. Trends Res.*, vol. 69, no. 10–12, pp. 1187–1190, Jun. 2006.

[12] R. R. Carrillo, E. Ros, C. Boucheny, and O.-J. M.-D. Coenen, "A real time spiking cerebellum model for learning robot control," *Biosystems*, vol. 94, no. 1/2, pp. 18–27, Oct.–Nov. 2008.

[13] M. Kawato, "Brain controlled robots," *HFSP J.*, vol. 2, no. 3, pp. 136–142, Jun. 2008.

[14] J. C. Houk, J. T. Buckingham, and A. G. Barto, "Models of cerebellum and motor learning," *Behav. Brain Sci.*, vol. 19, no. 3, pp. 368–383, 1996.

[15] P. Van der Smagt, "Cerebellar control of robot arms," *Connect. Sci.*, vol. 10, no. 3/4, pp. 301–320, Dec. 1998.

[16] R. C. Miall, "The cerebellum, predictive control and motor coordination," in *Proc. Novartis Found. Symp.*, 2007, vol. 218, Sensory Guidance of Movements, pp. 272–290.

[17] S. Khemaissia and A. Morris, "Use of an artificial neuroadaptive robot model to describe adaptive and learning motor mechanisms in the central nervous system," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 28, no. 3, pp. 404–416, Jun. 1998.

[18] D. M. Wolpert, R. C. Miall, and M. Kawato, "Internal models in the cerebellum," *Trends Cognit. Sci.*, vol. 2, no. 9, pp. 338–347, Sep. 1998.

[19] J. S. Albus, "A theory of cerebellar function," *Math Biosci.*, vol. 10, no. 1/2, pp. 25–61, Feb. 1971.

[20] M. Ito, "Cerebellar circuitry as a neuronal machine," *Progr. Neurobiol.*, vol. 78, no. 3–5, pp. 272–303, Feb.–Apr. 2006.

[21] P. Dean and J. Porril, "Adaptive filter models of the cerebellum. Computational analysis," *Cerebellum*, vol. 7, no. 4, pp. 567–571, 2008.

[22] G. A. Jacobson, D. Rokni, and Y. Yarom, "A model of the olivo-cerebellar system as a temporal pattern generator," *Trends Neurosci.*, vol. 31, no. 12, pp. 617–625, Dec. 2008.

[23] G. A. Jacobson, I. Lev, Y. Yarom, and D. Cohen, "Invariant phase structure of olivo-cerebellar oscillations and putative role in temporal pattern generation," *Proc. Nat. Acad. Sci.*, vol. 106, no. 9, pp. 3579–3584, Mar. 2009.

[24] C. I. De Zeeuw, J. I. Simpson, C. C. Hoogenraad, N. Galjart, S. K. Koekkoek, and T. J. Ruigrok, "Microcircuitry and function of the inferior olive," *Trends Neurosci.*, vol. 21, no. 9, pp. 391–400, Sep. 1998.

[25] A. Mathy, S. S. Ho, J. T. Davie, I. C. Duguid, B. A. Clark, and M. Häusser, "Encoding of oscillations by axonal bursts in inferior olive neurons," *Neuron*, vol. 62, no. 3, pp. 388–399, May 2009.

[26] M. Ito and M. Kano, "Long-lasting depression of parallel fiber-Purkinje cell transmission induced by conjunctive stimulation of parallel fibers and climbing fibers in the cerebellar cortex," *Neurosci. Lett.*, vol. 33, no. 3, pp. 253–258, Dec. 1982.

[27] M. Ito, "Cerebellar long-term depression: Characterizations, signal transduction and functional roles," *Physiol. Rev.*, vol. 81, no. 3, pp. 1143–1195, 2001.

[28] N. Schweighofer, J. Spoelstra, M. A. Arbib, and M. Kawato, "Role of the cerebellum in reaching movements in human. II. A neural model of the intermediate cerebellum," *Eur. J. Neurosci.*, vol. 10, no. 1, pp. 95–105, Jan. 1998.

[29] J. Simpson, D. Wylle, and C. I. De Zeeuw, "On climbing fiber signals and their consequences," *Behav. Brain Sci.*, vol. 19, no. 3, pp. 368–383, 1996.

[30] I. Raman and B. P. Bean, "Ionic currents undelying spontaneous action potentials in isolated cerebellar Purkinje neurons," *J. Neurosci.*, vol. 19, no. 5, pp. 1663–1674, Mar. 1999.

[31] E. Bauswein, F. P. Kolb, B. Leimbeck, and F. J. Rubia, "Simple and complex spike activity of cerebellar Purkinje cells during active and passive movements in the awake monkey," *J. Physiol.*, vol. 339, pp. 379–394, Jun. 1983.

[32] J. L. Gardner, S. N. Tokiyama, and S. G. Lisberger, "A population decoding framework for motion after effects on smooth pursuit eye movement," *J. Neurosci.*, vol. 24, no. 41, pp. 9035–9048, Oct. 2004.

[33] R. Soetedjo and A. F. Fuchs, "Complex spike activity of Purkinje cells in the oculomotor vermis during behavioral adaptation of monkey saccades," *J. Neurosci.*, vol. 26, no. 29, pp. 7741–7755, Jul. 2006.

[34] E. D'Angelo, T. Nieus, M. Bezzi, A. Arleo, and O. J.-M. D. Coenen, "Modeling synaptic transmission and quantifying information transfer in the granular layer of the cerebellum," *Lecture Notes Comp. Sci.*, vol. 3512, pp. 107–114, 2005a.

[35] M. Kawato and H. Gomi, "A computational model of four regions of the cerebellum based on feedback-error learning," *Biol. Cybern.*, vol. 68, no. 2, pp. 95–103, 1992.

[36] C. D. Hansel, D. J. Linden, and E. D'Angelo, "Beyond parallel fiber LTD: The diversity of synaptic and non-synaptic plasticity in the cerebellum," *Nat. Neurosci.*, vol. 4, no. 5, pp. 467–475, May 2001.

[37] D. Marr, "A theory of cerebellar cortex," *J. Physiol.*, vol. 202, no. 2, pp. 437–470, Jun. 1969.

[38] M. Ito, *The Cerebellum and Neural Control*. New York: Raven, 984.

[39] J. Spoelstra, M. A. Arbib, and N. Schweighofer, "Cerebellar adaptive control of a biomimetic manipulator," *Neurocomputing*, vol. 26/27, pp. 881–889, Jun. 1999.

[40] D. M. Wolpert and Z. Ghahramani, "Computational principles of movement neuroscience," *Nat. Neurosci.*, vol. 3, no. Suppl., pp. 1212–1217, Nov. 2000.

[41] S. Kuroda, K. Yammoto, H. Miyamoto, K. Doya, and M. Kawato, "Statistical characteristics of climbing fiber spikies necessary for efficient cerebellar learning," *Biol. Cybern.*, vol. 84, no. 3, pp. 183–192, Mar. 2001.

[42] E. Ros, E. M. Ortigosa, R. Agís, R. Carrillo, and M. Arnold, "Real-time computing platform for spiking neurons (RT-spike)," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 1050–1063, Jul. 2006.

[43] R. Agís, J. Díaz, E. Ros, R. R. Carrillo, and E. M. Ortigosa, "Hardware event-driven simulation engine for spiking neural networks," *Int. J. Electron.*, vol. 94, no. 5, pp. 469–480, May 2007.

[44] J. Butterfaß, M. Grebenstein, H. Liu, and G. Hirzinger, "DLR hand II: Next generation of a dextrous robot hand," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2001, pp. 109–114.

[45] R. E. Kettner, S. Mahamud, H.-C. Leung, N. Sitkoff, J. C. Houk, B. W. Peterson, and A. G. Barto, "Prediction of complex two-dimensional trajectories by a cerebellar model of smooth pursuit eye movement," *J. Neurophysiol.*, vol. 77, no. 4, pp. 2115–2130, Apr. 1997.

[46] A. Haith and S. Vijayakumar, "Robustness of VOR and OKR adaptation under kinematics and dynamics transformations," in *Proc. 6th IEEE ICDL*, London, U.K., 2007, pp. 37–42.

[47] C. Boucheny, R. R. Carrillo, E. Ros, and O. J.-M. D. Coenen, "Real-time spiking neural network: An adaptive cerebellar model," *LNCS*, vol. 3512, pp. 136–144, 2005.

[48] E. Nakano, H. Imamizu, R. Osu, Y. Uno, H. Gomi, T. Yoshioka, and M. Kawato, "Quantitative examinations of internal representations of arm trajectory planning. Minimum commanded torque change model," *J. Neurophysiol.*, vol. 81, no. 5, pp. 2140–2155, May 1999.

[49] E. J. Hwang and R. Shadmehr, "Internal models of limb dynamic and the encoding of limb state," *J. Neural Eng.*, vol. 2, no. 3, pp. S266–S278, Sep. 2005.

[50] E. Todorov, "Optimality principles in sensorimotor control (review)," *Nat. Neurosci.*, vol. 7, no. 9, pp. 907–915, Sep. 2004.

[51] M. Kawato, K. Furukawa, and R. Suzuki, "A hierarchical neural-network model for control and learning of voluntary movement," *Biol. Cybern.*, vol. 57, no. 3, pp. 169–185, 1987.

[52] J. Porrill, P. Dean, and J. V. Stone, "Recurrent cerebellar architecture solves the motor-error problem," *Proc. Biol. Sci.*, vol. 271, no. 1541, pp. 789–796, Apr. 2004.

[53] B. B. Andersen, L. Korbo, and B. Pakkenberg, "A quantitative study of the human cerebellum with unbiased stereological techniques," *J. Comp. Neurol.*, vol. 326, no. 4, pp. 549–560, Dec. 1992.

[54] C. Assad, S. Dastoor, S. Trujillo, and L. Xu, "Cerebellar dynamic state estimation for a biomorphic robot arm," in *Proc. Syst., Man. Cybern.*, Oct. 2005, vol. 1, pp. 877–882.

[55] E. D'Angelo, P. Rossi, D. Gall, F. Prestori, T. Nieus, A. Maffei, and E. Sola, "LTP of synaptic transmission at the Mossy Fiber-Granule cell relay of the cerebellum," *Progress in Brain Research*, vol. 128, pp. 69–80, 2005.

[56] W. Gerstner and W. Kistler, *Spiking Neuron Models: Singleneurons, Populations, Plasticity*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[57] D. Jaeger, "No parallel fiber volleys in the cerebellar cortex: Evidence from cross-correlation analysis between Purkinje cells in a computer model and in recordings from anesthetized rats," *J. Comput. Neurosci.*, vol. 14, no. 3, pp. 311–327, May/Jun. 2003.

[58] A. Roth and M. Häusser, "Compartmental models of rat cerebellar Purkinje cells based on simultaneous somatic and dendritic patch-clamp recordings," *J. Physiol.*, vol. 535, pp. 445–472, Sep. 2001.

[59] S. Solinas, R. Maex, and E. De Schutter, "Synchronization of Purkinje cell pairs along the parallel fibre fiber axis: A model," *Neurocomputing*, vol. 52–54, pp. 97–102, Jun. 2003.

[60] D. Jaeger, E. De Schutter, and J. Bower, "The role of synaptic and voltage-gated currents in the control of Purkinje cell spiking: A modeling study," *J. Neurosci.*, vol. 17, no. 1, pp. 91–106, Jan. 1997.

[61] M. Häuser and B. Clarck, "Tonic synaptic inhibition modulates neuronal output pattern and spatiotemporal synaptic integration," *Neuron*, vol. 19, no. 3, pp. 665–678, Sep. 1997.

[62] R. S. Sutton and A. G. Barto, "Toward a modern theory of adaptive networks: Expectation and prediction," *Psychol. Rev.*, vol. 88, no. 2, pp. 135–170, Mar. 1981.

[63] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-13, no. 5, pp. 934–846, Sep./Oct. 1983.

[64] V. Lev-Ram, S. B. Meht, D. Kleinfeld, and R. Y. Tsien, "Reversing cerebellar long term depression," *Proc. Nat. Acad. Sci.*, vol. 100, no. 26, pp. 15 989–15 993, Dec. 2003.

[65] M. Fujita, "Adaptive filter model of the cerebellum," *Biol. Cybern.*, vol. 45, no. 3, pp. 195–206, 1982.

[66] T. J. Sewjnowski, "Storing covariance with nonlinearity interacting neurons," *J. Math. Biol.*, vol. 4, no. 4, pp. 303–321, Oct. 1977.

[67] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

[68] J. G. Keating and W. T. Thach, "Nonclock behavior of inferior olive neurons. Interspike interval of Purkinje cell complex spike discharge in the awake behaving monkey is random," *J. Neurophysiol.*, vol. 73, no. 4, pp. 1329–1340, Apr. 1995.

[69] A. Linares-Barranco, M. Oster, D. Cascado, G. Jiménez, A. Civit, and B. Linares-Barranco, "Inter-spike-intervals analysis of AER Poisson-like generator hardware," *Neurocomputing*, vol. 70, no. 16–18, pp. 2692–2700, Oct. 2007.

[70] J. Spoelstra, N. Schweighofer, and M. A. Arbib, "Cerebellar learning of accurate predictive control for fast-reaching movements," *Biol. Cybern.*, vol. 82, no. 4, pp. 321–333, Apr. 2000.

[71] J. L. Contreras-Vidal, S. Grossberg, and D. Bullock, "A neural model of cerebellar learning for arm movement control: Cortico-spino-cerebellar dynamics," *Learn. Memory*, vol. 3, no. 6, pp. 475–502, Mar./Apr. 1997.

[72] D. Purves, G. J. Augustine, and D. Fitzpatrick, *Neuroscience*, 2nd ed. Sunderland, MA: Sinaver Associates Inc., 2001.

[73] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Clim. Res.*, vol. 30, no. 1, pp. 79–82, Dec. 2005.

[74] M. Kawato and D. M. Wolpert, "Internal models for motor control," in *Proc. Novartis Found. Symp.*, 1998, vol. 218, pp. 291–307.

Niceto Rafael Luque received the B.S. degree in electronics engineering and the M.S. degree in automation and industrial electronics from the University of Córdoba, Córdoba, Spain, in 2003 and 2006, respectively, and the M.S. degree in computer architecture and networks from the University of Granada, Granada, Spain, in 2007.

He is currently participating in an EU project related to adaptive learning mechanisms and bio-inspired control. His main research interests include biologically processing control schemes, lightweight robot, and spiking neural networks.

Jesús Alberto Garrido received the M.S. degree in computer science and the M.S. degree in computer architecture and networks from the University of Granada, Granada, Spain, in 2006 and 2007, respectively.

He is currently participating in an EU project related to adaptive learning mechanisms and bio-inspired control. His main research interests include biologically processing control schemes, lightweight robots, and spiking neurons.

Richard Rafael Carrillo received the Ph.D. degree in computer and electronics engineering from the University of Cagliari, Cagliari, Italy, in 2008, and in computer science from the University of Granada, Granada, Spain, in 2009.

He is currently a Researcher in the Department of Computer Architecture and Electronics, University of Almeria, Almeria, Spain. He is interested in efficient implementations for the simulation of spiking neural networks and models of neural circuits which could be used in engineering (e.g., robot control).

Olivier J.-M. D. Coenen received the Ph.D. degree jointly from the Department of Physics, Physics/Biophysics Program, University of California at San Diego (UCSD), La Jolla, and Computational Neurobiology Laboratory, Howard Hughes Medical Institute, The Salk Institute for Biological Studies, La Jolla, CA, in 1998.

His research interests include developing theories and models of neural networks.

Eduardo Ros received the Ph.D. degree from the University of Granada, Granada, Spain, in 1997.

He is currently Full Professor in the Department of Computer Architecture and Technology, University of Granada, where he is currently the responsible Researcher of two European projects related to bio-inspired processing schemes. His research interests include simulation of biologically plausible processing schemes, hardware implementation of digital circuits for real-time processing in embedded systems, and high-performance computer vision.