DTU Library

A Cerebellum-Inspired Learning Approach for Adaptive and Anticipatory Control

Tolu, Silvia; Capolei, Marie Claire; Vannucci, Lorenzo; Laschi, Cecilia; Falotico, Egidio; Hernandez, Mauricio Vanegas

Link back to DTU Orbit

World Scientific
www.worldscientific.com

A Cerebellum-Inspired Learning Approach for Adaptive and Anticipatory Control

Silvia Tolu* and Marie Claire Capolei†

Automation and Control Group
Department of Electrical Engineering, Technical University of Denmark
Richard Petersens Plads, Building 326, Kgs. Lyngby, 2800, Denmark
*stolu@elektro.dtu.dk
†macca@elektro.dtu.dk

Lorenzo Vannucci‡, Cecilia Laschi§ and Egidio Falotico¶

The BioRobotics Institute, Scuola Superiore SantAnna
Viale Rinaldo Piaggio 34, Pontedera, 56025, Pisa, Italy
‡lorenzo.vannucci@santannapisa.it
§cecilia.laschi@santannapisa.it
¶egidio.falotico@santannapisa.it

Mauricio Vanegas Hernández‖

Rokoko Electronics ApS Sankt Gertruds Straede 6E
Copenhagen, 1129, Denmark
‖mauricio@rokoko.com

The cerebellum, which is responsible for motor control and learning, has been suggested to act as a Smith predictor for compensation of time-delays by means of internal forward models. However, insights about how forward model predictions are integrated in the Smith predictor have not yet been unveiled. To fill this gap, a novel bio-inspired modular control architecture that merges a recurrent cerebellar-like loop for adaptive control and a Smith predictor controller is proposed. The goal is to provide accurate anticipatory corrections to the generation of the motor commands in spite of sensory delays and to validate the robustness of the proposed control method to input and physical dynamic changes. The outcome of the proposed architecture with other two control schemes that do not include the Smith control strategy or the cerebellar-like corrections are compared. The results obtained on four sets of experiments confirm that the cerebellum-like circuit provides more effective corrections when only the Smith strategy is adopted and that minor tuning in the parameters, fast adaptation and reproducible configuration are enabled.

Keywords: Internal forward model; cerebellum; motor control; adaptive learning; Smith predictor; bio-inspired; recurrent.

---

*Corresponding author.

1. Introduction

Human beings show highly skilled motor responses in spite of their complex musculoskeletal dynamics, transport delays and response latencies. A reactive feedback system, originating in the brain-stem and spinal cord, and a feed-forward anticipatory system, originating in the cerebellum, allow fine motor control of the body.[1]

The cerebellum has a complex neural organization that gives rise to a massive signal-processing capability for accomplishing several types of motor learning.[2] The cerebellum's role in motor function, i.e. its contribution to the execution of coordinated, and precise complex actions, is well recognized.[3] It has been widely accepted that the cerebellum stores internal models to represent input–output properties of a body part.[4–6] Two types of internal models have been proposed: forward and inverse models.[4,7] This paper focuses on forward models. In his seminal paper on cerebellum and internal models, Ebner[5] affirmed: "Forward models use the commands for an action and information about the present state to predict the consequences of that action." More recent evidence[8] hypothesized that the cerebellum may use the motor command or "efference copy" signal sent to an effector to overcome the problems associated with long delays in sensory feedback.

Considering the structure of the control system for voluntary movements, Ito[9] depicted a plausible control path of the connections between the cerebral cortex and the cerebellum that has anatomical and physiological bases: the forward model-based control loop. Miall[10] suggested that the cerebellum could act as a Smith predictor[11] for compensation of time delays that occur in the sensorimotor transmission and neural central processing by means of forward models. As a matter of fact, the engineering control strategy known as Smith predictor, which is a type of predictive controller for systems with pure time delay, is based on the forward model of the robot plant; it provides a fast prediction of the outcome of a motor command and a delayed copy of that prediction, which will match in time the actual feedback arising from the movement. However, no study has confirmed Miall's hypothesis; no control loop merging a computational model of the cerebellum and an internal forward model acting as a Smith predictor has ever been presented. Twenty years later, Porrill

et al.[12] argued Miall's hypothesis appeared to be in principle biologically implausible because of the constraint of having a suitable teaching signal. Nonetheless, they also affirmed: "However, even for plausible schemes, such as forward models for noise cancellation and novelty-detection, and the recurrent architecture for adaptive inverse control, there is unlikely to be a simple mapping between microzone function and internal model structure".[12] In contrast with the previous statements, we focused our investigation on conceiving a plausible mapping by integrating the Smith predictor strategy into the biological recurrent control loop[13] while preserving the intrinsic and suitable teaching signal.

In this study, a methodology for integrating the Smith predictor mechanism with a cerebellar-like network has been devised. The former includes the forward model-based learning for providing sensory predictions to the second one that automatically supplies corrective control signals making the control adaptive. Any biological system has an inherent delay from when a command has been issued to a muscle to when the response of the limb has been received. In this work, the delays are the ones that arise from the communication system functioning and the motors feedback. The main objective of this work is to gain insights into the sensory signal processing in motor control by answering the following questions: (i) In the proposed architecture, what is the role and connectivity of the forward cerebellar model? (ii) How are the sensory signals distributed in order to drive the learning and corrections in combination of the Smith predictor and the cerebellar-like network?

The forward model and the cerebellar-like microcircuit were reproduced by combining machine learning and computational neuroscience techniques. The control scheme was then tested on a real physical platform, the Fable modular robot,[14] combining basic modules that led to a mutual influence in the joint dynamics. In this context, a real modular robot represents a suitable benchmark platform to investigate the role of the cerebellum in anticipatory control and the relation with its internal modularity. We propose a cerebellar-based modular controller that is capable of effectively reducing the errors in few iterations of the same task for both a single and double robot module configuration.

1.1. *Background*

Since the late 1960s, several authors, such as Marr (1969),[15] Albus (1971),[16] Fujita (1982),[17] shared their view of the cerebellum as an adaptive controller. The use of cerebellar models for robotic control has become an active research for developing accurate low-gain control schemes for robotic platforms of several degrees of freedom and compliant joints (see review on cerebellar control[18]). For example, these models have been used in robotic manipulation tasks, where the cerebellum abstracts dynamic models of the robotic platform and provides adaptive feed-forward corrections to the imprecise control signal obtained from an inverse dynamics controller[19,20] or feedback error learning controller.[21] Several works have been carried out on the cerebellar internal model theory,[22–26] and two main control alternatives have been advocated, which are known as feed-forward,[21,27–30] and recurrent control loops;[13,20,26,31,32] a comparison among the two schemes is also available in the literature.[33] Biologically plausible cerebellar algorithms have been successfully embedded in the aforementioned loops for validating the cerebellar adaptive functions in different tasks, e.g. in arm manipulation adaptation under perturbations,[19,21,30,34,35] in eyeblink classical conditioning,[36,37] in gaze stabilization through the vestibulo-ocular reflex with a humanoid robot,[29,38] in gain and timing adaptation for a target reaching task,[39,40] in minimizing sensory discrepancy and cancelling the noise.[31] Although these cerebellar-like circuits differ in the synaptic plasticity models or network models, they all analyzed how the cerebellum is able to adjust the motor output or to predict the action and minimize the sensory discrepancy.

The recurrent cerebellar loop, based on the use of the forward model, was originally proposed by Porrill *et al.*[31] to simplify the adaptive control of nonlinear and redundant biological motor systems; in their experiments, the cerebellum had to compensate for miscalibrations in a two deegrees of freedom (DOFs) planar arm. Another attempt to show how the recurrent adaptive control algorithm can successfully improve the performance of a robotic implementation for gaze stabilization was made by Ref. 32. They came to the conclusion that the architecture was able to adapt to a time varying plant and so, it could be employed in general engineering

applications where sensory feedback is very noisy and/or delayed. Luque *et al.*, in 2011,[20] were one of the first authors who evaluated the capability of a spiking cerebellar model embedded in the recurrent loop architecture to control a robotic arm of three degrees of freedom. Two years later, Tolu *et al.*,[13] proposed a new bio-inspired recurrent architecture in which the feedback-error-learning[41] and adaptive predictive control strategies were combined. Both of the previous studies reported the advantage of employing the recurrent architecture to ensure a fast convergence in learned profile curve dynamics. In contrast with Ref. 20, Tolu *et al.*[13] guaranteed the system capability to be robust against both dynamics and kinematics transformations even in nonlinear plants with seven DOFs. Although of significant importance, this study did not fully exploit the potential of the forward model predictions, as they were just used to provide an input-space representation to the cerebellum-like network and not for improving the corrections based on the prediction error.

1.2. *Paper outline*

Section 2 gives an explanation of the adopted control strategy and a description of each main component with the equations included. In this regard, the properties and characteristics of the machine learning engine, of the cerebellar microcircuit, and of their connection are presented and analyzed. Section 3 begins by outlining the four sets of experiments that include highly dynamic tasks under varying modular set-ups. Finally, in Secs. 4 and 5, we discuss the performance and the robustness of the proposed control method in learning forward models and compensating to deviations in the target trajectory together with concluding remarks.

2. Material and Methods

2.1. *The cerebellar-like adaptive smith predictor*

The proposed control strategy is illustrated in Fig. 1(c) and it will be referred as the cerebellar-like adaptive Smith predictor in the rest of this paper. A set of four main blocks is created for each active robot module and each block is also internally split up in terms of DOFs. The blocks are: the trajectory
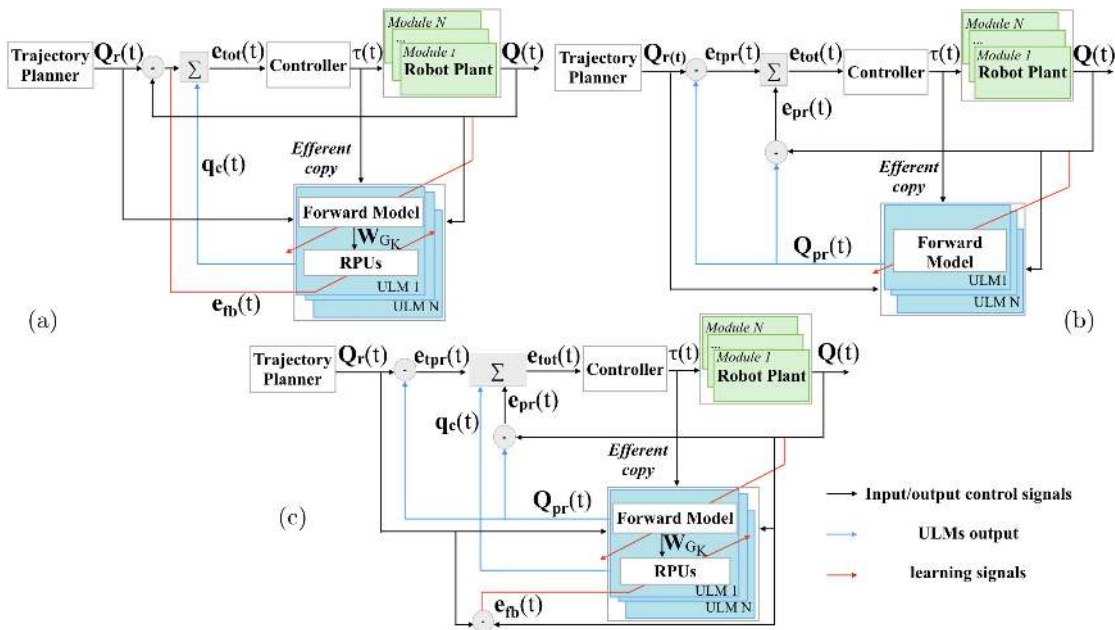
Fig. 1. Three modular control architectures (a, b and c) were used for outcome comparison between the $\mathbf{Q}_r(t)$ reference joint positions and the $\mathbf{Q}(t)$ actual joint positions. The derivative terms are not depicted for clarity. In each main block, N sub-blocks are created, where N is the number of active robot modules. Every blocks receives N for J signals, where J is the number of joints per module. (a) This is the RAFEL control loop proposed in Ref. 13, (b) This is the Smith predictor strategy (c) The proposed architecture is depicted here. It is based on both the cerebellar-like sensory corrections and the Smith predictive mechanism.

planner, the controller, the Unit Learning Machine (ULM), and the robot plant.

Two previous control strategies presented in the literature have been combined. The first one is shown in Fig. 1(a) and it is known as the recurrent adaptive cerebellar-based control loop (RAFEL).[13] In RAFEL, inside the ULM or cerebellar-like microcomplex, the *Forward Model* plays the role of the granular cerebellar layer that sends the W_{GK} preprocessed weights (K is the number of receptive fields (RFs)) to the *Readout Plasticity Units* (RPUs) sub-component that includes the Purkinje cerebellar layers. The cerebellar corrective output (q_c) of the RPUs is summed to the feedback error (e_{fb}) into a total error signal (e_{tot}) that is converted by the controller into a τ torque command to the joint actuators. The second control strategy is the Smith-based control mechanism that is shown in Fig. 1(b). It is based on a fast internal feedback loop ($\mathbf{Q}_{pr}(t)$ forward model predictions) and on an external feedback loop in which errors are converted by the controller into a τ torque command to the joint actuators. There are no cerebellar-like corrections acting in the loop.

Further details about each block in terms of learning and prediction mechanisms, processing flow, and interconnections, are given in the following subsections.

2.2. *The control loop description*

The trajectory planner delivers the reference or $\mathbf{Q}_{r,j}(t)$ desired joint positions for the Fable robot to follow. Ideally, if there were no noise, disturbances, or delays during the functioning of the system, the desired positions and velocities and the actual outcomes from the robot plant would match thanks to the torque control command provided by the controller. However, a correction that takes into account if any mismatch occurred in the robot plant output, has to be added to the desired movements and the sum is converted into a τ motor command signal through the controller. A delayed copy of the $\mathbf{Q}_{pr}(t - \delta)$ forward model prediction is compared with the $\mathbf{Q}(t - \delta)$ previous sensory feedback in output from the robot plant. The result of this comparison is fed into the system to correct the movement if necessary. The equations presented below are

referred to position contributions. Velocity contributions can be derived in a similar way, but they have not been explicit for clarity.

In Fig. 1(c), the controller receives $\mathbf{e}_{\text{tot},j}(t)$ control input i.e. the sum of the following contributions (1)

$$
\begin{aligned}
\mathbf{e}_{\text{tot}_j}(t) &= (\mathbf{Q}_{r_j}(t) - \mathbf{Q}_{\text{pr}_j}(t)) + (\mathbf{Q}_j(t - \delta) \\
&\quad - \mathbf{Q}_{\text{pr}_j}(t - \delta)) + \mathbf{q}_{c_j}(t) \\
&= \mathbf{e}_{\text{tpr}_j}(t) + \mathbf{e}_{\text{pr}_j}(t) + \mathbf{q}_{c_j}(t),
\end{aligned}
\tag{1}
$$

where $\mathbf{e}_{\text{tpr},j}(t)$ is the trajectory joint error, $\mathbf{e}_{\text{pr},j}(t)$ is the prediction joint error, and $\mathbf{q}_{c,j}(t)$ is the deep-cerebellar nuclei (DCN) cell behavior or cerebellar joint corrective position computed as in Eq. (8). Each term is associated to the j_{th} joint. The $\tau(t)$ torque command for the j_{th} joint is obtained by the following equation:

$$
\begin{aligned}
\tau_j(t) &= k_{p_j} \cdot \mathbf{e}_{\text{tot},j}(t) + k_{d_j} \cdot \dot{\mathbf{e}}_{\text{tot},j}(t) \\
&\quad + k_{i_j} \cdot \frac{d\dot{\mathbf{e}}_{\text{tot},j}}{dt}.
\end{aligned}
\tag{2}
$$

The control signal is thus a sum of three terms: a first term that is proportional to the position error, a second term that is proportional to the velocity error, and a third term that is proportional to the derivative of the velocity term.

The controller parameters for each joint j are: the proportional gain k_p, the derivative gain k_d, the integral gain k_i, and the natural frequency w_j that determines the speed of response of joint j. The parameter values are shown in Table 1 for each joint and they were intentionally tuned to make the controller initially stable, but not optimal and robust to disturbances, during the execution of the task. Indeed, the purpose of the work is to show how the RPUs adaptive module and the Smith predictor strategy are together contributing to the improvement of the control in an adaptive way. In the work done, both joint velocity and position corrections have been taken

Table 1. The proportional (k_p), derivative (k_d), and integral (k_i) gains for joints 1 and 2. The natural frequency w_1 (joint 1) is equal to 4.5, and w_2 (joint 2) is equal to 5.0.

Joint $N.°$	k_p	k_d	k_i
1	$w_1^2 \cdot 2$	$w_1 \cdot 0.4$	$w_1/40.0$
2	$w_2^2 \cdot 4$	$w_2 \cdot 0.4$	$w_2/40.0$

into account for adaptive control. Equations in the text refer to the position terms, but a similar analysis could be derived for the velocity terms. Here, they are omitted for clarity.

2.2.1. *The unit learning machine*

The ULM is a set of structured cerebellar neural circuits that encodes internal models in order to precisely perform the control of the body part.[42] Forward models are formed and adjusted through a supervised learning process as the movement is repeated to mimic the behavior of a natural process,[43] to facilitate more precise coordinated movements[44] and to increase the system control compliance.[45] The main advantage of this approach is to allow the design of control systems that can operate in unknown or changing environments, when the dynamical robot model is unknown[13,21] or every time a subject learns to use a new tool or uses different tools.[23]

The network of ULMs is inspired on the cerebellar modular neural organization proposed by Ref. 46; the modularity of the ULMs is achieved by replicating them in terms of the number of robot modules (N) with an identical parameter configuration; one set of ULMs is dedicated to position terms and another one to velocity terms, but this is not explicitly depicted in Fig. 1 for clarity. Each ULM artificially reproduces the main cerebellar layers of the canonical micro-complex circuit shown in Fig. 2 that were grouped in two sub-blocks: the *Forward Model* and RPUs. The *Forward Model* and *RPUs* components that are dedicated to position learning are separated from the ones for velocity learning. The *Forward Model* sub-block is dedicated to learn the internal forward model of a single robot module, but it receives the sensory input data from every robot module that is active and connected. If another robot module is connected to the active ones, an extra ULM is created and all the ULMs start receiving data also from it. The *RPUs* output inside the ULM block is the adaptive cerebellar correction for each robot module as represented in Figs. 1(a)–1(c).

Forward Model. The *Forward Model* component receives and processes the sensorimotor information during the entire experiment execution to provide an optimal incremental input space representation likewise in the cerebellar granular layer in terms
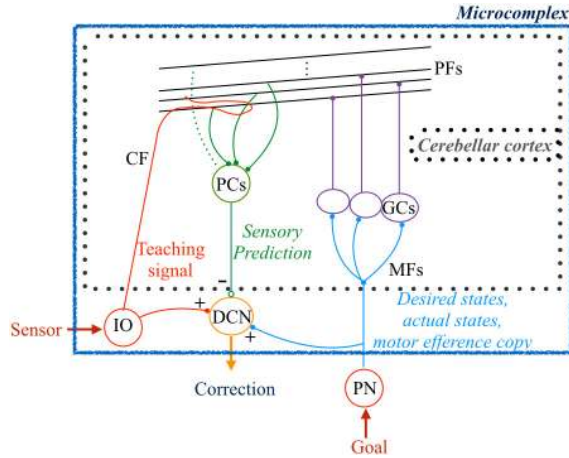
Fig. 2. The cerebellar functional neural unit is called microcomplex. Mossy Fibers (MFs) and climbing fibers (CFs) are the principal input to the cerebellar cortex. MFs that originate from the spinal cord and from a wide range of nuclei in the brain stem, namely PN, distribute multisensory information, which refers to the desired movement, motor commands, and the actual state of the limbs, onto different GCs. GCs excite the main output of the cerebellar cortex, the Purkinje cells (PCs) via parallel fibers (PF)-PC synapses. The other afferent of the cerebellum, the CF, arise from the IO and is thought to convey an error computed during the movement, thus providing the teaching signal for the cerebellar learning. Other cell types such as the Golgi, the basket, and the stellate cells are not depicted in the figure for simplicity.

of RFs.[13] The RFs are generated online along the execution of the trajectory. The LWPR algorithm[47] accomplishes the learning of the forward dynamic model of a robot module, while it is subjected to corrections during the reaching task.[13] So, the LWPR engine is running in the *Forward Model* sub-block inside the ULM with multiple functions: (1) it learns the forward model of the robot; (2) it predicts the $\mathbf{Q}_{\mathrm{pr}}(t)$ future state of the robot module; (3) it computes the W_{Gk} granular weights from the sensorimotor information coming through the MFs and it provides them to the RPUs.

Among the global nonlinear function approximators, such as Gaussian Process Regression (GPR), or Support Vector Regression (SVR), LWPR method has been selected since it approximates nonlinear functions in high-dimensional spaces at a low computational cost. It automatically and efficiently evaluates the required number of local correlation modules to optimize the network size by incremental learning, the Partial Least Squares regression (PLS). Because

of this, the LWPR algorithm does not require any prior knowledge of the data, thus allowing fast online learning. Moreover, there is no restriction in the number of different tasks that can be retained and interposition among them. All the previous facilitate the cerebellar learning accuracy and speed, thus confirming the hypothesis of Ref. 31 that they could be improved by an optimized granular sparse code, i.e. a neural code in which the ratio of active neurons is low at any time. Furthermore, more recent studies on cerebellar physiology link the PCs responses necessary for motor learning to learning-related neural responses in granular cells. Thus, the learning capability of the cerebellum is highly correlated to the changes in the granular layer during motor control.[8] Bearing in mind the similarity between the LWPR learning mechanism and the cerebellar circuit (see Fig. 1 in Ref. 21), each LWPR module and its associated RF weights can be seen as providing the firing rate of a PF, while the set of active RF weights can be seen as the current state of the granular layer-processing module.

From Fig. 1(c), the $\mathbf{Q}_{\mathrm{pr}}(t)$ outputs of the *Forward Model* sub-block represent the predicted future states or positions of the robot and they are used for addressing a fast internal feedback control. A schematic representation of the input/output signals to the learning and prediction functions of the LWPR algorithm is depicted in Fig. 3. The learning function is trained to acquire the forward model of the robot module. The predict function is called to estimate the future state $\mathbf{Q}_{\mathrm{pr}M}(t)$ of the robot module M based on the previous $\mathbf{Q}(t - \delta)$, $\dot{\mathbf{Q}}(t\delta)$) robot plant state, the previous $\mathbf{Q}_r(t - \delta)$, $\dot{\mathbf{Q}}_r(t - \delta)$) desired robot plant state, and the previous $\tau(t - \delta)$ efferent command copy sent to the robot, where δ is the inherent delay of the control of the system.

Besides, the LWPR incrementally divides the input space into a set of K RFs whose $W_{Gk}(t)$ time-varying parameters are defined in (3) by the c_k center and a Gaussian area characterized by a positive definite D_k distance matrix

$$W_{Gk}(t) = e^{-\frac{1}{2}(x_i(t)-c_k)^T D_k(x_i(t)-c_k)}, \qquad (3)$$

where $x_i(t)$ are the input data points at sample t ($i \in [1, M]$, M is the total number of inputs), and k is a local model or RFs ($k =\in [1, K]$, K is the total number of RFs).
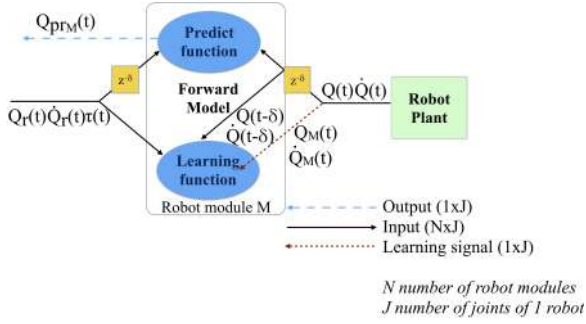
Fig. 3. At every moment in time (t), the predict and learning functions of the LWPR algorithm for a robot module (M) receive input signals from all the J joints of all N active robot modules. The forward model prediction outputs ($\mathbf{Q}_{\mathrm{pr}M}(t)$) related to robot module M are then simultaneously employed as a feedback signal to compute the torque commands that will drive the robot plant.

At each sample t, the $x_i(t)$ points are assigned to the closest RF based on its $W_{Gk}(t)$ weight activation, and consequently, the c_k center is incrementally updated. The number of local models increases with the complexity of the input space. If an input data falls into the validity region of a local model, its own distance matrix and regression parameters are updated. Likewise in the cerebellum processing flow, the $W_{Gk}(t)$ RF weights (3) are processed by the RPUs sub-component.

The $\mathbf{Q}_{\mathrm{pr},j}(t)$ global forward model output for the j_{th} joint of a robot module is computed as the weighted mean of all the $\hat{\mathbf{q}}_k(t)$ predictions of the k_{th} linear local models created (4).

$$Q_{\mathrm{pr},j}(t) = \frac{\sum_{k=1}^{K} w_{gk}(t)\hat{q}_k(t)}{\sum_{k=1}^{K} w_{gk}(t)}. \tag{4}$$

In other words, the input data points $x_i(t)$ enters every RF (local model), which provides the $\hat{q}_k(t)$ prediction seen in (4).

The Readout Plasticity Units. Inside the *RPUs* the following cerebellar layers are represented:

- **Purkinje cell (PC) layer:** the activity at this layer is function of the granular layer state. The learned parameters are modified during the execution of the planned movement by the error or teaching signal.
- **DCN cells:** The response is related to the integration of both excitatory and inhibitory contribution from MFs, inferior olive (IO) and PCs, respectively.

Each PC and DCN is associated with a joint of the robot. Several forms of plasticity mechanisms work in balance at different layers to provide a final corrective control signal to perform the desired robotic task.

More in detail, the RPUs set provides the $\mathbf{q}_c(t)$ cerebellar corrections, where t states the current sample. These corrections are modulated through the $\mathbf{e}_{\mathrm{fb}}(t)$ teaching signal computed by the difference between the $\mathbf{Q}_r(t)$ reference joint positions and the $\mathbf{Q}(t)$ actual joint positions of the robot plant, obtained from the encoders of the motors. The dimensionality of the previous-mentioned vectors is

$$Q \in \mathbb{R}^n, \tag{5}$$

where n is number of DOFs of each robot module.

Unlike the work in Ref. 13, the here adopted cerebellar-like model was inspired from the biological cerebellar micro-complex circuit shown in Fig. 2 that includes extra plasticity rules based on the cerebellar theories described in Ref. 48. Thus, inside the RPUs sub-block, the PF-PC, PC-DCN, MF-DCN and IO-DCN synaptic connections are implemented. Our main aim is to maintain the functional cerebellar information processing by using artificial cells with analog activity values. In the context of movement control, several studies,[25,42,48,49] suggested that the CFs that arise in the IO, may signal the presence of an unexpected sensory stimulus during an action, which nature depends on the type of the internal model, so they may convey the error signals that drive a long-term depression (LTD)-based learning process at parallel-Purkinje (PF-PC) synapses. In the case of forward model learning, the CFs may convey either sensory signals. Apart from the PF-PC synaptic plasticity, the plasticity at the DCN synapses can account for learning.[19] DCN neurons are innervated with both excitatory and inhibitory connections: the first from CFs and MFs, and the second from PCs. Their relationships are still not completely revealed, but it seems that the IO-DCN plasticity provides fast corrections at early stages during the learning.[48]

The $W_{Gk_j}(t)$ weights are then processed for computing the $\mathbf{q}_{c_j}(t)$ cerebellar corrections on the basis of the $\mathbf{e}_{\mathrm{fb}_j}(t)$ feedback error or teaching signal carried out by the CF from the IO associated with the j_{th} joint. Thus, the specific $\mathrm{PF}_{k_j}(t)$ pathway to the Purkinje layer carries the $W_{\mathrm{Gk}_j}(t)$ signal to the

$PC_j(t)$ synapse. The $PC_j(t)$ output represents the firing rate of the PCs associated with the j_{th} joint and it is modeled as a weighted linear combination of the $W_{Gk_j}(t)$

$$PC_j(t) = \sum_k W_{(PF-PC)_kj}(t) \cdot W_{Gj}(t). \quad (6)$$

$$\Delta W_{(PF-PC)_kj}(t) = \beta \cdot CF_j(t) \cdot W_{Gj}(t), \quad (7)$$

where $\Delta w_{PFk-PC_j}(t)$ represents the weight change between the k_{th} PF and the target PC associated with the j_{th} joint, β is a small positive learning rate set to $5 \cdot 10^{-3}$ ($7 \cdot 10^{-4}$ for the velocity configuration) in all the experiments, $CF_j(t)$ signal corresponds to the $e_{fb_j}(t)$ teaching signal. According to Fig. 2, the DCN receives input signals from two differentiated pathways. The first pathway comes from the cerebellar cortex trough PCs and the second is through MFs and CFs. The roles of the aforementioned pathways in motor control were discussed in Ref. 48. Equation (8) describes the DCN cell behavior

$$DCN_j(t) = W_{MF-DCN_j} - PC_j(t) \cdot W_{PC-DCN_j}$$
$$+ CF_j(t) \cdot W_{IO-DCN_j}, \quad (8)$$

where $DCN_j(t)$ represents the firing rate of the DCN cell associated to the j_{th} robotic joint; W_{MF-DCN_j} is the synaptic strength of the MF-DCN connection at the joint j_{th}; W_{PC-DCN_j} represents the synaptic strength of the PC-DCN connection of the j_{th} joint; $CF_j(t)$ corresponds to the $e_{fb_j}(t)$ teaching signal associated to the j_{th} joint and W_{IO-DCN_j} represents the synaptic strength of the IO-DCN connection of the j_{th} joint. The sensory information is then related with the corrective cerebellar output $DCN_j(t)$. We have integrated in our previous cerebellar-like circuit[13] the rules described in Eqs. (9), (10) and (11) that represent the W_{PC-DCN_j}, W_{MF-DCN_j}, and W_{IO-DCN_j} synaptic plasticities of the j_{th} joint. However, some changes related to the parameters and signals suggested in Ref. 48 for the robotic simulations were applied in the implementation with the real robot. All the synaptic strengths are progressively adapted during the learning process according to different synaptic plasticity mechanisms which are represented by the following equations:

$$\Delta W_{PC-DCN_j}(t) = \frac{LTP_{MAX} \cdot |PC_j(t)|^\alpha}{(1. + abs(DCN_j(t)))^\alpha}$$
$$- LTD_{MAX} \cdot |PC_j(t)|, \quad (9)$$

where LTP_{MAX} and LTD_{MAX} are the maximum long-term potentiation/long-term depression values set to $1.0 \cdot 10^{-4}$ and $1.0 \cdot 10^{-3}$, respectively ($1.0 \cdot 10^{-5}$ and $1.0 \cdot 10^{-4}$ for joint velocity configuration); α is the LTP decaying factor that was set to 10 in order to maintain stable behavior versus a fast LTP action decreasing; a faster LTP action (1000) as proposed in Refs. 19 and 48 affects the control stability of the system.

$$\Delta W_{MF-DCN_j}(t) = \frac{LTP_{MAX}}{(|PC_j(t)| + 1)^\alpha}$$
$$- LTD_{MAX} \cdot PC_j(t)), \quad (10)$$

where LTP_{MAX} and LTD_{MAX} are the maximum long term potentiation/long terms depression values set to $1.0 \cdot 10^{-4}$ and $-1.0 \cdot 10^{-3}$, respectively ($1.0 \cdot 10^{-5}$ and $-1.0 \cdot 10^{-4}$ for joint velocity configuration); α is the LTP decaying factor that was set to 10 in order to maintain stable behavior versus a fast LTP action decreasing.

$$\Delta W_{IO-DCN_j}(t) = MTP_{MAX} \cdot |CF_j(t)|$$
$$- \frac{MTD_{MAX}}{(|CF_j(t)| + 1)^\alpha}, \quad (11)$$

where MTP_{MAX} and MTD_{MAX} are the maximum modulating plasticity terms set to $1.0 \cdot 10^{-3}$ and $-1.0 \cdot 10^{-6}$, respectively, for both joint position and velocity configurations; α is the MTD decaying factor that was set to 10 in order to maintain a positive ratio between behavior versus fast MTD action decreasing. The $CF_j(t)$ activity in the IO-DCN plasticity represented the $e_{fb(t)}$ feedback joint position error instead of the normalized joint feedback error as used in Ref. 48.

In order to both preserve the control loop stability and refer to the bio-inspired control system structure for voluntary movements,[50] the feedback error learning theory presented in Ref. 51 was took into consideration. In our work, the feedback error learning mechanism is obtained by using the Smith predictor strategy whose anticipatory actions contribute toward achieving a fast improvement of the performance and fast learning cerebellar consolidation. As a matter of fact, the cerebellar-like learning is facilitated by the supervised learning in terms of feedback error carried trough the CF in conjunction with the PF-PC synapses.[41] In addition to this, the prediction error obtained by comparing the internal forward model outcome with the

current robot plant outcome, indirectly contributes
to the learning and adaptation of the ULM mod-
ule through the generation of the optimal τ control
input.

2.2.2. *The modular robotic system*

Fable is a modular robot consisting of detachable
modules.[14] The modules can be active (with motors)
or passive and can be assembled together to shape
configurations that differ in morphology and topol-
ogy. Each module has two actuated revolute joints.
In this work, tests on a single active module with
2-DOFs and on two connected active modules (4-
DOFs) lying in the configuration setup shown in
Fig. 4(a) were conducted. Joints are actuated by
Dynamixel AX-12A motors with an accuracy of
0.29°. Modules receive a command wirelessly from
the user computer that is serially connected to a don-
gle. The dongle provides a shared 2 Mbit radio com-
munication link between the user controlled applica-
tion and the modules. Modules are addressed using
an ID and their module type. The web services
(API) allow distributed control; a separate thread
is executed on the computer for each module con-
troller. The system application, which the user can
run locally on the computer, has been developed
using Python v.3.4. The user-program calls simple
APIs functions on the remote modules trough the
dongles connected to the computer, as shown in
Fig. 4(b).

The control of the Fable robotic system has an
inherent delay: each control-loop execution takes
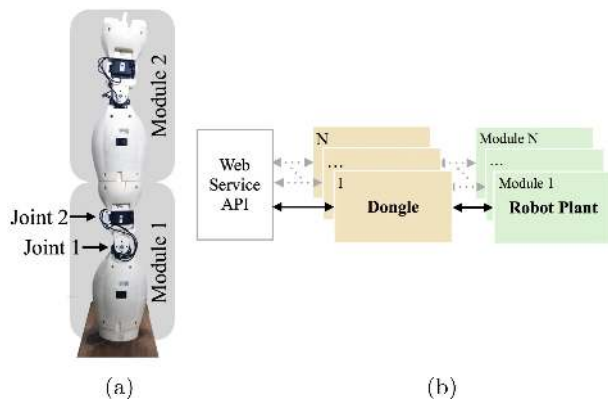approximately 5 ms in a computer; once the torque is

Fig. 4. (a) The Fable modular robot composed of mod-
ule 1 and module 2. (b) The communication system.

estimated, it is sent to the joint through the dongle
which has a latency of about 1 ms. Besides this, there
is an unknown variable delay related to the operative
system, which is lower than 0.1 ms.

3. Experiments and Results

The outcome of the three control architectures shown
in Fig. 1 has been compared and analyzed in this sec-
tion. In all the following tests, the two-joint modules
had to follow the figure-eight trajectory described by
the following equations in angular coordinates:

$$Q_{rj1}(t) = -A\left(\frac{1}{2\pi}\right)^2 \sin(2\pi t),$$
$$Q_{rj2}(t) = -2A\left(\frac{1}{4\pi}\right)^2 \cos\left(4\pi t + \frac{\pi}{2}\right),$$

(12)

where $A = 2000$; the amplitudes of joints 1 and 2
are approximately 50.0° and 25.0°, respectively. Each
completed figure-eight movement constitutes an *iter-
ation*, which duration is 1 s (the sampling time is
equal to 5 ms).

The performances of the proposed architectures
depicted in Fig. 1 are analyzed by executing four
different tasks that allow proper assessment of the
learning capability of the cerebellar model.

In each task, the performance is measured by
computing the Mean Absolute Error (MAE) between
the reference joint trajectory $(Q_{r,j})$ and the actual
robot joint outcome (Q_j) along the whole trajectory.
During all the trials, there is no offline learning, i.e
the learning of the forward model starts when the
simulation begins, so the cerebellum-like weights are
activated; however, the Smith prediction as well as
the cerebellar corrections are applied into the sys-
tem after one initial iteration of the figure-eight. This
provides the initial stability given by the feedback
controller for the learning phase. The robot module
batteries have to be fully charged in order to get the
best performance.

1. *Basic Task.* Basic operation with one robot mod-
ule. The robot module stands in a vertical position
and the initial joint angles are set to zero. The figure-
eight movement is repeated 200 iterations with a
fixed frequency and amplitude.

The first experiment simply compares the per-
formance of the three architectures shown in Fig. 1.
Results for the architecture proposed can be found
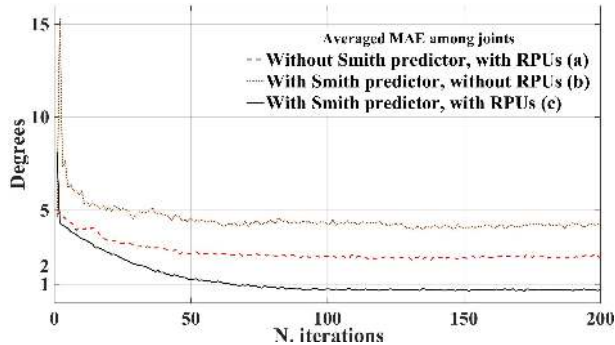in Fig. 5. It can be observed (black line) that the

Fig. 5. (Color online) Accuracy in position of the control loop obtained with one robot module in three test-cases (architectures a, b and c in Fig. 1): (1) with both the Smith predictor and cerebellar-like corrections (black solid line); (2) with the Smith predictor contribution and without the cerebellar-like (RPUs) corrections (brown dot dashed line); (3) without the anticipatory action provided by the Smith predictor and with the RPUs corrective actions (red dashed line). Corrections are applied to the desired joint coordinates when the robot is following the target trajectory.

adaptive and predictive control method is able to adapt to robot dynamics by making the averaged MAE decrease to 0.70° at the last iteration.

The effectiveness of the control can be observed by comparing the result with the one obtained without the cerebellar-like outcome (brown dot dashed line). The outcome from the RPUs is actually what

leads to finer adjustments in the control. The task was also executed without the Smith predictor (red dashed line) to show that the contribution of the RPUs helps to anticipate the correction from the very beginning of the trial (see Fig. 5). The experiment was repeated eight times for the three control loops and the standard deviation (std) values corresponding to the minimum MAE values are indicated in Table 2. They are an indicator of how much, on average, measurements differ from each other. The RPUs corrections help to maintain the STD values low and the accuracy is higher than the case without them.

The number of RFs created by the LWPR was 11 for the position states (see Table 3). The distance matrix D_k was set to 10.5 for all the experiments. The initial D_k value, which describes the size of the newly created RFs, was chosen by analyzing the prediction error obtained, as shown in Table 3. Smaller values for D yield wider RFs, which oversmooth at the start of training and lead to slow convergence until they span the entire input space; it tends to reduce the generalization ability; bigger values for D yield smaller RFs, which might lead to fast convergence, but also to overfitting.

2. *Trajectory switching task.* Trajectory switching with one robot module in the same initial configuration of task 1.

The second set of experiments focuses on the controller response to a set of sinusoidal inputs at different amplitudes and frequencies that are changed online every 30 iterations of the figure-eight movement. In the first test, the amplitude of the movement was changed every 30 iterations, with A ranging in the $[2200, 2000, 1800]$ set for both joints, while the frequency remained constant. Results for this

Table 2. Standard deviation and min MAE values (degrees) for the basic experiment task.

Control loop	min MAE	std
a	2.34	0.04
b	3.99	0.07
c	0.62	0.02

Table 3. Testing the performance of 1-module basic task by varying the distance matrix D that is inversely proportional to the diameter of the RFs. The table data show the D matrix value, the number of RFs, the minimum MAE value for each joint and averaged among the joints, and finally, when the latter occurred during the whole executed simulation. The best result is highlighted in bold.

	Distance matrix D	5.0	7.0	9.0	**10.5**	15.0	17.0
	N° RFs	6	8	9	**11**	13	15
Test	Min MAE joint 1 (∘)	1.07	0.78	0.73	**0.65**	0.75	0.67
basic task	Min MAE joint 2 (∘)	1.27	0.94	0.88	**0.72**	0.80	0.84
	Min avg. MAE (∘)	1.17	0.86	0.81	**0.69**	0.78	0.76
	N° iteration	97	100	102	**102**	96	98

Int. J. Neur. Syst. Downloaded from www.worldscientific.com by TECHNICAL UNIVERSITY OF DENMARK on 11/27/19. Re-use and distribution is strictly not permitted, except for Open Access articles.
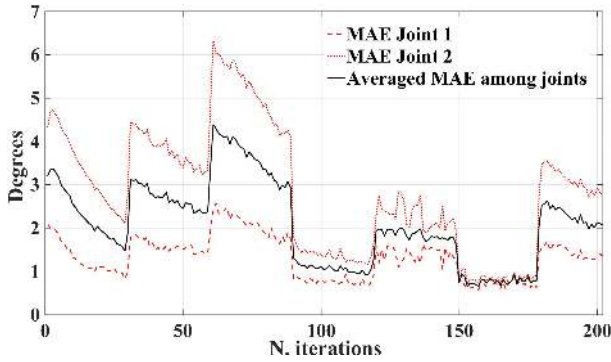

Fig. 6. Test of amplitude change carried out with the model architecture in Fig. 1(c). The sequence was: A = [2200, 2000, 1800, 2000, 2200, 2000, 1800]. The switches occur after every 30 iterations.

experiment are shown in Fig. 6, where it can be observed that the predictive architecture is robust to the changes during the executed action, since the error decreases faster after a switch especially in advanced learning phase. The number of RFs increased to 20 in this experiment.

In the second experiment, the amplitude of the movement remained constant, while the frequency was changed after every 30 iterations, ranging in the $[0.9, 1.0, 1.2 \, \text{Hz}]$ set. Figure 7 shows the behavior obtained with one robot module. When a change occurred, the cerebellar-like adaptive Smith predictor performance is valuable in terms of averaged MAE, it decreased around $1°$. The number of RFs also increased to 20 during this experiment despite fast changes of the frequency. Thanks to the previous learning acquired by the RPUs during the changes, the system is able to provide fast adjustments and
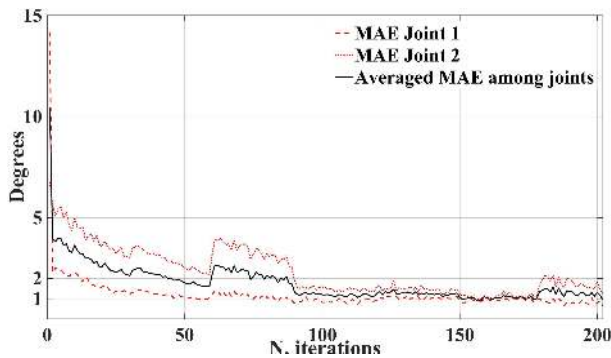

Fig. 7. Test of frequency change carried out with the model architecture c in Fig. 1. The sequence was: f(Hz)=[0.9, 1.0, 1.2, 1.0, 0.9, 1.0, 1.2] and the switches occur after every 30 iterations.

maintain a stable behavior that is especially perceived after 100 iterations.

3. *Dynamic changes task.* The robot module stands in the same initial configuration of task 2. In this task, the capabilities of the proposed architecture to react to changes in dynamics were tested by applying an external load on the robot end-effector. The load weighted $105.9 \, \text{g}$ and was applied by hand from the beginning of the trial in the experiment shown in Fig. 8, and after 40 iterations in the test shown in Fig. 9.

It can be noticed that the cerebellar-like adaptive Smith predictor does not seem to be influenced by the change in dynamics, and the cerebellum is
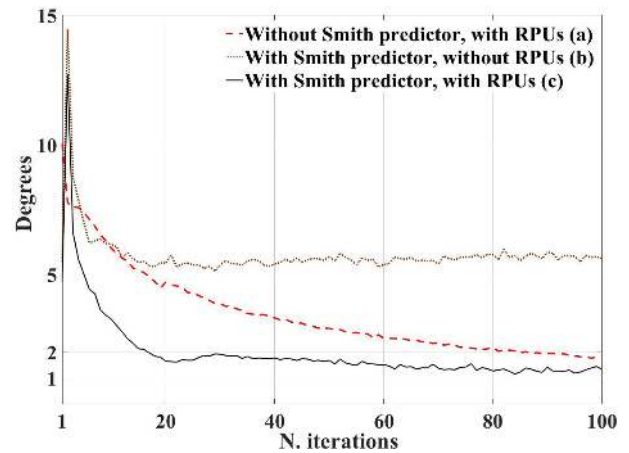

Fig. 8. The averaged MAE obtained manipulating an external load at the end-effector from the beginning of the test.
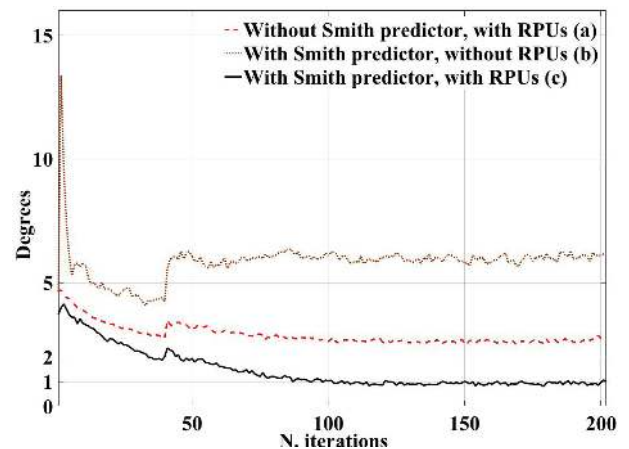

Fig. 9. The averaged MAE obtained manipulating an external load at the end-effector after 40 iterations from the beginning of the test.

actually the one lowering the error soon after few iterations after the load application. Moreover, the initial MAE value is lower than the ones obtained without cerebellar contributions. As a matter of fact, the cerebellum is able to apply adaptive corrections thanks to the previous experience and learning, which is not allowed with only the Smith predictor. However, their combination helps refining the corrections to get a faster robust performance.

4. *Basic task with two Fable modules.* Basic operation and dynamic changes with two robot modules. The amplitude and frequency of the figure-eight movement are fixed and repeated by two separated robot modules. The dynamics of both robot modules change.

The final set of experiments focuses on the controller capability to deal with the dynamic disturbances caused by the connection of two modules and demonstrates its effectiveness for modular robots. In this task, two Fable modules were concatenated one on top of the other forming a column structure (Fig. 4(a) and oriented top–down to overcome the payload engine limitations. Both joint modules had to follow the figure-eight trajectory given by equation 12.

The results represented in Fig. 10 are given by the two modules attached together from the beginning, differently from Fig. 11 for which experiment the two modules were initially separated and only after 30 iterations were joined. In Figs. 10 and 11,
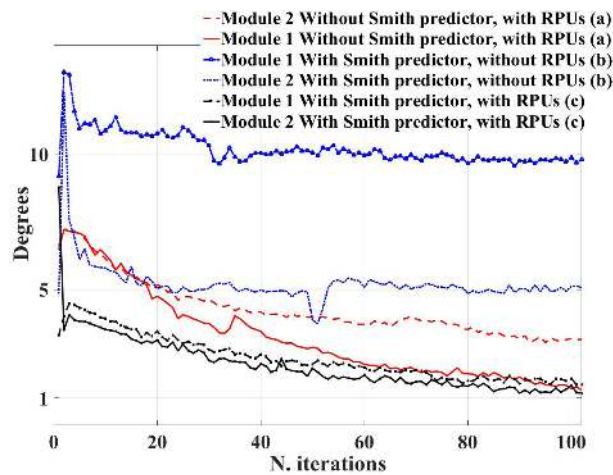

Fig. 10. The averaged MAE obtained with the modular setup as result of the three control architectures a, b, and c shown in Fig. 1. The two modules are joined from the beginning of the test.
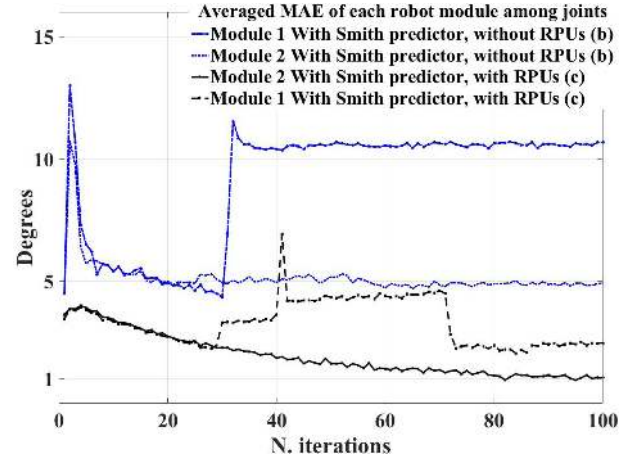

Fig. 11. The averaged MAE obtained with the modular setup when one module is attached to the other after 30 iterations from the beginning of the test.

the averaged MAE among joints is around 2° and 1° at the last iteration for module 1 and 2 respectively, by using our proposed control method. In Fig. 10 the outcome of the three control architectures in Figs. 1(a), 1(b) and 1(c) was compared. Again, the combination Smith predictor and cerebellar feed-forward controller is the one making the difference in the control during the experiment compared to the other control strategies. The number of RFs created by the LWPR was equal to 20 for the experiment in Fig. 10 and 27 for the experiment in Fig. 11. Finally, the outcome in Fig. 11 highlights the high contribution of the cerebellum-like network with the Smith predictor strategy. Indeed, the cerebellar-like corrections enable a better robot performance with the Smith predictor from the beginning of the test and a fast reduction of the error after the two modules are concatenated.

4. Discussion

4.1. *Forward cerebellar model contribution to robustness and generalization capabilities*

As anticipated, our findings support the hypothesis that the cerebellum could use the Smith predictor strategy based on forward model learning to provide fast and accurate corrections. The sensory signals were distributed in order to drive the learning and compensate the time delays inherent to the control system.

Test results, in Sec. 3, about the performance and validity of our controller in on-line motor control tasks revealed that: (1) the error decreases significantly and fast during the tasks; (2) the control is robust against changes in dynamics or in motion input. The errors are compensated throughout the generation of the internal forward model, prediction errors, and cerebellar corrections. The results show an improvement in the accuracy compared to a classic control approach and that the combination of the Smith predictor with a forward cerebellar model indeed contributes to deliver fine and effective corrections (Fig. 5). The proposed control architecture presents a lower final error than the one achieved without the Smith predictor and without cerebellar-like corrections during the control of the task. We argue that the angular error is due to limits of the hardware (gear set) and the motor's resolution. It is worth to mention that the system is friendly with respect to the parameters setup. As a matter of fact, all the experiments were run without any modification in the internal control parameters. Besides, an optimal feedback controller as well as an analytical robot model are not needed.

4.2. *Implications for cerebellar role in anticipatory control*

The results from experiments to changes in motion (Figs. 6 and 7) and in dynamics (Figs. 8, 9, 10 and 11) have confirmed that Smith control predictions help to deal with the sensory delay of the control loop and the modular RPUs adjust the control output to those changes. Indeed these are central features of cerebellar computation and function and they have been investigated in a recent work.[52]

All the tests were performed on a physical modular robot in contrast with previous works based on the recurrent cerebellar architecture in simulated robotic tasks.[13,20] Luque *et al.* (2011)[20] evaluated the capability of a spiking cerebellar model embedded in a recurrent control loop to control a simulated robot arm. In their work, they were able to get as minimum MAE error approximately 4° after 100 iterations of the same task. In contrast with their work, here both feedback-error-learning mechanism and predictive actions of the robot plant were used to enable a better improvement in the robot control performance. This confirms the role of cerebellum in

driving the performance and at the same time getting benefits from the learned output.[8]

In contrast with previous approaches,[24,53] there is no offline learning. This has two effects: from one side, it allows the system to adapt to changes in dynamics while performing the task, but it has the downside of having an initial learning phase in which the error is higher. However, it has been shown, with similar controllers, that this issue can be overcome by storing the learned forward model.[29] Anyhow, results show that active cerebellar corrections lower initial errors from the first trials of the experiment. In our model, at the same time, the forward model predicts the future state, the cerebellum updates its synaptic weights and computes its corrections based on the same sensory inputs of the former. This could explain how the sensory signals are distributed in order to drive the learning and anticipate corrections.

5. Concluding Remarks

In this work, a novel cerebellar-like adaptive Smith predictor was obtained by integrating a computational model of the cerebellum circuit embedded in a recurrent loop with a Smith predictor. The proposed control architecture is biologically inspired by the theory of the cerebellar internal forward models formation for anticipatory motor control. Other authors tried to provide anticipatory corrective responses through learning from motor errors, but they show a mechanism to correct feed-forward motor responses that is tied to the dynamics of the plant and so, they cannot be easily generalized to new robot configurations. These previous works are based on the combination of forward and inverse models without using cerebellar computational algorithms for robot control, such as Wolpert and Kawato,[54] who built a model for sensorimotor learning and control (MOSAIC), with the aim of learning and selecting the most appropriate set of internal models for a given environment. Indeed, a modular approach based on internal forward model pairing would lead to advantages since the controller would be able of providing appropriate motor commands for multiple contexts, tasks and experiences.

This work was also inspired by the modularity of the cerebellar network for implementing multiple microzones, one of each is devoted to control and learn the internal forward model of a single robot

module. Future work will regard the study about how the cerebellum processes the sensory inputs and organizes its internal structures for motor control tasks under changing morphologies and topologies. In this case, more complex nonlinarities could be handled by combining traditional PID approaches with biologically plausible methods that give the first ones the capability of continuously learning by adapting the gains. Some attempts have already been applied in control, such as membrane controllers for trajectory tracking and Spiking Neural P systems for communication strategy among neurons.[55,56] Finally, we argue that the performance could be improved by using advanced techniques for optimization of control parameters.[57,58]

Acknowledgments

References

1. A. J. Bastian, Learning to predict the future: The cerebellum adapts feedforward movement control, *Curr. Opin. Neurobiol.* **16**(6) (2006) 645–649, Motor systems/Neurobiology of behavior.
2. E. D'Angelo, A. Antonietti, S. Casali, C. Casellato, J. A. Garrido, N. R. Luque, L. Mapelli, S. Masoli, A. Pedrocchi, F. Prestori, M. F. Rizza and E. Ros, Modeling the cerebellar microcircuit: New strategies for a long-standing issue, *Front. Cell. Neurosci.* **10** (2016) 176.
3. M. Manto, J. Bower, A. Conforto, J. Delgado-García, S. Da Guarda, M. Gerwig, C. Habas, N. Hagura, R. Ivry, P. Marien, M. Molinari, E. Naito, D. Nowak, N. Ben Taib, D. Pelisson, C. Tesche, C. Tilikete and D. Timmann, Consensus paper: Roles of the cerebellum in motor control-the diversity of ideas on cerebellar involvement in movement, *Cerebellum* **11**(6) (2012) 457–487.
4. H. Lalazar and E. Vaadia, Neural basis of sensorimotor learning: Modifying internal models, *Current Opinion Neurobiol.* **18**(6) (2008) 573–581.
5. T. J. Ebner, *Cerebellum and Internal Models*, *Handbook of the Cerebellum and Cerebellar Disorders*, eds. M. Manto, J. D. Schmahmann, F. Rossi, D. L. Gruol and N. Koibuchi (Springer Netherlands, Dordrecht, 2013), pp. 1279–1295.
6. T. Ishikawa, S. Tomatsu, J. Izawa and S. Kakei, The cerebro-cerebellum: Could it be loci of forward models?, *Neurosci. Res.* **104** (2016) 72–79.
7. D. M. Wolpert and Z. Ghahramani, Computational principles of movement neuroscience, *Nat Neurosci* **3** (2000) 1212–1217.
8. A. Giovannucci, A. Badura, B. Deverett, F. Najafi, T. Pereira, Z. Gao, I. Ozden, A. Kloth, E. Pnevmatikakis, P. L, C. De Zeeuw, J. Medina and S. Wang, Cerebellar granule cells acquire a widespread predictive feedback signal during motor learning, *Nat. Neurosci.* **20**(5) (2017) 727–734.
9. M. Ito, *The Cerebellum*: *Brain for an Implicit Self* (FT Press, Upper Saddle River, N.J., 2012).
10. R. C. Miall, D. J. Weir, D. Wolpert and J. F. Stein, Is the cerebellum a smith predictor?, *J. Motor behav.* **25**(3) (1993) 203–216.
11. O. Smith, *Feedback Control Systems* McGraw-Hill series in Control Systems Engineering (McGraw-Hill, 1958).
12. J. Porrill, P. Dean and S. R. Anderson, Adaptive filters and internal models: Multilevel description of cerebellar function, *Neural Netw.* **47** (2013) 134–149.
13. S. Tolu, M. Vanegas, J. A. Garrido, N. R. Luque and E. Ros, Adaptive and predictive control of a simulated robot arm, *Int. J. Neural Syst.* **23**(3) (2013) 1350010.
14. M. Pacheco, R. Fogh, H. Lund and D. Christensen, *Fable II: Design of a Modular Robot for Creative Learning*, Proc. 2015 IEEE Int. Conf. Robotics and Automation. (IEEE, 2015), pp. 6134–6139.
15. D. Marr, A theory of cerebellar cortex, *J. Physiol.* **202**(2) (1969) 437–470.
16. J. S. Albus, A theory of cerebellar function, *Math. Biosci.* **10**(1) (1971) 25–61.
17. M. Fujita, Adaptive filter model of the cerebellum, *Biol. Cybernet.* **45** (1982) 195–206.
18. B. Siciliano and O. Khatib, *Springer Handbook of Robotics*, 2nd edn. (Springer Publishing Company, Incorporated, 2016).
19. J. A. Garrido Alcazar, N. R. Luque, E. D'Angelo and E. Ros, Distributed cerebellar plasticity implements adaptable gain control in a manipulation task: A closed-loop robotic simulation, *Front. Neural Circuits* **7** (2013) 159.
20. N. R. Luque, J. A. Garrido, R. R. Carrillo, S. Tolu and E. Ros, Adaptive cerebellar spiking model embedded in the control loop: Context switching and robustness against noise, *Int. J. Neural Syst.* **21**(5) (2011) 385–401.
21. S. Tolu, M. Vanegas, N. R. Luque, J. A. Garrido and E. Ros, Bio-inspired adaptive feedback error learning architecture for motor control, *Biolog. Cybern.* **106**(8–9) (2012) 507–522.
22. D. Wolpert, R. Miall and M. Kawato, Internal models in the cerebellum, *Trends Cogn. Sci.* **2**(9) (1998) 338–347.

23. H. Imamizu, T. Kuroda, T. Yoshioka and M. Kawato, Functional magnetic resonance imaging examination of two modular architectures for switching multiple internal models, *J. Neurosci.* **24** (2004) 1173–1181.

24. L. Lonini, L. Dipietro, L. Zollo, E. Guglielmelli and H. I. Krebs, An internal model for acquisition and retention of motor learning during arm reaching, *Neural Comput.* **21**(7) (2009) 2009–2027.

25. H. Imamizu and M. Kawato, Cerebellar internal models: Implications for the dexterous use of tools, *Cerebellum* **11**(2) (2012) 325–335.

26. M. Antonelli, A. J. Duran, E. Chinellato and A. P. del Pobil, Learning the visualoculomotor transformation: Effects on saccade control and space representation, *Robot. Auton. Syst.* **71** (2015) 13–22.

27. T. Shibata and S. Schaal, Biomimetic gaze stabilization based on feedback-error-learning with nonparametric regression networks, *Neural Netw.* **14**(2) (2001) 201–216.

28. L. Vannucci, E. Falotico, S. Tolu, P. Dario, H. H. Lund and C. Laschi, Eye-head stabilization mechanism for a humanoid robot tested on human inertial data, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **9793** (2016) 341–352.

29. L. Vannucci, E. Falotico, S. Tolu, V. Cacucciolo, P. Dario, H. Hautop Lund and C. Laschi, A comprehensive gaze stabilization controller based on cerebellar internal models, *Bioinspiration Biomimetics* **12**(6) (2017) 1–15.

30. I. B. Ojeda, S. Tolu and H. H. Lund, A scalable neuro-inspired robot controller integrating a machine learning algorithm and a spiking cerebellar-like network, *Biomimetic and Biohybrid Systems - 6th Int. Conf., Living Machines 2017, Proc. Stanford, CA, USA, July 26-28, 2017* (2017), pp. 375–386.

31. P. J. Porrill and Dean, Recurrent cerebellar loops simplify adaptive control of redundant and nonlinear motor systems, *Neural Comput.* **19**(1) (2007) 170–193.

32. A. Lenz, T. Balakrishnan, A. G. Pipe and C. Melhuish, An adaptive gaze stabilization controller inspired by the vestibulo-ocular reflex, *Bioinspiration biomimetics* **3**(3) (2008) 035001.

33. E. Franchi, E. Falotico, D. Zambrano, G. Muscolo, L. Marazzato, P. Dario and C. Laschi, A comparison between two bio-inspired adaptive models of vestibulo-ocular reflex (VOR) implemented on the iCub robot, *2010 10th IEEE-RAS Int. Conf. Humanoid Robots, Humanoids 2010* (Nashville, TN, USA, 2010), pp. 251–256.

34. N. R. Luque, R. R. Carrillo, F. Naveros, J. A. Garrido and M. Sez-Lara, Integrated neural and robotic simulations. simulation of cerebellar neurobiological substrate for an object-oriented dynamic model abstraction process, *Robot. Auton. Syst.* **62**(12) (2014) 1702–1716.

35. M. C. Capolei, E. Angelidis, E. Falotico, H. Hautop Lund and S. Tolu, A biomimetic control method increases the adaptability of a humanoid robot acting in a dynamic environment, *Front. Neurorobotics.* **13**(70) (2019) 1–15.

36. A. Antonietti, C. Casellato, E. D'Angelo and A. Pedrocchi, Model-driven analysis of eyeblink classical conditioning reveals the underlying structure of cerebellar plasticity and neuronal activity, *IEEE Trans. Neural Netw. Learn. Syst.* **28** (2017) 2748–2762.

37. A. Geminiani, C. Casellato, A. Antonietti, E. D'Angelo and A. Pedrocchi, A multiple-plasticity spiking neural network embedded in a closed-loop control system to model cerebellar pathologies, *Int. J. Neural Syst.* **28**(05) (2018) 1750017.

38. L. Vannucci, S. Tolu, E. Falotico, P. Dario, H. Hautop Lund and C. Laschi, Adaptive gaze stabilization through cerebellar internal models in a humanoid robot, *Proc. IEEE RAS and EMBS Int. Conf. Biomedical Robotics and Biomechatronics* (Singapore, 2016), pp. 25–30.

39. R. R. Carrillo, E. Ros, C. Boucheny and O. J.-M. Coenen, A real-time spiking cerebellum model for learning robot control, *Biosystems* **94**(1) (2008) 18–27.

40. C. Casellato, A. Antonietti, J. Garrido, R. R Carrillo, N. R. Luque, E. Ros, A. Pedrocchi and E. D'Angelo, Adaptive robotic control driven by a versatile spiking cerebellar network, *PloS One* **9**(11) (2014) e112265.

41. M. Kawato, Cerebellum: Models, *Encyclopedia of Neuroscience*, ed. in LR Squire, (2) (Elsevier Ltd., Oxford: Academic Press, 2009), pp. 757–767.

42. M. Ito, Control of mental activities by internal models in the cerebellum, *Nat. Rev. Neurosci.* **9**(4) (2008) 304–313.

43. R. Miall and D. Wolpert, Forward models for physiological motor control, *Neural Netw.* **9**(8) (1996) 1265–1279.

44. P. Dean, J. Porrill, C.-F. Ekerot and H. Jorntell, The cerebellar microcircuit as an adaptive filter: Experimental and computational evidence, *Nat. Rev. Neurosci.* **11** (2010) 30–43.

45. S. Schaal and N. Schweighofer, Computational motor control in humans and robots, *Curr. Opinion Neurobiol.* **15**(6) (2005) 675–682.

46. H. Imamizu, T. Kuroda, S. Miyauchi, T. Yoshioka and M. Kawato, Modular organization of internal models of tools in the human cerebellum, *Proc. Nat. Acad. Sci.* **100**(9) (2003) 5461–5466.

47. S. Vijayakumar and S. Schaal, Locally weighted projection regression: An o(n) algorithm for incremental real time learning in high dimensional space, in *Proc. Seventeenth Int. Conf. Machine Learning (ICML 2000)* (San Francisco, USA, 2000), pp. 1079–1086.

48. N. R. Luque, J. A. Garrido, R. R. Carrillo, E. D'Angelo and E. Ros, Fast convergence of learning requires plasticity between inferior olive and deep cerebellar nuclei in a manipulation task: A closed-loop robotic simulation, *Front. Comput. Neurosci.* **8**(97) (2014) 1–16.

49. R. Apps and M. Garwicz, Anatomical and physiological foundations of cerebellar information processing, *Nat. Rev. Neurosci.* **6**(4) (2005) 297–311.

50. M. Ito, Cerebellar circuitry as a neuronal machine, *Progress in Neurobiology* **78**(35) (2006) 272–303.

51. H. Gomi and M. Kawato, Adaptive feedback control models of the vestibulocerebellum and spinocerebellum, *Biol. Cybernet.* **68**(2) (1992) 105–114.

52. I. Herreros, X. D. Arsiwalla and P. Verschure, A forward model at purkinje cell synapses facilitates cerebellar anticipatory motor control, *Conf. Neutral Information Processing System 2016* (Barcelona, Spain, 2016).

53. D. Nguyen-Tuong, M. Seeger and J. Peters, Model learning with local gaussian process regression, *Adv. Robot.* **23**(15) (2009) 2015–2034.

54. D. M. Wolpert and M. Kawato, Multiple paired forward and inverse models for motor control, *Neural Netw.* **11**(7–8) (1998) 1317–1329.

55. X. Wang, G. Zhang, F. Neri, T. Jiang, J. Zhao, M. Gheorghe, F. Ipate and R. Lefticaru, Design and implementation of membrane controllers for trajectory tracking of nonholonomic wheeled mobile robots, *Integrated Comput. Aided Eng.* **23**(1) (2015) 15–30.

56. L. Pan, G. Pun, G. Zhang and F. Neri, Spiking neural p systems with communication on request, *Int. J. Neural Syst.* **27**(8) (2017) 1750042.

57. A. Antonietti, D. Martina, C. Casellato, E. DAngelo and A. Pedrocchi, Control of a humanoid nao robot by an adaptive bioinspired cerebellar module in 3d motion tasks, *Comput. Int. neurosci.* **2019** (2019) 1–15.

58. E. Massi, L. Vannucci, U. Albanese, M. C. Capolei, A. Vandesompele, G. Urbain, A. M. Sabatini, J. Dambre, C. Laschi, S. Tolu and E. Falotico, Combining evolutionary and adaptive control strategies for quadruped robotic locomotion, *Front. Neurorobotics.* **13**(71) (2019) 1–19.