International Journal of Neural Systems, Vol. 21, No. 5 (2011) 385–401
© World Scientific Publishing Company
DOI: 10.1142/S0129065711002900

ADAPTIVE CEREBELLAR SPIKING MODEL EMBEDDED IN THE CONTROL LOOP: CONTEXT SWITCHING AND ROBUSTNESS AGAINST NOISE

N. R. LUQUE[*,‡], J. A. GARRIDO[*,§], R. R. CARRILLO[†,¶],
S. TOLU[*,‖] and E. ROS[*,**]
[*]*Department of Computer Architecture and Technology*
CITIC, University of Granada
Periodista Daniel Saucedo s/n, Granada, Spain

[†]*Department of Computer Architecture and Electronics*
University of Almería, Ctra. Sacramento s/n
La Cañada de San Urbano, Almería, Spain
[‡]*nluque@atc.ugr.es*
[§]*jgarrido@atc.ugr.es*
[¶]*rcarrillo@atc.ugr.es*
[‖]*stolu@atc.ugr.es*
[**]*eros@atc.ugr.es*

This work evaluates the capability of a spiking cerebellar model embedded in different loop architectures (recurrent, forward, and forward&recurrent) to control a robotic arm (three degrees of freedom) using a biologically-inspired approach. The implemented spiking network relies on synaptic plasticity (long-term potentiation and long-term depression) to adapt and cope with perturbations in the manipulation scenario: changes in dynamics and kinematics of the simulated robot. Furthermore, the effect of several degrees of noise in the cerebellar input pathway (mossy fibers) was assessed depending on the employed control architecture. The implemented cerebellar model managed to adapt in the three control architectures to different dynamics and kinematics providing corrective actions for more accurate movements. According to the obtained results, coupling both control architectures (forward&recurrent) provides benefits of the two of them and leads to a higher robustness against noise.

Keywords: Cerebellum; STDP; robot simulation; learning; biological control system; noise.

1. Introduction

The efficiency and complexity of animal movement suggest that the biological motor controller does not consider the articulated animal limbs as strings of independent linked bodies. The necessary force for coordinated animal movements, such as reaching and walking, is smartly and conjointly determined for each joint. Due to the nonlinear relationship between joint forces and limb movements and the need to satisfy certain constraints on a movement in different controlling scenarios, biology seems to use advanced controlling mechanisms of interest also to advanced robotics.

Although numerous details of cerebellar microcircuitry have been determined, the functional contribution of the cerebellum to the motor system function remains an open issue. The complexity and the sophistication of the primate motor control system are overwhelming. This motor control is highly multi-dimensional and non-linear, making its characterization troublesome.[1] However, the cerebellum is commonly supposed to be responsible for timing, fine-tuning, and coordinating the motor system.[2–4] It is fair to think that emulating the functionality of the cerebellar microcircuitry would allow the control of non-stiff-joint "robotic arms" properly driven with relatively low-power actuators (as is the case of biological counterparts).[5] Very diverse models have been proposed and evaluated.[6–11] The cerebellum is usually divided into three parts. One region that is mainly associated with the vestibular system, another part related with the brainstem and spinal cord, and a third region, the cerebrocerebellum, that has extensive interconnections with the cerebral cortex and is likely to be involved in motor coordination.[12] However, this knowledge about the cerebellar cortex has not been used in robotics as successfully as in biology.

The control function of any individual region of the cerebellum relies both on its internal microcircuitry and on the way it is connected with other parts of the motor system.[13,14] These connections and their functionality still remain an open issue.[15]

The cerebellum is involved in a feedback loop for muscle control. When the cortex sends a message for motor movement to the lower motor neurons in the brain stem and to the spinal cord, it also sends a copy of this message to the cerebellum. This is transmitted from the pyramidal fibers in the cortex on the cortico-pontine-cerebellar tract to the cerebellum. Additionally, the cerebellum also receives information from muscle spindles, joints and tendons.[16] Therefore, the cerebellum receives motor commands and also, actual sensory signals. This allows the extraction of corrective models on the "manipulating arm" for accurate movements. Traditionally, when considering the control system, it is assumed that the efference copy of motor commands predicts the sensory consequences of actions, including the sensorimotor pathways delays (the spinal cord inverse model transforms the torques into muscle tension) and also allowing its integration with the sensory information related with the actual state.[17] The concept of internal feedback from an internal model of the arm (or body)[18–20] has been extensively accepted (known as the forward model and formed in the cerebellum via the cerebrocerebellar communication loop).

Furthermore, a wide range of cerebellar motor-control-system approaches has been developed. This is a very active research field (for a review, please, refer to Ref. 10).

By using a forward model combined with an inverse dynamics model, the efference copy of the motor command output from the inverse model can be used as an input for a forward model. A forward dynamics transformation is able to predict the dynamics of the muscles from the state of the system and therefore, can be used to compute a controller output.

On the other hand, it has been recently suggested that cerebellar microzones typically receive mossy fiber (MFs) inputs that are related to the outputs of those microzones.[21] This configuration leads to a rather modular scheme. This modularity seems to facilitate the potential role of the cerebellum in adding corrective signals on the sensory space rather than onto motor signals. There are biological evidences that suggest that motor cortex functionality is heterogeneous allowing both control possibilities (addition of corrective terms in both the sensory and motor space).[22] The cerebellum computing corrective terms in the sensory space have motivated some authors to suggest a different cerebellar control loop which is called recurrent model.[14,21]

This biologically inspired cerebellar architecture based on the cerebellar connectivity can deal with the so-called distal error problem. The natural error signal for learning motor commands is the difference between actual and correct commands ('motor error'). However, in autonomous systems, the correct command is typically unknown. Only information about the sensory consequences of incorrect commands is available, which leads to an error representation (based on sensory signals). This is related to the motor error; however, this relation may be complex. Therefore, sensory-based error estimations are called 'distal errors'. How to use this information to drive motor learning is the distal error (or motor error) problem.

These two cerebellar architectures have been proposed as biologically-inspired approaches. Thus, it is fair to think that both architectures may co-exist and work together in the cerebellum developing a complementary functionality (see Fig. 1). This is the main issue under study in this paper.

The configuration illustrated in Fig. 1 has remarkable analogies with the classical inner loop/outer loop control architecture (see Fig. 2).

The inner loop/outer loop architecture groups many classical robot-control strategies from the literature.[23,24] This separation of the inner loop and outer loop terms is important for several reasons; in

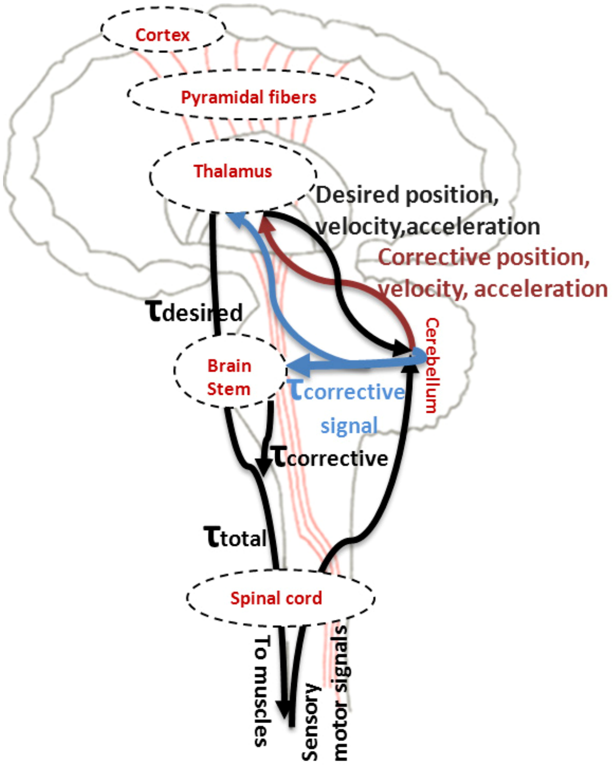


Fig. 1. Biological circuitry projection of the recurrent-forward control loops.

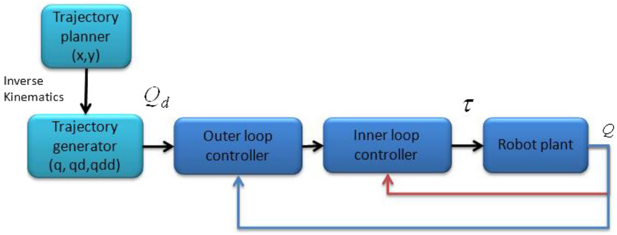


Fig. 2. Inner/outer control loop.

the inner loop, the calculation of the torque commands (non-linear terms) is computed to achieve a better precision and computation speed. Besides, the structure of the inner loop control remains fixed; what control designers may modify more freely to customize the control system architecture is mainly in the outer loop. Thus, the outer loop can be totally modified without restrictions to achieve several other goals without the need to modify the dedicated inner loop control. For instance, additional compensation terms may be included in the outer loop to enhance robustness to parametric uncertainty, unknown dynamics, and external disturbances or tracking of task space trajectories instead of joint space trajectories, regulating both motion and force. Drawing an analogy between the inner/outer control loop and the presented composed control architecture, the inner loop corresponds to the forward architecture that supplies the torque corrections and the outer control loop corresponds to the recurrent architecture that supplies the position/velocity corrections.

This paper studies how an adaptive spiking cerebellum-like module which includes long-term depression (LTD) and long-term potentiation (LTP) at parallel-fiber to Purkinje-cell synapses (PF-PC) is embedded in diverse control loops (forward, recurrent, and a combination of both architectures) to infer corrective models which compensate deviations in the robot trajectory when the dynamics and kinematics of the controlled robotic arm are altered and noise (related to the inherent noise of the muscle spindle signal) is introduced in the cerebellar input (MFs).[25,26] The main goal of this work is a comparative evaluation of these control architectures which shows how forward and recurrent architectures complement each other in the framework of a manipulation task and how robustly they behave in the presence of noise.

2. Methods

As was exposed in the *introduction section*, nowadays, biologically inspired neural processing is an open issue where spiking neural networks play a fundamental role.[27–36] For a comprehensive review on spiking neural networks, please, refer to Ref. 37.

For extensive spiking network simulations, an advanced event-driven simulator based on lookup tables (EDLUT) has been further developed and used.[38,39]

For the robot plant simulation and the evaluated control loops, an interface between the EDLUT and the simulator of the LWR (Light-Weight Robot) developed at DLR (German Aerospace Center)[40] has been implemented. In this way, we were able to evaluate robotic movements of the LWR manipulating different objects that significantly affected the dynamics and kinematics of the robotic arm.

2.1. *Robotic arm simulator*

Different control loops have been integrated within the robot plant simulator of the LWR.[40] The simulated-robot-plant physical characteristics can be dynamically modified to match different contexts. The LWR robot is a 7-Degrees-of-Freedom (DOF) arm consisting of revolute joints. For the sake of simplicity, in our experiments, the number of actual joints (degrees of freedom) has been reduced to three. Specifically, the first (we will refer to it as Q_1), second (Q_2), and fifth joint (Q_3) have been used and the others have been kept fixed. This robot is especially suited for interactions with humans. In this scenario, the use of robots based on low-power actuators in order to reduce danger for humans in case of malfunctioning is of special interest. Furthermore, the accuracy in position and trajectory is not fully exploitable because of the dynamically changing interaction characteristics (non-static and structured scenario as is the case in other robotic control application fields). Figure 3 includes a simple scheme of the robot arm indicating the three non-fixed joints used in our experiments.

2.2. *Control scheme*

Lee Miller in Ref. 41 proposes a cerebellar control system based on a predictive signal (supplied by the cerebellum) with the aim of giving progressive and proper motor control commands. According to this approach, our first cerebellar control-loop model has been developed as a forward (FD) cerebellar control.

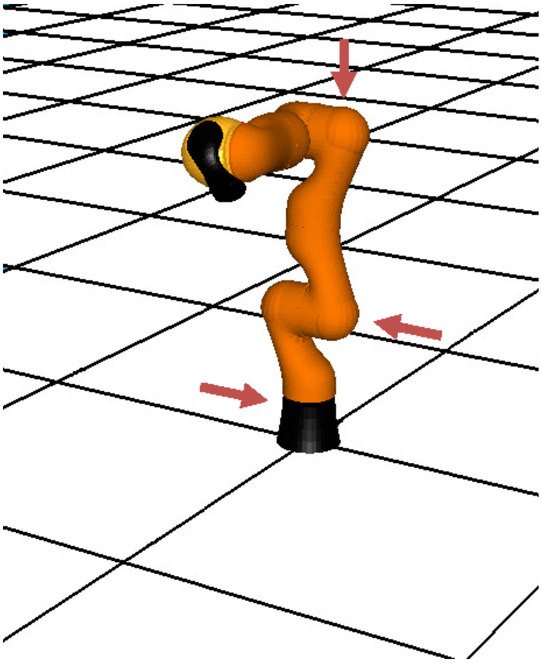


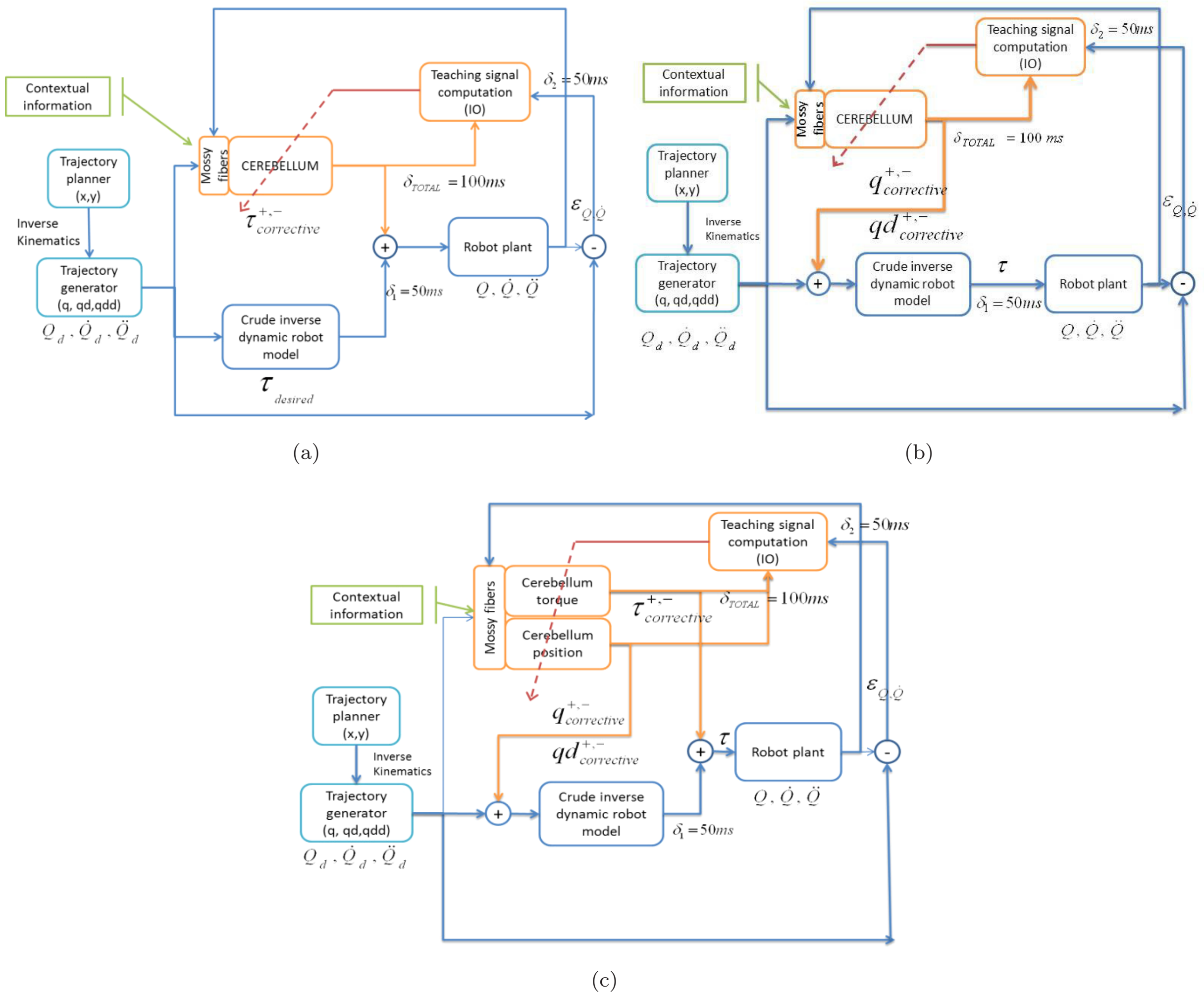
Fig. 3. LWR robot-arm. The three joints used in our experiments are explicitly indicated, all other joints are fixed.

In this control loop (see Fig. 4(a)), the desired arm states (robot end-effector position at each time) are generated by the trajectory planner to follow the desired trajectory. This trajectory in Cartesian coordinates is translated into joint coordinates (positions (Q), velocities ($\dot{Q}$), and accelerations ($\ddot{Q}$)) by the trajectory generator that consists of a crude inverse kinematic model representing the output of the motor cortex and other motor areas (while motor cortex provides a basic command which is appropriate for slow single-joint movements, the cerebellum provides the necessary correction for multi-joint movements).[26] In our experiment, the robot follows the trajectory described in Eq. (1) (see Fig. 5).

$$Q_1 = 0.1\sin(\pi t), \quad Q_2 = 0.1\sin(\pi t + \theta), \quad Q_3 = 0.1\sin(\pi t + 2\theta). \tag{1}$$

In the forward architecture, these desired arm states in joint coordinates are used at each time step to compute crude torque commands (crude inverse dynamic robot model). They are also used together with the contextual information (which could be obtained through visual, haptic information or cognitive "labels" as model profiles) related to the manipulated object, as input to the cerebellum model which produces the predictive corrective commands ($\tau_{\text{corrective}}$) that are added to these crude torque commands (τ_{desired}). Total torque (τ) is delayed (on account of delays of the biological motor pathways, this is δ_1 in Fig. 4) and supplied to the robot plant. The difference (ε) between the actual robot trajectory and the desired one is also delayed (δ_2 in Fig. 4) and used by the teaching signal computation module to calculate the inferior olive (IO) activity that reaches the cerebellum through the climbing fibers. This signal will be used by the cerebellum to adapt its output as described in the learning process section.

On the other hand, the presented recurrent architecture helps the cerebellum to find out temporal regularities in trajectory distortions. In this way, the cerebellum is able to compute predictive corrective position and velocity commands to compensate the deviation caused by the dynamic and kinematic modifications on the base-robot arm.



(a)

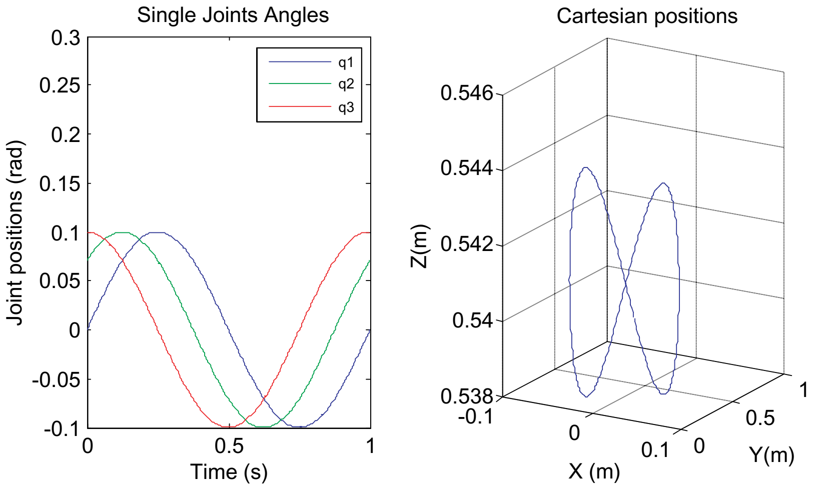
(b)

(c)

Fig. 4. Forward (FD), recurrent (RR), and forward&recurrent control loop (FD&RR). (a) In this FD control loop, the adaptive cerebellar module is embedded in the forward control loop and delivers add-on corrective torques to compensate deviations in the base dynamics and kinematics of the robotic arm model when manipulating different objects. (b) Recurrent control loop, the adaptive cerebellar model infers a model from the error signal related to a sensorimotor input to produce effective corrective position and velocity add-on terms. In this way, instead of propagating data from input to output as the forward architecture does, the recurrent architecture also propagates data from later processing stages to earlier ones. (c) FD&RR control loop delivers add-on corrective actions to compensate deviations in the base dynamic and kinematic robotic arm model when manipulating objects. In this forward&recurrent control loop, the adaptive cerebellar modules infer a model of effective corrective position, velocity, and torque add-on terms from the error signal related to sensorimotor input.

According to this hypothesis and based on the control loop described in Ref. 14, the recurrent control architecture shown in Fig. 4(b) has been developed.

In the recurrent architecture (RR), the arm states in joint coordinates are also used together (joint related information) with the contextual information (related to the manipulated object) as input to the cerebellum which produces the predictive corrective position and velocity commands ($q_{\text{corrective}}$, $qd_{\text{corrective}}$) which are added to the desired position and velocity trajectory commands. The final total torque computed by the crude inverse dynamics and the error signal are handled in the same way as the previously-presented forward architecture.



(a) (b)

Fig. 5. Three-joint periodic trajectory describing 8-shaped movements: (a) Angular coordinates of each joint of the Light Weight Robot, (b) 3D view of the robot end-effector trajectory in Cartesian coordinates. This 8-like trajectory ensures a sufficiently rich movement that allows robot arm dynamics to be revealed.[42] The interaction torque values generated in a multi-joint movement demand a more complex cerebellar control task than a summed combination of single-jointed movements.

These two cerebellar architectures have been proposed as biologically-inspired approaches; thus, it is interesting to study their potential complementary role in a control correcting scenario. Therefore, the developed forward&recurrent (FD&RR) architecture as presented in Fig. 4(c) will be evaluated.

Under normal conditions, without extra mass added to the robot end effector, the "crude inverse dynamic robot" module calculates rough motor commands to control the robotic arm. Under altered dynamics conditions, in contrast, the rough motor commands are very inaccurate to compensate for the new undergone forces (inertia, etc.), and this leads to distortions in the performed trajectories. During repeated trials, the cerebellar model is able to learn a corrective dynamics model for each manipulated object and supplies:

(a) Corrective motor torques in FD architecture.
(b) Corrective trajectory positions and velocities in RR architecture.
(c) Corrective motor torques and corrective trajectory position and velocities in FD&RR architecture.

2.3. *EDLUT: Spiking neuron simulator*

EDLUT is an open software platform which allows fast event-driven simulation of relatively-complex neural networks through an innovative method: the neural network[43] simulations are split into the two stages; Cell behavior characterization: each neural model included in the network (usually defined by a set of differential equations which govern the neural state) is simulated for every possible neural state and the consequent evolution of each neural state variable is stored in lookup tables. Then, in a second stage, when a simulation of a network containing these models is required, it can be performed without requiring a computationally-costly numerical procedure for solving the differential equations defining the neural model. EDLUT is used for the simulation of the embedded cerebellar module.

2.4. *Neural models*

The simulated spiking network consists of two different integrate-and-fire (I&F) cell types.[43] The used cell models are a modified version of the spike-response model (SRM) with synapses modeled as input-driven conductance.[44,45] Thus, the neuron models account for dynamic synaptic conductance rather than simply for constant current flows, providing an improved description over simpler I&F models.[46]

The synaptic conductance follows a decaying exponential function triggered by input spikes as stated in Eq. (2):

$$g_{\text{exc}}(t) = \begin{cases} 0, & t < t_0 \\ g_{\text{exc}}(t_0)e^{-\frac{(t-t_0)}{\tau_{\text{exc}}}} & t \geq t_0 \end{cases} \tag{2a}$$

$$g_{\text{inh}}(t) = \begin{cases} 0, & t < t_0 \\ g_{\text{inh}}(t_0)e^{-\frac{(t-t_0)}{\tau_{\text{inh}}}} & t \geq t_0 \end{cases} \tag{2b}$$

where g_{exc} and g_{inh} represent the excitatory and inhibitory synaptic conductance. τ_{exc} and τ_{inh} represent the corresponding synaptic time constants. Finally t_0 represents the last spike time arrival (already computed).[38] This exponential representation has several advantages. Firstly, it is an effective representation of realistic synaptic conductance. Secondly, each synapse type requires only a single state variable per neuron, because the effect of input spikes through several synapses of the same type can simply be recursively summed when updating the total conductance if they have the same time constants. Therefore, when an input spike is received at time t, for example, through an excitatory synapse, its corresponding conductance $g_{\text{exc(pre-spike)}}(t)$ is abruptly incremented in a term $G_{\text{exc},j}$ as described in Eq. (3):

$$g_{\text{exc(post-spike)}}(t) = G_{\text{exc},j} + g_{\text{exc(pre-spike)}}(t). \tag{3}$$

$\text{G}_{\text{exc},j}$ is the weight of synapse j (a similar relation holds for inhibitory synapses) and $g_{\text{exc(pre/post-spike)}}$ represents the excitatory synaptic conductance before (*pre*) and after (*post*) the spike arrival, respectively.

In our simulations, the synaptic parameters have been chosen to represent excitatory AMPA-receptor-mediated synapse time constants and inhibitory GABAergic synapse time constants of cerebellar granule cells.[47–51] Note that different cells might have different parameters (Table 1).[52–55]

The neuron membrane potential V_m at time t is defined by differential Eq. (4).

$$C_m \frac{dV_m}{dt} = g_{\text{exc}}(t)(E_{\text{exc}} - V_m) + g_{\text{inh}}(t)(E_{\text{inh}} - V_m) + G_{\text{rest}}(E_{\text{rest}} - V_m). \tag{4}$$

Where the conductance values $g_{\text{exc}}(t)$ and $g_{\text{inh}}(t)$ integrate all the contributions received through individual excitatory and inhibitory synapses respectively, G_{rest} represents the resting conductance, and E_{exc}, E_{inh}, and E_{rest} represent the corresponding reversal potentials. Equation (4) is amenable to numerical analysis. In this way, V_m, g_{exc}, and g_{inh}, can be calculated for a given time after a previous neural state or input spike allowing the event-driven simulation scheme. The firing time (t_f) is the time when the membrane potential (V_m) reaches the firing threshold (V_{th}) and an output spike is emitted. It can be calculated from the membrane potential evolution.

Table 1. Parameters of the cell types.

Parameter	Granule cell	Purkinje cell
Refractory period	1 ms	2 ms
Membrane capacitance	2 pF	400 pF
*Total excitatory peak conductance	1 nS · 100	1.3 nS· ·175000 · 10%*
Total inhibitory peak conductance	1 nS · 200	3 nS · 150
Threshold	−40 mV	−52 mV
Resting potential	−70 mV	−70 mV
Resting conductance	0.2 nS	16 nS
Resting time constant (τ_m)	10 ms	25 ms
Excitatory-synapse time constant (τ_{exc})	0.5 ms	0.5 ms
Inhibitory-synapse time constant (τ_{inh})	10 ms	1.6 ms

Note: Parameters obtained from the following papers: Granule cell (GrC)[47–51] and Purkinje cell (PC)[52–55].
*Where 10% means the ratio of active connections PF-PC (out of the total 175000 PFs).

Table 1 shows the equation parameters corresponding to the two neural models used in the simulated cerebellum.

2.5. *Cerebellum model*

Two different cerebellar module configurations based on the scheme of Fig. 6 have been used. The first one corresponds to the previously called forward control architecture providing corrective torque terms and the second one corresponds to the recurrent control architecture providing corrective terms in the sensory space. Here we briefly indicate the different cerebellar module layers:

(i) *Mossy fibers* (256 *fibers*) (*MFs*): Mossy fibers carry both contextual information and joint sensory information related to desired and actual positions and velocities. An MF is modeled as a

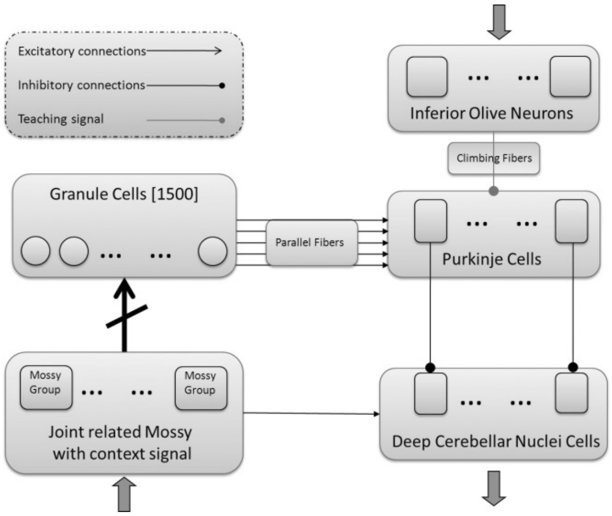


Fig. 6. Cerebellum model scheme. In FD and RR configurations, the cerebellar input, which encodes both the desired and actual position and velocity of each joint during the trajectory, is conveyed (upward arrow) through the mossy fibers (MFs) to the granular layer (crossed arrow indicates a random connectivity, i.e. each Granular cell receiving four randomly chosen MFs). Inputs encoding the error are sent (upper downward arrow) through the inferior olive (IO). Cells of the deep cerebellar nuclei (DCN) collect activity from the MFs (excitatory inputs) and the Purkinje cells (inhibitory inputs) and provide the cerebellar outputs (lower downward arrow). The DCN output is added as a corrective activity in the control loop. In the forward-architecture the output is added as corrective torques to the control torques (Fig. 4(a)). In the recurrent-architecture cerebellum configuration, the output is added as trajectory corrections (in position and velocity) in the control loop (Fig. 4(b)). Both outputs work complementarily in Fig. 4(c).

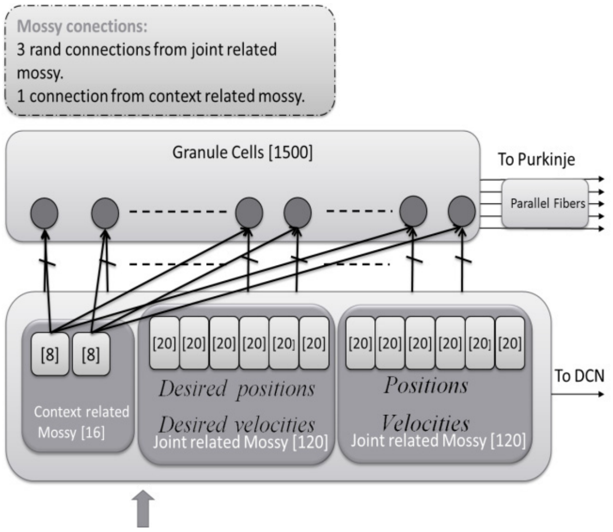

Mossy conections:
3 rand connections from joint related mossy.
1 connection from context related mossy.
Granule Cells [1500]
To Purkinje
Parallel Fibers
[8]
[8]
Context related Mossy [16]
[20]
Desired positions
Desired velocities
Joint related Mossy [120]
Positions
Velocities
Joint related Mossy [120]
To DCN

Fig. 7. Granular layer model. Explicit and Implicit context encoding approach.[59] Each granule cell receives excitation from an explicit-context-encoding fiber and three other randomly chosen MFs from the current and desired position and velocity groups.

leaky I&F neuron, whose input current is calculated using overlapping Gaussian functions as receptive fields on the input-variable value space.[39] This is carried out by modeling the contribution received from muscle or skin related afferents at a high level of abstraction. This cerebellar input layer (MFs) has been divided into 14 groups of fibers: 12 groups of twenty-grouped fibers encode both actual and desired joint velocity and position sensor information; the other 2 groups encode the context. The explicit contextual information is encoded by these 2 groups of eight-grouped cells (16 context input fibers). These MFs encode information assumed to be received through other sensory systems (such as vision). Each different context (object under manipulation) activates differently this population of neurons. Figure 7 illustrates this input connectivity.

(ii) *Granular layer* (1500 *cells*) (*GR*): A simplified granular layer of the cerebellum has been designed with the purpose of obtaining suitable signals at parallel fiber (PF) signals. The information provided by MFs is transformed into a sparse representation that facilitates discrimination of very similar inputs in the large granule cell (GR) layer,[56] in which each cell receives four excitatory connections: three connections from randomly chosen joint-related MFs groups and the other one, from a context-related MF group. PFs are the output of this layer. The sensorimotor corrective models are learned and stored as weight values at the PF-PC connections.

(iii) *Climbing fibers* (*CF*) (48 *climbing fibers in forward architecture,* 96 *climbing fibers in recurrent architecture*): This layer consists of 6 groups of CFs. In recurrent architecture, each group is composed by 16 CFs (each of them is subdivided in 2 subgroups of 8 CFs). In the forward architecture, each group is composed by 8 CFs. Each CF carries the teaching spikes (obtained from error signals) from the IO to a Purkinje cell.

(iv) *Purkinje Cells* (*PC*) (48 *Purkinje cells in forward architecture*) (96 *Purkinje cells in recurrent architecture*): In the forward architecture, this layer is divided into 6 groups of 8 cells. In the recurrent architecture, this layer is composed by 6 groups of 16 PCs; each group is also subdivided into 2 subgroups of 8 Purkinje cells. Each granule layer cell is connected to 80 per cent of PCs. Each PC receives the teaching signals used for the synaptic plasticity from a CF. Every two PCs; a cell of the deep-cerebellar-nuclei is inhibited. The PF-PC synaptic conductances are set to an initial value (15nS) at the beginning of the simulation, and are modified by the learning mechanism during the training process. This value is significantly lower in the corresponding rat cerebellum synapses.[57] However, a reduced version of the cerebellum (1500 GRs) is being modeled; therefore, each PC only receives activity from 1200 PFs (80% of GR). In a full model of the cerebellum, each PC should receive activity from 150000 PFs.[58] Therefore, PF-PC weight values have been scaled in order to obtain a similar PC excitation.

(v) *Deep Cerebellar Nucleus cells* (*DCN*) (24 *DCN cells in forward architecture*) (48 *DCN cells in recurrent architecture*): In the forward model, the cerebellum output is generated by 6 groups of these cells (2 groups per joint) whose activity provides corrective actions to the specified arm commands. Each neuron group in the DCN receives excitation from every MF and inhibition from the two corresponding PCs. In this way, the sub circuit PC-DCN-IO is organized in six microzones. In the forward architecture, the cerebellar corrective output (torque) for each joint is encoded by a couple of these groups. One group is dedicated to compensate for negative errors (agonist) and the other one is dedicated to compensate for positive errors (antagonist). In the case of the recurrent architecture, the cerebellum output is generated by 6 groups of these cells; 3 groups correspond to the joint–position corrections (one group per joint) and the other three groups correspond to the joint–velocity corrections. Each group is subdivided into 2 subgroups (of 4 cells); one subgroup handles positive error corrections and the other one handles negative error corrections.

2.6. *Learning process*

Although there seems to exist adaptation processes at several sites within the cerebellar circuitry,[60,61] one of the main synaptic adaptation mechanisms (induced by CF activity) seems to be the long-term depression (LTD) at PF-PC synapses[62,63] that has been correlated to cerebellar motor learning.[64] Therefore, the IO output (CF activity) is interpreted as an error-related signal [65–68] which drives this plasticity. When the conductivity of a PF-PC synapse becomes very low by this adaptation, the corresponding PC will not inhibit its corresponding deep cerebellar nucleus cells.[56,69] Another type of plasticity, long-term potentiation (LTP), which occurs at the same site, does not require the activation of CF[70] and compensates the effect of LTD.

Spike-timing-dependent plasticity (STDP) mechanisms to reproduce these adaptation processes have been implemented.[71] Since LTD synaptic plasticity requires the co-activation of PF and CF input, every time a CF spike is received by a PC, the conductance of all PF synapses corresponding to that PC are decreased according to Eq. (6a). That is, the past spike activity received through each PF is convolved with the integral kernel defined by Eq. (5) and the result is used to obtain the corresponding conductance decreases. This integral kernel, which correlates the IO and PF activity, was designed in such a way that it shows a peak at 100 milliseconds[72–74]; which makes the PF activity that was received 100 ms before the CF spike relevant. This time delay matches the sensorimotor delays of our system (see Fig. 4). After this mechanism is repetitively activated, when the same pattern of PF activation appears, the PC will not become active and the corresponding DCN will produce activity recognizing the learned pattern. The opposite adaptation process (LTP) is implemented by increasing the weight of a PF-PC synapse each time it transmits a spike as defined in Eq. (6b).[39,43,68,71]

$$k(t) = e^{-(\frac{t-t_0}{\tau})} \sin\left(2\pi\left(\frac{t-t_0}{\tau}\right)\right)^{20}. \tag{5}$$

$$LTD : \forall\, i, \quad \Delta w_i$$

$$= \beta \int_{-\infty}^{\mathrm{IOspike}} k(t_{\mathrm{IOspike}} - t)\delta(t)_{\mathrm{PF}_i}\, dt. \tag{6a}$$

$$LTP : \Delta w_i = \alpha. \tag{6b}$$

Where t_0 is a time constant determined by the biological path delay which is fixed to 100 ms and τ is a time reference which is set to 1 s in order to normalize the arguments in the learning rule. Δw_i represents the synaptic weight increment at the ith PF reaching that PC, t_{IOspike} stands for the time in which the corresponding CF transmits a spike, $\delta(t)_{PFi}$ is a Dirac delta function which represents the activity in the ith PF (1 when the PF carries a spike, 0 when it does not). Finally α and β are constant values that modulate the synaptic weight changes at PF-PC synapses ($\alpha = 0.005$ and $\beta = -0.1$).

2.7. *The error signal drives teaching signal*

The trajectory position/velocity error is used to calculate the teaching signal. This teaching signal follows Eq. (7).

$$\begin{aligned}\varepsilon_{\text{position}_i} &= (Q_{\text{desired}_i} - Q_{\text{actual}_i})[(t + t_{\text{delays}}) - t_i].\\ \varepsilon_{\text{velocity}_i} &= (\dot{Q}_{\text{desired}_i} - \dot{Q}_{\text{actual}_i})[(t + t_{\text{delays}}) - t_i].\\ &\qquad i = 1, 2, \ldots n, \quad \text{joints}\end{aligned} \tag{7}$$

And the computed error in the forward and recurrent architectures is given by Eq. (8).

$$\begin{aligned}FD\varepsilon_{\text{delayed}_i} &= k_{pi} \cdot \varepsilon_{\text{position}_i} + k_{vi} \cdot \varepsilon_{\text{velocity}_i};\\ RR\varepsilon_{p\text{delayed}_i} &= k_{pi} \cdot \varepsilon_{\text{position}_i};\\ RR\varepsilon_{v\text{delayed}_i} &= k_{vi} \cdot \varepsilon_{\text{velocity}_i}\\ i &= 1, 2 \ldots n, \quad \text{joints.}\end{aligned} \tag{8}$$

Where $k_{pi} \cdot \varepsilon_{\text{position}i}$ represents the product of a constant value (gain) at each joint and the position error in this joint (difference between desired joint position and actual joint position ($Q_{\text{desired}} - Q_{\text{actual}}$)).

$k_{vi} \cdot \varepsilon_{\text{velocity}i}$ represents the product between a constant value (gain) at each joint and the velocity error in this joint (difference between desired joint velocity and actual joint velocity ($\dot{Q}_{\text{desired}} - \dot{Q}_{\text{actual}}$)).

Position/velocity error signals are delayed to align them in time according to biological delay pathways (t_{delays} represents the signal delays in the control loop). Biologically speaking, this time-matching of the desired and actual joint states can be explained 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.[75]

IO cells respond with probabilistic Poisson process encoding the teaching signal into a low frequency probabilistic spike train (from 0 to 10 Hz, average 1 Hz).[76]

2.8. *Decoding the cerebellar output*

The output variables $\tau_{\text{corrective}}$ (in the FD configuration) or $(q, \dot{q})_{\text{corrective}}$ (in the RR configuration) are extracted from the firing rates of the DCN belonging to the related population, Eq. (9).

$$\tau^{+/-}_{\text{corrective}}/(q, \dot{q})^{+/-}_{\text{corrective}} = \sum_{j=1}^{4} \bar{v}_j(t). \tag{9}$$

Where $\bar{v}_j(t)$ is the firing rate of neuron j at time t, and the over-line indicates that the measures are averaged over a sliding time window of 100 ms, inspired by the low frequency filtering performed by motoneurons.

2.9. *Experimental methods*

Firstly, the behavior of different control architectures has been studied in a noisy scenario by using a Gaussian/uniform additive white noise on MF input signals. Table 2 indicates the different tested levels of noise.

The MFs signals are driven when an animal performs different activities. When an arm is moved along a learned trajectory, this arm movement is accompanied by predictable changes occurring primarily in MFs inputs reporting kinesthetics of this movement. Noise on the produced neural control signal (which may vary the firing time of motor neurons) will cause deviation in actual trajectories from the desired ones: $Q(t) = Q_{\text{desired}}(t) + \varepsilon(t)$ where Q_{desired} represents the desired trajectory/velocities to be followed. We have studied how the system behaves against two noise models (Table 2): (a) ε is a random signal with uniform distribution and a non-repeatable seed, (b) ε is a random signal with Gaussian distribution and zero mean. Although Golgi cells

Table 2. Noise levels on mossy fiber signals.

$SNR = 10\log\frac{\mathrm{E}[x^2(n)]}{\varepsilon^2(n)}$	Uniform distribution	Gaussian distribution
Noise 1x	32 dB	23 dB
Noise 2x	18 dB	15.5 dB
Noise 4x	4 dB	8 dB

seem to play a crucial role in removing noise,[77] the evaluated cerebellar circuitry may help to accomplish this task.

Different added noise levels were checked. The used Signal-to-Noise-Ratio can be seen in Table 2.

In addition, different experiments have been also carried out to evaluate how different biologically-inspired control architectures work by abstracting models and switching between different contexts with a suitable cerebellar configuration.

3. Results

3.1. *Noise on MF input*

The learning process is evaluated by using the Mean Absolute Error (MAE) curve. For the calculation of the MAE of a trajectory execution, the addition of the error in radians produced by each joint independently along the whole trajectory has been used. 800 trials of the defined 8-shaped trajectory using the FD, RR, and FD&RR control-loop architectures have been executed. The robot end effector was loaded with 2 kg to increase the inertia (thus, the initial dynamics model needs to be corrected through learning).

Figure 8 shows the global mean absolute error (MAE) evolution obtained from the robot joint coordinates in radians. The noise was set to 1x, 2x, and 4x and was generated from a uniform distribution (a, b, and c plots, respectively) and a Gaussian distribution (d, e, and f plots).

As it is shown in Fig. 8, FD&RR architecture remains more stable against random noise than FD and RR architectures used independently. When the random noise level is 1x, FD and FD&RR responses are similar, but the more noise there is, the better response is obtained in FD&RR compared to RR and FD. In Figs. 8(b) and 8(c), it is easy to see that the convergence speed and output stability is better in FD&RR. FD&RR uses both configurations in a complementary way, to support FD with the corrections provided by RR. This causes FD&RR to have more stability and better performance when the noise is higher.

White Gaussian noise (Figs. 8(a)–8(c)) allows a good precision in the cerebellum output correction (torque predictions); prediction errors remain highly delimited around mean values. The probability density function has its maximum value at the mean, that is, during the learning process, the generated noise values tend to accumulate around the mean, the cerebellum learns this tendency and compensates it. In contrast, white uniform noise (Figs. 8(d)–8(f)) makes prediction torques less accurate since its probability density function does not have a single maximum value (thus, this is a harder task). The generated noise values do not tend to accumulate around any specific value therefore; the cerebellum cannot easily abstract any tendency information.

3.2. *Context switching between two dynamic/kinematic models*

Firstly, a set of experiments have been executed to study the capability of the cerebellar model to infer different corrective models when the dynamics of the robotic arm is modified by manipulating different objects using FD, RR, and FD&RR architectures. During a first learning process, the robot was loaded with a 1kg weight and executed 450 trials of the 8-like trajectory (Fig. 9(a)). During a second learning process, the robot was loaded with a 2 kg weight (Fig. 9(b)).

As shown in Figs. 9(a) and 9(b), FD&RR architecture takes advantage of both configurations; it uses the cerebellar corrections in torques and in positions and velocities to provide a better profile in the obtained MAE curves.

In order to evaluate the ability of the cerebellar module to infer and store different corrective models simultaneously using different control-loop architectures, an experiment in which the dynamics of the robotic arm is changed during the learning process every 15 trials has been carried out. The context alternates between manipulating a 2 kg object and a 1 kg object (Fig. 9(c)). The context-related cerebellar input is supplied with different signals in each context to enable the cerebellum to differentiate both contexts allowing different models (contexts) to be efficiently learned and retrieved in a non-destructive manner. Finally, the three different proposed control architectures (RR, FD, and FD&RR) are compared in a kinematic context switching scenario (Fig. 9(d)). This kinematic context switching scenario consists of a deformation of the end-effector (angle).

In RR architecture, the relationship between the produced robot-arm state error and the cerebellar output is direct, the cerebellum receives the position and velocity error-related signals, which are properly

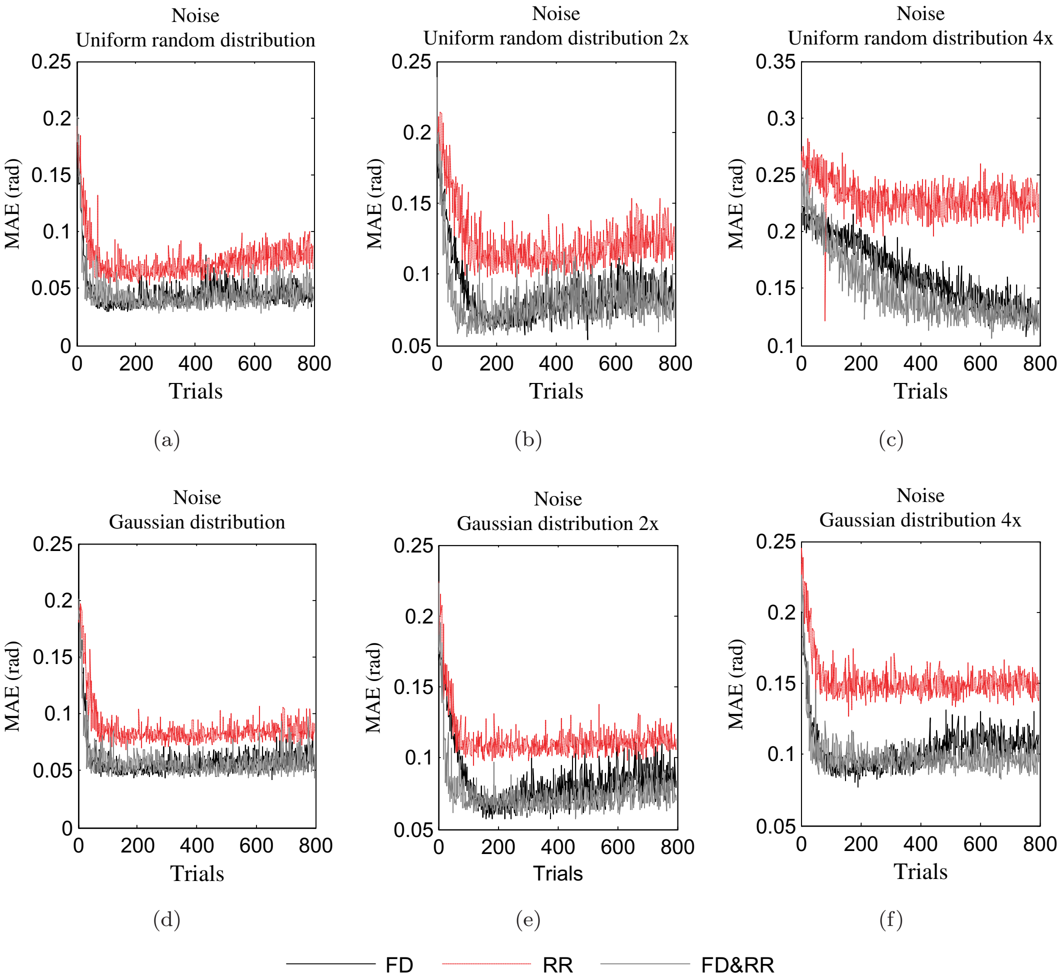
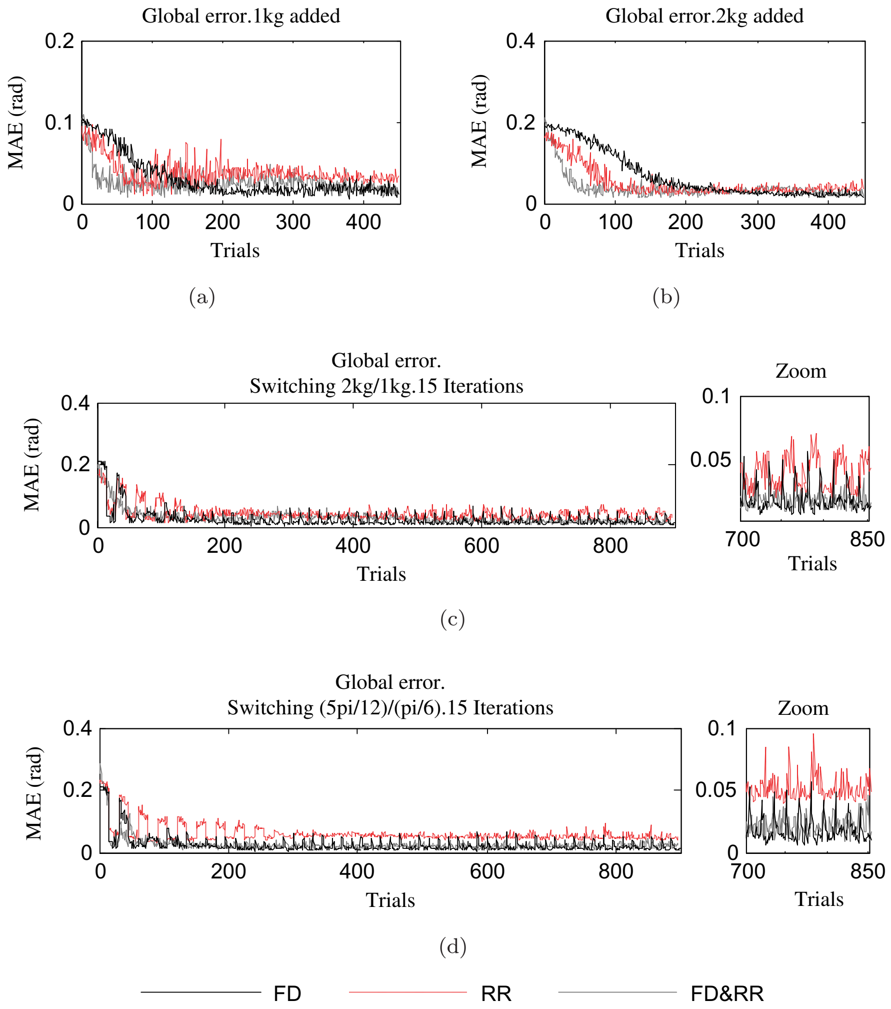


Fig. 8. Accuracy evolution of the three control architectures (FD, RR, and FD&RR) when introducing noise on input signals of MFs. Mean absolute error (MAE) of the joint coordinates in radians of the robot loaded with a 2 kg weight using forward, recurrent, and FD&RR architectures during a learning process of 800 trials of the 8-like trajectory execution. (a), (b), and (c) correspond to 1x, 2x, and 4x additive noise respectively using a uniform distribution. (d), (e), and (f) correspond to 1x, 2x, and 4x additive noise respectively using a Gaussian distribution.

provided by CFs, and cerebellar outputs consist of trajectory corrections of position and velocity, being these input and output dimensions equivalent. Therefore, the cerebellum does not need to implement a complex model representation translation. With this dimension matching, the cerebellum is able to learn and provide a quick response (a faster convergence is obtained in RR than in FD in Figs. 9(a) and 9(b)). Nevertheless, our crude inverse dynamic robot models need to be fed with clean and continuous corrected inputs in order to supply accurate torque values which command the robotic arm properly. Due to these required input characteristics of this dynamic model, the final torque commands in RR architecture are not as good as the ones delivered by the FD architecture (the RR MAE error curve is less stable than the FD case). The accuracy of the cerebellum corrective output involves making a trade-off between number of cells and simulation time.

In the FD architecture, the accuracy is not improved by correcting the input of the robot crude inverse dynamics; the cerebellum supplies torque command corrections almost directly to the robotic arm (see Fig. 4(a)). However, in the FD architecture, the relationship between the produced robot-arm state error and the cerebellar output is not straightforward. The position and velocity error signal is



(a)

(b)

(c)

(d)

Fig. 9. Accuracy evolution of the three control architectures with and without context switching. Mean absolute error (MAE) of the robot joint coordinates in radians using FD, RR, and FD&RR architectures executing the 8-like trajectory. (a) and (b) The context is not changed: The robot manipulates a 1 kg object and a 2 kg object respectively during a 450-trial learning process. (c) The dynamics of the robotic arm is alternately changed between the two contexts every 15 trials. In the first context, the end segment of the robot arm is loaded with a 2 kg object. In the second one, it is loaded with a 1kg object. (d) The kinematics of the plant is alternately changed between two contexts every 15 trials: in the first context, the robot must follow the trajectory using an end segment which is deformed $5\pi/12$ radians. In the second one, the robot end segment is deformed $\pi/6$ radians (this corresponds to kinematics changes that may be caused by manipulating an object of a certain length).

conveyed by CFs while cerebellar output supplies torque corrections. These input and output dimensions are not equivalent, the cerebellum learning task is of higher complexity. Thus, the learning convergence is slower but the command torques are more precise (FD MAE error curve is more stable than the one of the RR architecture).

Again, FD&RR combines the advantages of both the RR and the FD architectures. It has a high convergence speed and good output stability after learning. FD&RR uses the position and velocity corrections given by RR to facilitate the FD torque correction task, and they mutually complete each other.

Figure 9(c) shows the MAE evolution of a 900-trial learning process. It is shown that the learning is performed in a non-destructive manner since once the final error for each context is reached, this error value is maintained stable when the context changes (therefore, the previously-learned context model was not destroyed). This feature relies on the separation capability of the granular layer for sensory signals related to different contexts. Again, FD&RR reaches a better behavior compared to RR and FD architectures (Fig. 9(c)). FD&RR allows better inter-context transitions (error peaks between two different contexts are almost negligible because of its superior convergence speed (thanks to the RR loop)) and it also achieves a better final stability.

The MAE evolution of a 900-trial learning process is shown in Fig. 9(d). The results are similar to the dynamics context-switching scenario, in single-context learning (as we have within 15 iterations), high convergence speed, and good performance learning curve profile in the long term seem to be desirable aims. RR makes the transition between contexts softer; in the long term, no transition peaks are observed. When the kinematics of the robotic plant changes, no inertia tensors are involved, so the "crude inverse dynamic robot model" module has an easier task in computing the proper new torques. On the other hand, FD provides a better curve performance than RR (no crude inverse dynamic robot model is involved in processing the cerebellum output). As shown in Fig. 9(d), FD&RR configuration takes advantage of both loops obtaining smoother transitions between contexts and a good learning curve profile in long the term.

4. Conclusions

This work has focused on studying biologically-inspired robot-arm control architectures under dynamic and kinematic perturbations of the manipulation scenario. Furthermore, it has evaluated different control loops (RR, FD, and FD&RR) in several noisy scenarios. A cerebellar adaptive module embedded in these loops could effectively provide torque/position&velocity corrections to compensate for deviations in the dynamics/kinematics of a base robotic arm model (due to the manipulation of different objects and deformations of the end effector) increasing the movement accuracy.

The cerebellar model included an input representation which encodes context-specific inputs and current sensory signals encoding the actual arm states during the experiment.

It has been evaluated how a temporal-correlation kernel driving an error-related LTD and a compensatory LTP component (which complement each other) can achieve an effective adaptation of the corrective cerebellar output.

The obtained results indicate that coupling both control loop architectures (FD&RR) leads to a high robustness against noise. Employing the recurrent architecture (RR) to ensure a faster convergence in learned profile curve dynamics has been combined with exploiting the fact that the forward architecture (FD) provides a better accuracy gain and output stability in a noisy scenario.

In the same way, the results demonstrate that the composite control architecture in context switching has the capability to infer and store different corrective models simultaneously under dynamic/kinematic modifications better than FD or RR configurations on their own.

The assumption that the cerebellum is involved in forward modeling for motor control is familiar in the literature.[78,79] Our results suggest that both FD and RR loops could be present in the biological motor control in order to achieve a better performance. In fact, this proposed architecture (FD&RR) is compatible with several neurophysiological findings. Firstly, several studies have reported relations between motor cortex activity and various kinematic parameters of the motor output such as distance and speed[80–82] as well as parameters related to the dynamics of the movement.[22] As the motor cortex has been described as one of the targets of the cerebellar output,[83] the cerebellar output could influence these kinematic (RR loop) and dynamic parameters (FD loop). And secondly, results of virus tracing studies have shown that the regions of the cerebellar cortex that receive input from the motor cortex are the same as those that project to the motor cortex.[84] These observations suggest that several closed-loop circuits may be present in the cerebrocerebellar circuits as it occurs in the FD&RR architecture.

As future work, the scalability of these cerebellar configurations, the potential role of new nervous circuits, such as the cuneate nucleus and Golgi cells in noisy scenarios, other kinds of plasticity, and cell features and finally, scalability on the number of robot-plant joints will be studied.

Acknowledgments

This work has been supported by the EU grants SENSOPAC (IST 028056) and REALNET (IST-270434), the Spanish national project ARC-VISION (TEC2010-15396), and Spanish *Subprograma Juan de la Cierva 2009* (MICINN).

References

1. S. J. Blakemore, D. M. Wolpert and C. D. Frith, Central cancellation of self-produced tickle sensation, *Nat. Neurosci.* **1**(7) (1998) 635–640.
2. R. B. Ivry and S. W. Keele, Timing functions of the cerebellum, *J. Cog. Neurosci.* **1** (1989) 136–152.
3. M. Kawato and H. Gomi, A computational model of four regions of the cerebellum based on feedback-error learning, *Biol. Cybern.* **68**(2) (1992) 95–103.
4. R. C. Miall, D. J. Weir, D. M. Wolpert and J. F. Stein, Is the cerebellum a Smith predictor? *J. Mot. Behav.* **25** (1993) 203–216.
5. A. M. Smith, Does the cerebellum learn strategies for the optimal time-varying control of joint stiffness? *Behav. Brain Sci.* **19**(3) (1996) 339–527.
6. J. S. Albus, A theory of cerebellar function, *Math Biosci.* **10** (1971) 25–61.
7. D. Marr, A theory of cerebellar cortex, *J. Physiol.* **202** (1969) 437–470.
8. J. Porrill, P. Dean and J. V. Stone, Recurrent cerebellar architecture solves the motor-error problem, *Proceedings in Biol. Sci.* **271**(1541) (2004) 789–796.
9. A. Haith and S. Vijayakumar, Robustness of VOR and OKR adaptation under kinematics and dynamics transformations, in: *Proceedings of 6th IEEE (ICDL '07)*, London (2007).
10. M. Ito, Cerebellar circuitry as a neuronal machine, *Prog. Neurobiol.* **78** (2006) 272–303.
11. C. Assad, S. Dastoor, S. Trujillo and L. Xu, Cerebellar dynamic state estimation for a biomorphic robot arm, *Syst. Man Cybern. IEEE Trans.* (2005) 877–882.
12. E. O'Hearn and M. E. Molliver, Organizational principles and microcircuitry of the cerebellum, *International Review of Psychiatry* **13**(4) (2001) 232–246.
13. J. Porrill and P. Dean, Cerebellar motor learning: when is cortical plasticity not enough? *PLoS Computational Biology* **3** (2007a) 1935–1950.
14. J. Porrill and P. Dean, Recurrent cerebellar loops simplify adaptive control of redundant and nonlinear motor systems, *Neural Computation* **19**(1) (2007b) 170–193.
15. N. Ramnani and C. Miall, Expanding cerebellar horizons, *Trends Cog. Neurosci.* **5** (2001) 135–136.
16. F. Colin, L. Ris and E. Godeaux, in *The Cerebellum and its Disorders*, ed. M. U. Manto (Cambridge University Press, Cambridge, 2002).
17. S. J. Blakemore, C. D. Frith and D. M. Wolpert, The cerebellum is involved in predicting the sensory consequences of action, *NeuroReport* **12**(9) (2001) 1879–1885.
18. M. Ito, The cerebellum and neural control, *New York, Raven Press* (1984).
19. M. Kawato and D. M. Wolpert, Internal models for motor control, *Novartis Foundation Symp.* **218** (1998) 291–307.
20. R. C. Miall, The cerebellum, predictive control and motor coordination, in *Novartis Foundation Symposium. Sensory Guidance of Movements* **218** (2007) 272–290.
21. P. Dean, J. Porrill, C. F. Ekerot and H. Jörntel, The cerebellar microcircuit as an adaptive filter: Experimental and computational evidence, *Nat. Rev. Neurosci.* **11** (2010) 30–34.
22. L. E. Sergio, C. Hamel-Pâquet and J. F. Kalaska, Motor cortex neural correlates of output kinematics and kinetics during isometric-force and arm-reaching tasks, *J. Neurophysiol.* **94** (2005) 2353–2378.
23. W. S. Levine, *The Control Handbook* (CRC Press, Boca Raton, 1996), pp. 1339–1368.
24. C. Ott *et al.*, A humanoid two-arm system for dexterous manipulation, *IEEE-RAS International Conference on Humanoid Robots* (2006) 276–283.
25. 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, *European J. Neuroscience* **10** (1998a) 95–105.
26. N. Schweighofer, M. A. Arbib and M. Kawato, Role of the cerebellum in reaching movements in human. I. Distributed Inverse dynamics control, *European J. Neuroscience* **10** (1998b) 86–94.
27. J. L. Rossello, V. Canals, A. Morro and J. Verd, Chaos-based mixed signal implementation of spiking neurons, *Int. J. Neural Syst.* **6**(16) (2009) 465–471.
28. S. P. Johnston, G. Prasad, L. Maguire and T. M. McGinnity, An FPGA Hardware/Software Co-Design Towards Evolvable Spiking Neural Networks for Robotics Application, *Int. J. Neural Syst.* **20**(6) (2010) 447–461.
29. E. Nichols, L. J. McDaid and N. H. Siddique, Case study on a self-organizing spiking neural network for a robot navigation, *Int. J. Neural Syst.* **20**(6) (2010) 501–508.
30. S. Ghosh-Dastidar and H. Adeli, Improved spiking neural networks for EEG classification and epilepsy and seizure detection, *Integr. Comput-Aid E.* **14**(3) (2007) 187–212.
31. S. Ghosh-Dastidar and H. Adeli, A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection, *Neural Networks* **22**(10) (2009) 1419–1431.
32. J. Iglesias and A. E. P. Villa, Emergence of preferred firing sequences in large spiking neural networks

during simulated neuronal development, *Int. J. Neural Syst.* **18**(4) (2008) 267–277.

33. S. Soltic and N. Kasabov, Knowledge extraction from evolving spiking neural networks with rank order population coding, *Int. J. Neural Syst.* **20**(6) (2010) 437–445.
34. T. J. Strain, L. J. McDaid, L. P. Maguire and T. McGinnity, An STDP training algorithm for a spiking neural network with dynamic threshold neurons, *Int. J. Neural Syst.* **20**(6) (2010) 463–480.
35. S. Schliebs, N. Kasabv and M. Defoin-Platel, On the probabilistic optimization of spiking neural networks, *Int. J. Neural Syst.* **20**(6) (2010) 481–500.
36. R. Acharya, E. C. P. Chua, K. C. Chua, L. C. Min and T. Tamura, Analysis and automatic identification of sleep stages using higher order spectra, *Int. J. Neural Syst.* **20**(6) (2010) 509–521.
37. S. Ghosh-Dastidar and H. Adeli, Spiking neural networks, *Int. J. Neural Syst.* **4**(19) (2009).
38. E. Ros, R. 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 Comp.* **18** (2006) 2959–2993.
39. R. R. Carrillo, E. Ros, C. Boucheny and O. J.-M. D. Coenen, A real-time spiking cerebellum model for learning robot control, *Biosystems* **94**(1–2) (2008) 18–27.
40. J. Butterfaß, M. Grebenstein, H. Liu and G. Hirzinger, DLR Hand II: Next generation of a dextrous robot hand, *IEEE ICRA* (2001) 109–114.
41. L. E. Miller, R. N. Holdefer and J. C. Houk, The role of the cerebellum in modulating voluntary limb movement commands, *Archives Italiennes de Biologie* **140** (2002) 175–183.
42. H. Hoffmann, G. Petckos, S. Bitzer and S. Vijayakumar, *Sensor-Assisted Adaptive Motor Control Under Continuously Varying Context*, presented at ICINCO, Automation and Robotics (2007).
43. W. Gerstner and W. Kistler (eds.), *Spiking Neuron Models: Singleneurons, Populations, Plasticity* (Cambridge University, 2002).
44. F. J. Pelayo, E. Ros, X. Arreguit and A. Prieto, VLSI implementation of a neural model using spikes, *Analog Integr. Circ. S.* **13**(1–2) (1997) 111–121.
45. E. Ros *et al.*, Stimulus correlation and adaptive motion detection using spiking neurons, *Int. J. Neural Syst.* **9**(5) (1999) 485–490.
46. J. Feng, Is the integrate-and-fire model good enough? A review, *Neural Networks* **14** (2001) 955–975.
47. Z. Nusser, S. CullCandy and M. Farrant, Differences in synaptic GABA(A) receptor number underlie variation in GABA mini amplitude, *Neuron* **19** (1997) 697–709.
48. D. J. Rossi and M. Hamann, Spillover-mediated transmission at inhibitory synapses promoted by high affinity alpha(6) subunit GABA(A) receptors and glomerular geometry, *Neuron* **20** (1998) 783–795.
49. R. A. Silver, D. Colquhoun, S. G. CullCandy and B. Edmonds, Deactivation and desensitization of non-NMDA receptors in patches and the time course of EPSCs in rat cerebellar granule cells, *J. Physiol.* **493** (1996) 167–173.
50. S. Tia, J. F. Wang, N. Kotchabhakdi and S. Vicini, Developmental changes of inhibitory synaptic currents in cerebellar granule neurons: Role of GABA(A) receptor alpha 6 subunit, *Journal of Neuroscience* **16** (1996) 3630–3640.
51. E. D'Angelo, G. Defilippi, P. Rossi and V. Taglietti, Synaptic activation of Ca2$^+$ action potentials in immature rat cerebellar granule cells *in situ*, *J. Neurophysiol.* **78**(3) (1997) 1631–1642.
52. E. D'Angelo, T. Nieus, A. Maffei, S. Armano and P. Rossi, Theta-frequency bursting and resonance in cerebellar granule cells: experimental evidence and modeling of a slow K$^+$-dependent mechanism, *J. Neurosci.* **21** (2001) 759–770.
53. E. D'Angelo, G. Defilippi, P. Rossi and V. Taglietti, Ionic mechanism of electroresponsiveness in cerebellar granule cells implicates the action of a persistent sodium current, **80** (1998) 493–503.
54. E. D'Angelo, P. Rossi and V. Taglietti, Different proportions of N-Methyl-D-Aspartate and Non-N-Methyl-D-Aspartate receptor currents at the mossy fiber granule cell synapse of developing rat cerebellum, *Neurosci.* **53** (1993) 121–130.
55. T. Nieus, E. Sola, J. Mapelli, E. Saftenku and P. Rossi, LTP regulates burst initiation and frequency at mossy fiber-granule cell synapses of rat cerebellum: Experimental observations and theoretical predictions, *J. Neurophysiol.* **95** (2006) 686–699.
56. 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, *LNCS* **3512** (2005) 107–113.
57. P. Isope and B. Barbour, Properties of unitary granule cell− >Purkinje cell synapses in adult rat cerebellar slices, *J. Neurosci.* **22**(22) (2002) 9668–9678.
58. N. Brunel, V. Hakim, P. Isope, J. P. Nadal and B. Barbour, Optimal information storage and the distribution of synaptic weights: Perceptron versus Purkinje cell, *Neuron* **43** (2004) 745–757.
59. N. R. Luque, J. A. Garrido, R. R. Carrillo, O. J.-M. D. Coenen and E. Ros, Cerebellar input configuration towards object model abstraction in manipulation tasks, *Neural Network, IEEE Trans.* DOI: 10.1109/TNN.2011.2156809 (2011).

60. T. V. Bliss and T. Lomo, Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path, *J. Physiol.* **232** (1973) 331–356.

61. 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.* **4** (2001) 467–475.

62. 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. Letter* **33** (1982) 253–258.

63. G. Bi and M. Poo, Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type, *J. Neurosci.* **18** (1998) 10464–10472.

64. C. I. De Zeeuw and C. H. Yeo, Time and tide in cerebellar memory formation, *Curr. Opin. Nuerobiol.* **15** (2005) 667–674.

65. 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. Comp. Neurosci.* **14** (2003) 311–327.

66. A. Roth and M. Häusser, Compartmental models of rat cerebellar Purkinje cells based on simultaneous somatic and dendritic patch-clamp recordings, *J. Physiol* **535** (2001) 445–472.

67. S. Solinas, R. Maex and E. De Schutter, Synchronization of Purkinje cell pairs along the parallel fibrefiber axis: A model, *Neurocomputing* **52–54** (2003) 97–102.

68. R. R. Carrillo, E. Ros, B. Barbour, C. Boucheny and O. J.-M. D. Coenen, Event-driven simulation of neural population synchronization facilitated by electrical coupling, *Biosystems* **87**(2–3) (2007) 275–280.

69. 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, *Journal of Neuroscience* **17** (1997) 91–106.

70. P. A. Salin, R. C. Malenka and R. A. Nicoll, Cyclic AMP mediates a presynaptic form of LTP at cerebellar parallel fiber synapses, *Neuron* **16** (1996) 797–803.

71. N. R. Luque, J. A. Garrido, R. R. Carrillo, O. J.-M. D. Coenen and E. Ros, Cerebellar-like corrective model inference engine for manipulation tasks, *Syst Man Cybern., IEEE Transactions*, DOI: 10.1109/TSMCB.2011.2138693 (2011).

72. R. E. Kettner *et al.*, Prediction of complex two-dimensional trajectories by a cerebellar model of smooth pursuit eye movement, *J. Neurophysiol.* **77**(4) (1997) 2115–2130.

73. J. Spoelstra, M. A. Arbib and N. Schweighofer, Cerebellar adaptive control of a biomimetic manipulator, *Neurocomputing* **26–27** (1999) 881–889.

74. J. L. Raymond and S. G. Lisberger, Neural learning rules for the vestibulo-ocular reflex, *J. Neurosci.* **18**(21) (1998) 9112–9129.

75. J. L. Contreras-Vidal, S. Grossberg and D. A. Bullock, A neural model of cerebellar learning for arm movement control. Cortico-spino-cerebellar dynamics, *Learn. Memory* **3**(6) (1997) 475–502.

76. C. Boucheny, R. R. Carrillo, E. Ros and O. J.-M. D. Coenen, Real-time spiking neural network: An adaptive cerebellar model, *LNCS* **3512** (2005) 136–144.

77. D. Philipona and O. J.-M. D. Coenen, Model of granular layer encoding in the cerebellum, *Neurocomputing* **58–60** (2004) 575–580.

78. D. M. Wolpert and R. C. Miall, Forward models for physiological motor control, *Neural Netw.* **9** (1996) 1265–1279.

79. M. Kawato *et al.*, Internal forward models in the cerebellum: FMRI study on grip force and load force coupling, *Prog. Brain Res.* **142** (2003) 171–188.

80. J. Ashe and A. P. Georgopoulos, Movement parameters and neural activity in motor cortex and area 5, *Cereb. Cortex* **4** (1994) 590–600.

81. G. A. Reina, D. W. Moran and A. B. Schwartz, On the relationship between joint angular velocity and motor cortical discharge during reaching, *J. Neurophysiol.* **85** (2001) 2576–2589.

82. Q. G. Fu, D. Flament, J. D. Coltz and T. J. Ebner, Temporal encoding of movement kinematics in the discharge of primate primary motor and premotor neurons, *J. Neurophysiol.* **73** (1995) 836–854.

83. P. J. Orioli and P. L. Strick, Cerebellar connections with the motor cortex and the arcuate premotor area: An analysis employing retrograde transneuronal transport of WGA-HRP, *J. Comp. Neurol.* **288**(4) (1989) 612–626.

84. R. M. Kelly and P. L. Strick, Cerebellar loops with motor cortex and prefrontal cortex of a nonhuman primate, *J. Neurosci.* **23**(23) (2003) 8432–8444.