Cerebellar-Inspired Learning Rule for Gain Adaptation of Feedback Controllers

Ivan Herreros[1], Xerxes D. Arsiwalla[1], Cosimo Della Santina[2],
Jordi-Ysard Puigbo[1], Antonio Bicchi[2,3], Paul Verschure[1,4]

***Abstract*— The cerebellum is a crucial brain structure in enabling precise motor control in animals. Recent advances suggest that the timing of the plasticity rule of Purkinje cells, the main cells of the cerebellum, is matched to behavioral function. Simultaneously, counter-factual predictive control (CFPC), a cerebellar-based control scheme, has shown that the optimal rule for learning feed-forward action in an adaptive filter playing the role of the cerebellum must include a forward model of the system controlled. Here we show how the same learning rule obtained in CFPC, which we term as Model-enhanced least mean squares (ME-LMS), emerges in the problem of learning the gains of a feedback controller. To that end, we frame a model-reference adaptive control (MRAC) problem and derive an adaptive control scheme treating the gains of a feedback controller as if they were the weights of an adaptive linear unit. Our results demonstrate that the approach of controlling plasticity with a forward model of the subsystem controlled can provide a solution to a wide set of adaptive control problems.**

I. Introduction

The cerebellum is arguably the brain structure whose study has had a deeper impact on the robotics and control communities. The seminal theory of cerebellar function by Marr [15] and Albus [1] was translated by the latter into the cerebellar model articulation controller (CMAC) in the early seventies [2], which up until today is used both in research and applications. A decade later, Fujita [8] advanced the adaptive filter theory of cerebellar function based on the work by Widrow et al. [20]. Later, in the late eighties, Kawato formulated the influential feedback error learning (FEL) model of cerebellar function [13], in which the cerebellum, implemented as an adaptive filter, learned from, and supplemented, a feedback controller. Unlike CMAC, FEL had a strong impact within the neuroscientific community as a theory of biological motor control [22]. Within the robotics and control communities, FEL has been studied in terms of performance and convergence properties [16]. Later, the adaptive filter theory of cerebellar function was revived by Porrill and colleagues [5], proposing alternatives to FEL that have been applied to the control of bio-mimetic actuators, like pneumatic or elastomer muscles [14], [21].

[1] SPECS, University Pompeu Fabra, Carrer de Roc Boronat, 138, 08018 Barcelona, Spain

[2]Research Center “Enrico Piaggio”, University of Pisa, Largo Lucio Lazzarino 1, 56126 Pisa, Italy

[3]Department of Advanced Robotics, Istituto Italiano di Tecnologia, via Morego, 30, 16163 Genoa, Italy

[4] Instituci Catalana de Recerca i Estudis Avanats, Passeig de Llus Companys, 23, 08010 Barcelona, Spain

ivanherreros@gmail.com

More recently, the counterfactual predictive control (CFPC) scheme was proposed in [10], motivated from neuro-anatomy and physiology of eye-blink conditioning, a behavior dependent on the cerebellum. CFPC includes a reactive controller, which is an output-error feedback controller that models brain stem or spinal reflexes actuating on peripheral muscles, and a feed-forward adaptive component that models the cerebellum and learns to associate its own inputs with the errors that drive the reactive controller. CFPC proposes that the learning of adaptive terms in the linear filter should depend on the coincidence of an error signal with the output of a forward model implemented at the synaptic level, reproducing the dynamics of the downstream reactive closed-loop system. We refer to that learning rule as a model-enhanced least-mean squares (ME-LMS) rule to differentiate with the basic least-mean squares (LMS) rule that is implemented in previous models of the cerebellum, such as CMAC and FEL. In agreement with the theoretical insight of CFPC, recent physiological evidence in [19] shown that the timing of the plasticity rule of Purkinje cells is matched to behavioral function. That suggests that Purkinje cells, the main cells implied in learning at the level of the cerebellum, have plasticity rules that reflect the sensorimotor latencies and dynamics of the plants they control.

However, in the context of CFPC, the ME-LMS rule was derived as a batch gradient-descent rule for solving a feed-forward control task in discrete time. In that sense, it can be interpreted as providing a solution to a iterative-learning control scheme, an input design technique for learning to optimize the execution of a repetitive task [4]. Hence, it remained open the question as to whether a similar learning rule could support the acquisition of well-tuned feedback gains. That is, whether ME-LMS could be applied in an adaptive *feedback* control problem. Here we answer that using the model reference adaptive control (MRAC) frame [3]. In that frame, we first show that the biologically-inspired ME-LMS algorithm can be derived from first principles. More concretely, we show that the ME-LMS rule emerges from deriving the stochastic gradient descent rule for the general problem of updating the gains of linear proportional feedback controllers actuating on a LTI system. Finally we test in simulation the effectiveness of the proposed cerebellum-inspired architecture in controlling a damped-spring mass system, a non-minimum phase plant and, finally, closing the loop with the biology, a biologically-based model of a human limb.

II. DERIVATION OF THE ME-LMS LEARNING RULE

In the next we derive a learning rule for learning the controller gains for both state and output-error feedback controllers. The generic architectures for a full-state-feedback and a proportional (P) controller are shown in Fig. 1. To define the model-reference adaptive control (MRAC) problem, we set a reference model whose output we denote by r_{rm}. The error in following the output of the reference model, $e_{rm} = r_{rm} - y$, drives adaptation of the feedback gains. But note that in the proportional error feedback controller, $e = r - y$ is the signal feeding the error feedback gain.

A. ME-LMS for state-feedback

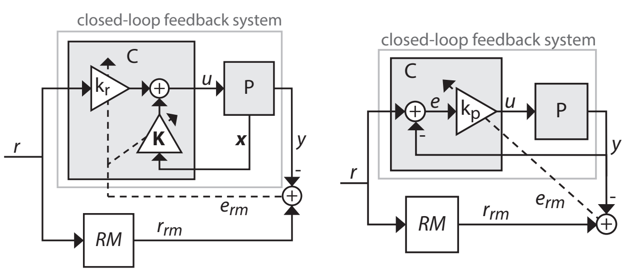

Fig. 1. Adaptive architecture for the full state feedback (*left*) and output-error proportional (P, *right*) control cases. Abbreviations: C, feedback controller, P, plant; RM, reference model; r, reference signal; r, reference signal outputted by the reference model; y, plant's output; e, output error; e_{rm}, output error relative to the reference model; $\mathbf{x}$, state of the plant; u, control signal; k_r, reference gain; $\mathbf{K}$, state feedback gains; and k_p, proportional error-feedback gain.

For the derivation purposes we assume that the adaptive control strategy is applied to a generic LTI dynamical system

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u \\ y &= \mathbf{C}\mathbf{x} \end{aligned} \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{N\times N}$, $\mathbf{B} \in \mathbb{R}^{N\times 1}$ and $\mathbf{C} \in \mathbb{R}^{1\times N}$ are the usual dynamics, input and output matrices, respectively; $\mathbf{x} \in \mathbb{R}^N$ is the state vector; and u and y, both scalars, are the input and output signals.

The control signal will be generated according to the following state-feedback law

$$u = \mathbf{K}\mathbf{x} + k_r r \tag{2}$$

where r is the reference signal, $\mathbf{K} \in \mathbb{R}^{1\times N}$ is the (row) vector of state feedback gains and k_r the reference gain. Both $\mathbf{K}$ and k_r are here time-dependent and will be updated by the learning rule controlling adaptation.

Substituting the control law within the dynamics equation, we obtain the closed-loop system description

$$\dot{\mathbf{x}} = (\mathbf{A} + \mathbf{B}\mathbf{K})\mathbf{x} + \mathbf{B}k_r r \tag{3}$$

$$y = \mathbf{C}\mathbf{x} \tag{4}$$

We set the L-2 norm of the error as the cost function to minimize

$$J = \frac{1}{2}e_{rm}^2$$

For convenience, we write now the control law as $u = \tilde{\mathbf{K}}^\mathsf{T}\tilde{\mathbf{x}}$ with $\tilde{\mathbf{K}} \in \mathbb{R}^{N+1} \equiv [k_1, \ldots, k_N, k_r]^\mathsf{T}$ and $\tilde{\mathbf{x}} \in \mathbb{R}^{N+1} \equiv [x_1, \ldots, x_N, r]^\mathsf{T}$. To derive the gradient descent algorithm for adjusting the vector of gains, $\tilde{\mathbf{K}}$, we need the gradient of J withe respect to $\tilde{\mathbf{K}}$:

$$\nabla_{\tilde{\mathbf{K}}} J = \frac{\partial e_{rm}}{\partial \tilde{\mathbf{K}}} e_{rm} = -\frac{\partial y}{\partial \tilde{\mathbf{K}}} e_{rm} \tag{5}$$

Now we will consider each of the gains individually, treating separately the state and the reference gains. Let k_i denote the feedback gain associated with the i-th state variable. We have that

$$\frac{\partial y}{\partial k_i} = \mathbf{C}\frac{\partial \mathbf{x}}{\partial k_i} \tag{6}$$

We compute the partial derivative of the state vector $\mathbf{x}$ with respect to k_i applying the partial derivative to the differential equation that governs the closed-loop dynamics:

$$\frac{\partial}{\partial k_i}\dot{\mathbf{x}} = \frac{\partial}{\partial k_i}\left((\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x} + \mathbf{B}k_r r\right) \tag{7}$$

Using the substitution $\mathbf{z}_i \equiv \frac{\partial \mathbf{x}}{\partial k_i}$ and applying the product rule in the derivation we obtain

$$\dot{\mathbf{z}}_i = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{z}_i + \mathbf{B}x_i \tag{8}$$

Introducing $h_i \equiv \mathbf{C}\mathbf{z}_i$, we get

$$\frac{\partial J}{\partial k_i} = h_i e_{rm}$$

Note that this has solved the problem of obtaining the partial derivative for all state feedback gains.

In the case of the reference gain, with $\mathbf{z}_r \equiv \frac{\partial \mathbf{x}}{\partial k_r}$, we obtain

$$\dot{\mathbf{z}}_r = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{z}_r + \mathbf{B}r \tag{9}$$

And introducing $h_r \equiv \mathbf{C}\mathbf{z}_r$,

$$\frac{\partial J}{\partial k_r} = h_r e$$

We will refer to the quantities h_i and h_r as eligibility traces. We can write the vector of eligibility traces as follows: $\tilde{\mathbf{h}} = [h_1, \ldots, h_N, h_r]^\mathsf{T}$.

With this we can solve for the gradient of the cost function as follows

$$\nabla_{\tilde{\mathbf{K}}} J = -\tilde{\mathbf{h}} e_{rm}$$

And consequently derive a learning rule for the gains that will follow a gradient descent:

$$\dot{\tilde{\mathbf{K}}} = \eta \tilde{\mathbf{h}} e_{rm} \tag{10}$$

Note that this rule is similar to the classical Widrow-Hoff or least mean squares (LMS) rule. However, in the standard LMS, the rate of change is obtained multiplying the error with the input signals of the filter ($\dot{\tilde{\mathbf{K}}} = \eta \tilde{\mathbf{x}} e_{rm}$) whereas in the rule we have derived the error multiplies the quantities in $\tilde{\mathbf{h}}$, which are obtained after passing the input signals through a forward model of the controlled system. For this reason, we

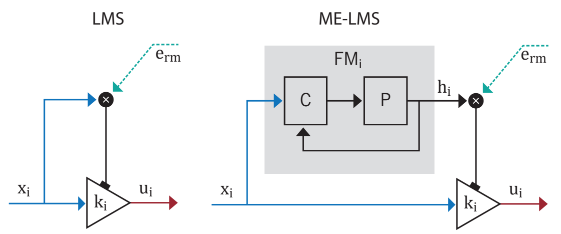

Fig. 2. Schematic of the implementation of the LMS (left) and the ME-LMS (right) rule at the level of a single adaptive weight. Note that since k_i is a gain of the controller C, this scheme is implicitly recursive: as the k_i gain changes (together with the other adaptive gains of the controller) the forward that it utilizes to drive plasticity, changes as well.

refer to the learning rule in equation 10 as model-enhanced least mean squares (ME-LMS). Moreover, $\tilde{h}_i$ is the eligibility trace associated to the input $\tilde{x}_i$ because, at the time that a particular error signal comes, it codes how much $\tilde{x}_i$ could have contributed to canceling that error.

B. *ME-LMS for output Error Proportional Control*

The control law of an output-error proportional controller is $u = k_p e = r - y = r - \mathbf{C}\mathbf{x}$. The corresponding closed-loop system becomes

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}\mathbf{C}k_p)\mathbf{x} + \mathbf{B}k_p r \quad (11)$$

$$y = \mathbf{C}\mathbf{x} \quad (12)$$

Following a derivation analogous to the previous one, we obtain the following expression for computing eligibility trace of the proportional gain (h_p).

$$\dot{\mathbf{z}}_p = (\mathbf{A} - \mathbf{B}\mathbf{C}k_p)\mathbf{z}_p + \mathbf{B}r \quad (13)$$

$$h_p = \mathbf{C}\mathbf{z}_p \quad (14)$$

hence, in this model, plasticity will be implemented by the rule $\dot{k}_p = \eta h_p e$.

C. *Model-enhanced least mean-squares vs. least mean squares rule*

The differences between LMS and ME-LMS care summarized as follows: In LMS, the change of the adaptive gain k_i is determined based on the temporal coincidence of a global error signal e_{rm} and the local input x_i (Fig. 2 *left*). In ME-LMS (Fig. 2 *right*) the change in the gain is controlled by the output of a gain-specific forward model FM_i, whose output h_i facilitates an eligibility trace for k_i. The term *eligibility trace* implies that h_i marks how much the input x_i could have contributed to decrease the current error. In that sense, it is a *trace* as long as to be generated h_i takes into account the history of x_i with a time-span implicitly determined by the dynamics of the forward model.

III. Applying ME-LMS to a linear damped spring-mass system

We evaluate here the performance of the proposed algorithm in controlling a standard damped-spring mass system: $m\ddot{q} + c\dot{q} + kq = u$, with $m = 1\text{Kg}$, $c = 0.5\,\frac{\text{Ns}}{\text{m}}$, $k = 0.5\,\frac{\text{N}}{\text{m}}$.

A. *Output error P-control*

For this problem we use a reference model built as chain of two leaky integrators with identical relaxation time constants ($\tau = \sqrt{0.5}s$). The impulse response curve of this reference model corresponds to a normalized double-exponential convolution that peaks at $0.5s$. Finally, we use as reference the following superposition two sinusoidal functions $r = \sin(5t/7) + \sin(5t/11)$. We first examine the cost function as a function of the feedback parameter, k_p varying it logarithmically from 0.01 to 100.0. Within that range, the cost function is convex and has minimum in the near $k_p = 0.815$ (Fig 3 *above-left*). At this point we check whether the ME-LMS converges to the optimum value for k_p. For comparison we also run the test with using the standard LMS rule, and a heuristically motivated alternative algorithm wherein we use a model of the open-loop system to generate the eligibility trace. We test two starting values each one at a different side of the minimum and we observe that in both cases the ME-LMS converges around the optimal k_p (Fig 3 *above-right*) while the alternative algorithms convergence to different equilibrium points which are non-optimal in cost terms. The difference in performance can also be appreciated by seeing how the different algorithms track r_{rm} at the end of the training period ($1h$ of simulated time) (Fig 3 *below-left*). Indeed, only the ME-LMS algorithm is in-phase with r_{rm}. Finally, in cost terms, only ME-LMS converges rapidly to the minimum value (Fig 3 *above-right*).

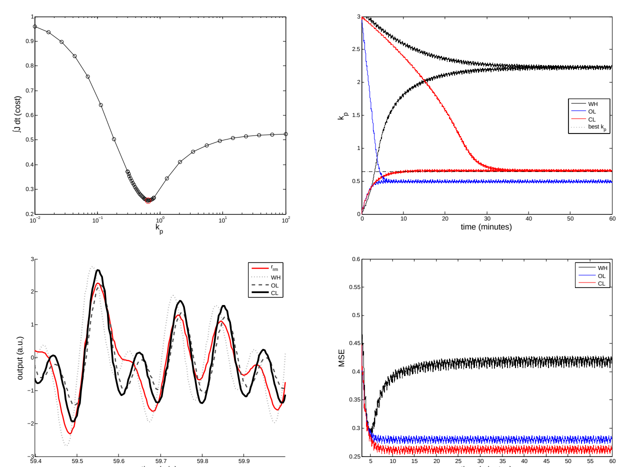

Fig. 3. Damped-spring mass system with P-control. *Above left*. Cost as a function of k_p. *Above right*. Convergence of k_p for three different models. *Below left*. Output trajectories and reference signals. *Below right*. Evolution of the error in mean-square error (MSE) units. *In all panels*: WH: standard LMS rule; OL: LMS with eligibility trace derived from the open-loop system; CL: ME-LMS with eligibility trace derived from the closed-loop system.

In summary, this result shows that even for the simplest feedback learning scenario, a LMS-like learning rule converges to the optimal gain only if it uses the eligibility trace generated by a forward model that reproduces the dynamics of the closed-loop system.

B. Full state feedback control

Here we use the ME-LMS algorithm in a full state feedback controller, keeping the same reference model and reference signal as in the previous example. This problem is harder in that now there are three parameters to adjust; the two feedback and a feed-forward gain but, in contrast to the previous example, here it is guaranteed that there exist optimal gains allowing the closed-loop system to exactly match the reference model.

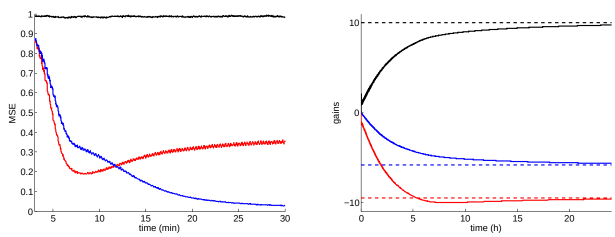

Fig. 4. Damped-spring mass system with FSF-control. *Left*: evolution of the error for the three algorithms. The color code is an in Fig. 3 *Right* Evolution of the two feedback gains and the reference gain for the ME-LMS.

In contrast with the previous examples where all the three algorithms succeeded in controlling the plant to some extent, in this case the pure LMS algorithm fails completely to improve performance (Fig. 4 *Left*). In contrast both algorithms enhanced with a model of either the open or the closed-loop system rapidly reduce error even though, only ME-LMS is still decreasing the error after 30 minutes of simulated time (Fig. 4 *Left*). Indeed, running a total of 24 hours of simulated time with the ME-LMS model, we observe that the adaptive gains converge the optimal ones (Fig. 4 *Right*).

IV. Applying ME-LMS to a non-minimum phase plant

A. Full state feedback control

In this section we apply ME-LMS to a non-minimum phase system, which is a system with zeros in the right-hand side of the complex plane. Acting to reduce an error in such a system requires foresight in that before minimizing an error one has to steer the plant in the direction that increases it. That property of the system, namely that errors cannot be reduced instantaneously, disqualifies the use of the standard LMS algorithm for the problem of adaptively tuning feedback gains. On the contrary, ME-LMS, as it takes explicitly into account the dynamics of the controlled system to control plasticity, can in principle appropriately adjust the gains of a feedback controller even when it is applied to a non-minimum phase system.

As a non-minimum phase system, we use the following a linearized model of a balance system (e.g., a self-balancing robot):

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 \\ m^2 l^2 g/\mu & -cJ_t/\mu & -\gamma lm/\mu \\ M_t mgl/\mu & -clm/\mu & -\gamma M_t/\mu \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ J_t/\mu \\ lm/\mu \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

where $\mu = M_t J_t - m^2 l^2$. The values, chosen to mimic the ones of a custom made prototype, are $M_t = 1.58Kg$, $m = 1.54Kg$, $l = 0.035$, $J_t = 1.98 \times 10^{-3}$, $\gamma = 0.01$ and $c = 0.1$. As an added difficulty, the plant is unstable in open loop. To deal with that, we set boundary conditions to our simulation. That is, whenever the system reaches a threshold velocity of $0.5m/s$ or and inclination in absolute value above 22.5 degrees the simulation re-starts and the system is brought back to the initial rest position. In that sense, the system is not fully autonomous but *assisted* by an external agent that makes it regain the balanced position.

In practice, the control problem consisted in following a low amplitude and slow velocity reference signal constructed as a sum of sinusoidals $0.05(\sin(\pi t/70) + \sin(\pi t/110))$. We used the same reference model as in the previous section.

For this system the problem of adjusting the velocity to a particular reference signal is under-constrained as there are two possible strategies: keeping the error in the linear velocity equal to zero while the angular position diverges or keeping that error equal to zero while maintaining the angular position stable. In order to bias the system towards the second solution we set the starting gains already tuned towards the right solution. However, that initial set of gains keep the robot balanced for less than $200ms$. Hence, we can divide this particular problem in two stages: first stabilizing the plant, and next make the controlled system converge to the dynamics of the reference model.

ME-LMS requires approximately 10 seconds for reaching a set of gains that stabilizes the plant following fifteen falls (Fig. 5 *top row*). Standard LMS fails as it converges to a solution that controls for the linear velocity but ignores the angular position (Fig. 5 *middle row*). Indeed, by the end of the 30 seconds of training, standard LMS has reduced the errors in velocity but the speed at which the plant loses balance remains unchanged. Regarding the learning dynamics, we observe how the feedback gains of the ME-LMS change rapidly until the robot maintains balance (Fig. 5 *below left*). After that, the change is gradual but sufficient to achieve following closely the target velocity by the end of the 10 mins training (Fig. 5 *below right*).

V. Discussion

The cerebellum is a crucial brain structure for accurate motor control. It is phylogenetically old and conserved through evolution in all vertebrates [9]. Much of the interest that the cerebellum gathered in the machine learning, robotics and adaptive control communities stemmed from its remarkable anatomy [7], with a general connectivity layout that resembles very closely the one of a supervised learning neural

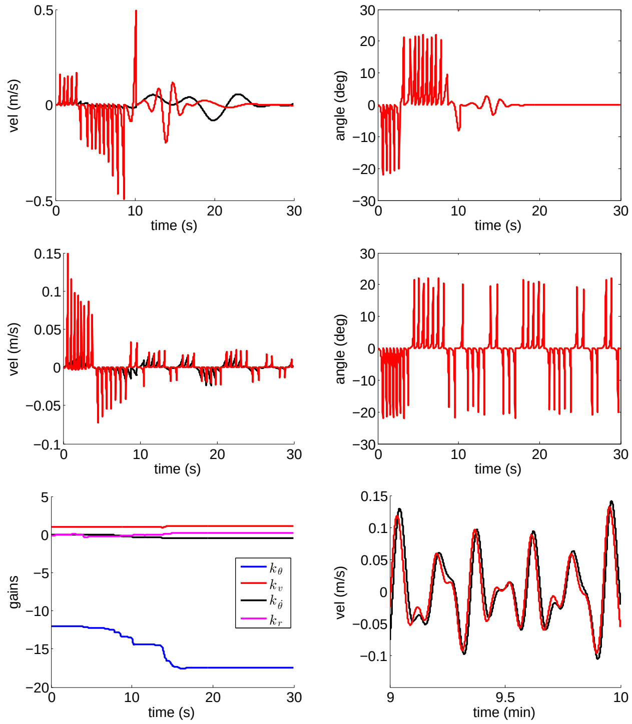

Fig. 5. ME-LMS applied to a Self-Balancing system. Top row refers to the ME-LMS algorithm and middle row to the standard LMS. Left panels show the velocity traces (red) and the reference signal (black). Right panels, show the angular position traces. Bottom left: gains of the ME-LMS system during the first 30 seconds of simulation. Bottom right: velocity trace the last minute of 10 mins training of the ME-LMS system.

network, such as a Perceptron [17]. However, here we have drawn inspiration for a recent insight regarding cerebellar physiology that has emerged simultaneously at both the theoretical [10] and experimental [19] domains. That is, that in order to solve appropriately the problem of motor control, neurons from a same type (i.e., the cerebellar Purkinje cells) might display different learning rules in different areas of the cerebellum, matched to the particular behavioral needs [6]. From a control theory perspective those behavioral needs correspond to the transport latencies and response dynamics associated to the controlled sub-system.

Here we have shown that the model-enhanced least-mean-squares (ME-LMS) learning rule can be easily derived for the task of learning the optimal feedback gains of a fully known plant. Second, we have shown hat the ME-LMS learning rule converges in a series of tasks in which conventional LMS would fail, as is the case of a non-minimum phase plant.

Regarding the derivation of ME-LMS presented here, it is worth noting that although a similar result was originally obtained in [10] using a cerebellar-based control architecture, the derivation presented here applies to two very general control architectures, namely proportional full-state-feedback and proportional error-feedback control. Hence, in that sense, the current derivation is cerebellar-independent.

Here we have revisited the LMS learning rule in the context of the control of dynamical systems. In this context, given the temporal correlation between control and output signals introduced by the dynamics of the plant, a learning algorithm has to deal with delayed error feedback information [18], [11]. That is, the effects of an action are spread in time (according the impulse response curve) and conversely, a particular effect (e.g., an error in performance) is caused by the recent history of actions (convolved with the same impulse response). Such a non-locality in time introduces the so-called temporal credit-assignment problem [18]. That is, to avoid that current errors repeat in the future, one has to identify which where the correct past actions that should have been taken. In the machine learning literature, there are two main strategies to deal with that problem: back propagation through time in neural networks [12], and the use of eligibility traces in the reinforcement learning, as for instance, in the temporal-differences learning algorithm [18]. Our results here add to the ones in [10] in suggesting that in learning to control a plant with known dynamics through gradient descent, one can solve the temporal credit assignment problem generating eligibility traces with a forward model of the controlled system. With that, we have translated into model reference adaptive control a bio-inspired machine learning technique.

VI. Conclusion and Future Work

In this work we proposed a control algorithm in which a linear feedback action is combined with an adaptation rule for the control gains inspired from cerebellar learning in animals. The controller is also analytically derived as gradient descent estimation of feedback gains for LTI systems.

We tested the effectiveness of the algorithm in three different simulation scenarios, including the control of a model of human upper limb, closing the loop with the biological inspiration. The algorithm presented better performance w.r.t. the classic LMS rule. In future work we plan to experimentally test it on bio-inspired robotic systems.

Although the algorithm presented in practice a stable behavior, to analytically prove the stability of the closed loop system is a challenging task. This is due not only to the non-linearities introduced by the possibility of adapting control gains (common in the context of adaptive control), but also to the strong interplay between the three dynamics involved: the system, the eligibility trace, and the control gains. However we consider this step very important for the full formalization of the proposed learning rule, and so we depute this study to future works.

Acknowledgments

The research leading to these results has received funding from the European Commissions Horizon 2020 socSMC project (socSMC-641321H2020-FETPROACT-2014) and by the European Research Councils CDAC project (ERC-2013-ADG 341196). And by the European Commission Grant no. H2020-ICT-645599 SOMA: SOft MAnipulation and the ERC Advanced Grant no. 291166 SoftHands.

REFERENCES

[1] James S Albus. A theory of cerebellar function. *Mathematical Biosciences*, 10(1):25–61, 1971.

[2] Js Albus. A new approach to manipulator control: The cerebellar model articulation controller (CMAC). *Journal of Dynamic Systems, Measurement, and Control*, (SEPTEMBER):220–227, 1975.

[3] Karl Johan Åström. Theory and applications of adaptive control??a survey. *Automatica*, 19(5):471–486, 1983.

[4] Douglas A Bristow, Marina Tharayil, and Andrew G Alleyne. A survey of iterative learning control. *IEEE Control Systems*, 26(3):96–114, 2006.

[5] Paul Dean, John Porrill, Carl-Fredrik Ekerot, and Henrik Jörntell. The cerebellar microcircuit as an adaptive filter: experimental and computational evidence. *Nature reviews. Neuroscience*, 11(1):30–43, jan 2010.

[6] Conor Dempsey and Nathaniel B Sawtell. The timing is right for cerebellar learning. *Neuron*, 92(5):931–933, 2016.

[7] J.C. Eccles, M. Ito, and J. Szentágothai. *The cerebellum as a neuronal machine*. Springer Berlin, 1967.

[8] M Fujita. Adaptive filter model of the cerebellum. *Biological cybernetics*, 45(3):195–206, 1982.

[9] Sten Grillner. Control of locomotion in bipeds, tetrapods, and fish. *Comprehensive Physiology*, 2011.

[10] Ivan Herreros, Xerxes Arsiwalla, and Paul Verschure. A forward model at purkinje cell synapses facilitates cerebellar anticipatory control. In *Advances in Neural Information Processing Systems*, pages 3828–3836, 2016.

[11] Michael I Jordan. Computational aspects of motor control and motor learning. In *Handbook of perception and action*, volume 2, pages 71–120. Academic Press, 1996.

[12] Michael I Jordan and David E Rumelhart. Forward models: Supervised learning with a distal teacher. *Cognitive science*, 16(3):307–354, 1992.

[13] M. Kawato, Kazunori Furukawa, and R. Suzuki. A hierarchical neural-network model for control and learning of voluntary movement. *Biological Cybernetics*, 57(3):169–185, 1987.

[14] Alexander Lenz, Sean R Anderson, Anthony G Pipe, Chris Melhuish, Paul Dean, and John Porrill. Cerebellar-inspired adaptive control of a robot eye actuated by pneumatic artificial muscles. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(6):1420–1433, 2009.

[15] D Marr. A theory of cerebellar cortex. *The Journal of physiology*, 202(2):437–470, 1969.

[16] Jun Nakanishi and Stefan Schaal. Feedback error learning and nonlinear adaptive control. *Neural Networks*, 17(10):1453–1465, 2004.

[17] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[18] Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988.

[19] Aparna Suvrathan, Hannah L Payne, and Jennifer L Raymond. Timing rules for synaptic plasticity matched to behavioral function. *Neuron*, 92(5):959–967, 2016.

[20] Bernard Widrow, Marcian E Hoff, et al. Adaptive switching circuits. In *IRE WESCON convention record*, volume 4, pages 96–104. New York, 1960.

[21] Emma D Wilson, Tareq Assaf, Martin J Pearson, Jonathan M Rossiter, Sean R Anderson, John Porrill, and Paul Dean. Cerebellar-inspired algorithm for adaptive control of nonlinear dielectric elastomer-based artificial muscle. *Journal of The Royal Society Interface*, 13(122):20160547, 2016.

[22] Daniel M Wolpert, RC Miall, and Mitsuo Kawato. Internal models in the cerebellum. *Trends in cognitive sciences*, 2(9):338–347, 1998.