Science Robotics
AAAS

Manuscript Template

FRONT MATTER

Title

- A cerebellar based solution to the non-deterministic time delay problem in robotic control
- Cerebellar control solution to time delays

Authors

Ignacio Abadía[1]*†, Francisco Naveros[1, 2]†, Eduardo Ros[1], Richard R. Carrillo[1]‡, Niceto R. Luque[1]*‡

Affiliations

[1]Research Centre for Information and Communication Technologies (CITIC), Dept. Computer Architecture and Technology, University of Granada, Granada, Spain.
[2]Computer School, Dept. of Architecture and Technology of Informatics Systems, Polytechnic University of Madrid, Madrid, Spain.
*Corresponding author: 1 (I.A.); nluque@ugr.es (N.R.L.)
†‡These authors contributed equally to this work.

Abstract

The presence of computation and transmission variable time delays within a robotic control loop is a major cause of instability, hindering safe human-robot interaction (HRI) under these circumstances. Classical control theory has been adapted to counteract the presence of such variable delays; however, the solutions provided to date cannot cope with HRI robotics inherent features. The highly nonlinear dynamics of HRI cobots (robots intended for human interaction in collaborative tasks), together with the growing use of flexible-joints and elastic materials providing passive compliance, prevent traditional control solutions from being applied. Conversely, human motor control natively deals with low power actuators, nonlinear dynamics and variable transmission time delays. Importantly, the cerebellum, pivotal to human motor control, is able to predict motor commands by correlating current and past sensorimotor signals, and to ultimately compensate for the existing sensorimotor human delay (tens of milliseconds). This work aims at bridging those inherent features of cerebellar motor control and current robotic challenges; namely, compliant control in the presence of variable sensorimotor delays. We implement a cerebellar-like spiking neural network (SNN) controller that is adaptive, compliant, and robust to variable sensorimotor delays by replicating the cerebellar mechanisms that embrace the presence of biological delays and allow motor learning and adaptation.

Summary

The biomimetic temporal learning of a cerebellar-like SNN allows compliant cobot control under long non-deterministic latency.

MAIN TEXT

Introduction

The engineering pursuit of the most efficient solutions to emerging challenges has pushed forward the development of technology and the consequent contribution to human progress. Among the latest challenges, human-robot interaction (HRI) has blossomed into a worldwide research discipline helping to lighten human labor (*1*), provide medical assistance (*2*), assist humans in space exploration (*3*), etc. Physical HRI must be safe for both actors, thus requiring compliant and adaptive controllers to operate collaborative robots (cobots). However, HRI can be compromised by contextual variables such as unstructured scenarios, unknown dynamics (*4*), or sensorimotor time delays (*5*). We have recently proven that spiking neural networks (SNNs) can be used for effective robot control providing both accuracy and compliance (*6*), key elements in safe HRI. Yet, the non-deterministic time delay control problem was sidestepped, constituting the focus of this study.

Unintentional time delays in robot control have two main sources: computation and transmission delays. Computation latency represents the time spent in data processing to generate a motor control command (*7*). Transmission latency depends on the communication technology and physical links used between controller and robot. For instance, in telerobotic architectures delays appear in the communication link between the human operator and the robot (*8*); cloud-robotics, a growing field, relies robot control on remote cloud computing resources that lead to computation and transmission latencies within the control loop (*9*); wireless communications carry additional time delays compared to wired connections (*10*). The aforementioned variety of scenarios illustrates the importance of accounting for time delays when designing closed loop robot controllers.

From a classic control perspective, time delays are a major cause of instability in control loops. Traditional controllers dealing with pure delays may cause a phase margin decrease of the robotic system and a higher sensitivity as its static gain increases (*5*). To stabilize time delayed systems, both adapted classic controllers and specifically designed controllers have been proposed (*11*). Under the first category, different proposals try to mitigate the effects of time delays by adapting traditional proportional-integral-derivative (PID) controllers: i) PID stabilization of linear time invariant (LTI) systems using the Hermite–Biehler theorem (*12*); ii) parameter space method to tune the PID coefficients for an LTI system with time delays (*13*); iii) using the Nyquist criterion to compute a set of PID controllers to stabilize a given n-order LTI system with time delay (*14*). Unfortunately, these families of methods cannot be easily applied to HRI cobots whose dynamics are strongly non-linear due to soft or elastic components (*4*). Regarding the second category, it includes the dead-time compensators (DTC) (*15*), a family of controllers specifically designed for systems with time delays: (i) Smith predictor based controllers (*16, 17*), only applicable when delays are constant (*11*); (ii) the finite spectrum assignment approach (*18, 19*). However, DTC solutions strong dependence on the accuracy of the system model (*11*) makes them non-reliable for HRI control, as the growing use of flexible-joints and elastic materials (*20, 21*) makes intractable the mathematical modeling of cobots non-linear dynamics (*22*).

These solutions prove the effort devoted to compensate for time delays in control systems. Here, we enlarge the family of solutions by taking inspiration from millions of years of biological evolution by which nature has arrived at an adaptive solution to perform motor control under variable delays; i.e., predictive control to deal with the sensorimotor pathway delays inherent to the central nervous system (CNS), in charge of human body motor control (*23*, *24*). In the cerebellar sensorimotor pathway exists a variable delay accounting for the time spent since a motor command is generated and propagated to the muscles (efferent delay δ_e) until its effect is sensed back at the cerebellum (afferent delay δ_a). These sensorimotor delays range from 100 to 150 ms approximately, with inter and intra individual variations (*25*). To compensate them, the cerebellum acquires internal representations of the sensorimotor transformations needed to generate the motor commands to achieve a desired movement (*26*), and generates predictive motor commands by a spike-timing-dependent plasticity (STDP) mechanism that correlates present and past sensorimotor signals, thus allowing motor learning even in the presence of sensorimotor delays (*27*).

To replicate human motor control and benefit from the aforementioned CNS inherent features, SNNs constitute the most biologically plausible approach since they model the transfer and processing of information as occurs in their biological counterparts; by means of the precise timing of spikes (*28*) which efficiently embed accurate timing. Thus, our cerebellar-like SNN controller adopts the biological delays and mimics the cerebellar STDP mechanism.

In next sections, we evaluate the performance of our SNN controller under time delays of different nature: steady and non-deterministic delays in both lab-controlled and realistic scenarios (i.e., Wi-Fi and cloud-robotics connections). We demonstrate that, besides compliant cobot control, the biological plausibility of our controller provides robustness against variable time delays in the transmission of sensorial information and motor commands, thus, applying an inherent feature of the CNS to a robotic control challenge.

Results

We placed our cerebellar-like SNN at the core of a robotic feedback control loop (Fig. 1). The SNN served as the torque controller able to operate all six degrees-of-freedom (DOF) of the robot arm acting on a trial-and-error basis. A STDP mechanism at the SNN mediated the trial-and-error torque control process facilitating acquisition of the robot arm dynamics when following a set of goal trajectories. During this learning process, the SNN torque controller received the input sensorial information and generated the subsequent output motor commands at 500 Hz rate; see (*6*) for an in-depth review of the learning process. The input sensorial information consisted of the actual robot state supplied by the robot sensors (position, Q_a, and velocity, $\dot{Q}_a$, per each of the six joints, *j1-j6*), the desired trajectory to be performed by the robot arm (position, Q_d, and velocity, $\dot{Q}_d$, per joint), and a teaching/error signal (ε) per joint obtained comparing the actual robot state to the desired trajectory. These analog input signals were later mapped into neuron activations (spikes) that the SNN torque controller computed to subsequently generate the corresponding neural responses. These spike-based neural responses were then mapped into analog motor commands (torque, τ, per joint) and sent to the robot (see Materials and Methods). After SNN learning stabilization and thereby achievement of the desired trajectory, we induced different transmission delays (δ_T) in the sensorimotor pathway to test whether our SNN inherits the cerebellar natural ability to deal with non-deterministic time delays (*25, 29*). We induced sensorial delays in the robot-to-controller (R2C) direction and motor delays in the controller-to-robot (C2R) direction, together with the intrinsic computation delays (δ_C) inherent to the SNN computation.

Our SNN controller reproduced the main properties of the cerebellar circuit, and consisted of 62040 neurons distributed in five neural layers (Fig. 1C). A population of 240 mossy fibers (MFs) conveyed the sensorimotor inputs onto 600 deep cerebellar nuclei (DCN) and 60000 granular cells (GCs). GCs expanded the coding space of MFs (*30*), and this GCs activity was later projected onto 600 Purkinje cells (PCs) via Parallel fibers (PFs). The synaptic learning mechanism (STDP) at PCs (see Materials and Methods) integrated the afferent signals from PFs (i.e., the axons of GCs), with the teaching/error signal from climbing fibers (CFs), i.e., axons of inferior olive (IO) cells. PCs finally inhibited DCN cells, which also integrated inputs from MFs and CFs to generate the cerebellar motor command controlling arm movement. Please see (*6*, *31*) for an in depth review of the roles of the different cerebellar neural layers.

****Figure 1 about here****

Cerebellar torque control provides learning convergence in the presence of time delays

Transmission delays were first artificially induced between the two ends of the robotic feedback control loop in R2C and C2R directions (Fig. 1A). To do so, a point-to-point Ethernet communication connected both ends (robot and controller), each end accommodating a buffer to hold the sensorimotor messages before being sent to the other end. On the robot side, the buffer held the sensorial information for a time δ_{R2C} before being sent to the controller, whereas at the controller side the buffer held the motor commands for a time δ_{C2R} before being sent to the robot. A total transmission delay of $\delta_T = \delta_{R2C} + \delta_{C2R}$ was induced ($\delta_{R2C} = \delta_{C2R} = \delta_T/2$). A 12 cm radius circular trajectory performed in two different xyz planes along with a sequence of a circular plus a Lissajous trajectory performed in the xy plane were used to verify that the cerebellar control solution was not task-dependent (see Materials and Methods for trajectory description). Consecutive trials of the trajectories were executed (i.e., a trial started at the end point of the previous one), each trial having a duration of 2 s (see fig. S1 for SNN learning convergence). Each induced delay δ_T was maintained for 100 trials, and then increased to the next value; that is, at least 200s of experiment duration per δ_T value. The performance metric given by the mean absolute error (MAE) illustrated the learning convergence of the SNN torque controller (see Materials and Methods) across a wide range of induced delays δ_T (Fig. 2A). Note that the SNN torque controller, regardless of the induced delay δ_T, improved the performance accuracy (MAE) of the factory-default position controller given under no-delay circumstances.

Since the factory-default position controller could not be tested in a time delay framework, we tuned a proportional-derivative (PD) controller for each of the motor tasks using the Ziegler-Nichols method (*32*). The resultant PD torque controller performed similarly to the factory-default position controller under no-delay circumstances (PD MAE = 0.076 rad/s vs. factory-default MAE = 0.077 rad/s for the horizontal circle trajectory, PD MAE = 0.054 rad/s vs. factory-default MAE = 0.055 rad/s for the inclined circle trajectory, and MAE = 0.068 rad/s for both the PD and factory-default controller for the circle-Lissajous sequence), thus, serving as a performance reference (Fig. 2A). In conducting a more in-depth assessment of our SNN, we also developed a conceptually closer analog neural network (ANN) controller. We used the analog cerebellar solution from (*33*, *34*) conveniently adapted for Baxter's 6 DOF in a feedback loop. This ANN model equipped the main form of SNN synaptic plasticity but lacked its temporal correlation capability, i.e., PC long-term depression was heterosynaptically–driven by CF, whilst PC long-term potentiation was related to PF activity (see Materials and Methods). As expected, the ANN performed similarly to the SNN cerebellar solution and better than the default-factory position controller under no-delay circumstances: ANN MAE = 0.021 ± 0.002 rad/s vs. SNN MAE = 0.018 ± 0.004 rad/s for the horizontal circle trajectory, ANN MAE = 0.017 ± 0.002 rad/s vs. SNN MAE = 0.017 ± 0.004 rad/s for the inclined circle trajectory and ANN MAE = 0.019 ± 0.001 rad/s vs. SNN MAE = 0.021 ± 0.004 rad/s for the circle-Lissajous sequence (Fig. 2A).

****Figure 2 about here****

As the induced delay δ_T increased from 0 to 50 ms, the PD and ANN controllers performance degraded significantly (Fig. 2A) due to the instability caused by the large variations/oscillations of the output torque response; i.e., torque variability increased from 0.019 to 0.036 Nm/ms (PD controller) and from 0.016 to 0.026 Nm/ms (ANN controller) per joint for the horizontal trajectory; from 0.026 to 0.051 Nm/ms (PD) and from 0.017 to 0.027 Nm/ms (ANN) per joint for the inclined circle trajectory; from 0.025 to 0.037 Nm/ms (PD) and from 0.026 to 0.028 Nm/ms (ANN) per joint for the circle-Lissajous sequence (Fig. 2, B, C, and D). PD control instability occurred from early stages: delays δ_T over 10 ms for the inclined circle trajectory and the circle-Lissajous sequence, and over 20 ms for the horizontal circle trajectory. The lower capacity to cope with delays for the circle-Lissajous sequence and the inclined circle trajectory, indicated how increasing arm-movement complexity demanded higher PD static gains, followed by an incremental sensitivity (*5*), i.e., the relationship between the input and the output robot system indicating how easily the input initiates a change in the output when the robot is in a steady-state condition. A fine balance between obtaining high performance by increasing PD gains whilst maintaining sensitivity low is required. An *in crescendo* sensitivity may ultimately induce instability (oscillatory PD responses) and compromise compliance with lower delay δ_T values. Similarly to the PD, the ANN controller was driven to instability with delays δ_T above 10 ms for the horizontal circle trajectory, and above 20 ms for the inclined circle and circle-Lissajous sequence. We stopped the experiments at δ_T = 50 ms since safety/compliance could not be guaranteed to the robot itself nor to the personnel due to increasing torque oscillations.

Conversely, the cerebellar predictive behavior of the SNN torque controller provided a stable compliant output regardless of time delays. As the delay δ_T increased from 0 to 80 ms, the MAE of the SNN torque controller barely deviated from the ideal horizontal and inclined circle trajectories and the circle-Lissajous sequence: average MAE = 0.024 ± 0.011, 0.022 ± 0.008 and 0.027 ± 0.007 rad/s respectively (Fig. 2A). For the PD and ANN controllers, 3-4 times larger MAE deviations were obtained: average MAE = 0.099 ± 0.027 (PD controller) and 0.053 ± 0.026 rad/s (ANN controller) for the horizontal circle trajectory, 0.092 ± 0.036 (PD) and 0.061 ± 0.030 rad/s (ANN) for the inclined circle trajectory, 0.097 ± 0.032 (PD) and 0.047 ± 0.021 rad/s (ANN) for the circle-Lissajous sequence. The compliance stability of the SNN controller was reflected in the evolution of the output torque commands as transmission delays were induced (Fig. 2, B, C, and D), i.e., the SNN torque output remained at 0.012 Nm/ms per joint for the two circle trajectories and 0.018 Nm/ms for the circle-Lissajous sequence regardless of the delay increment. The induced δ_T was limited to 80 ms according to the predictive time margin of the deployed learning mechanism (see Materials and Methods).

Outstanding levels of accuracy were achieved by the SNN torque controller in the execution of the trajectories (Fig. 3). Comparative ANN vs SNN results indicated the time-related capability of the SNN form of synaptic plasticity accountable for coping with the delay.

****Figure 3 about here****

STDP at PF-PC copes with the delay. Overcoming the 150 ms delay biological limitation

The presence of the biological sensorimotor delay causes a given sensorimotor state at time t to be received at the CNS at time $t + \delta_a$ (afferent delay), and the subsequent motor command to be applied at time $t + \delta_a + \delta_e$ (efferent delay). The tolerance of the biological learning mechanism to this sensorimotor delay hinges on its ability to use previous synaptic activity to generate predictive motor commands within a predictive time margin of $\delta_a + \delta_e$. Again, we induced transmission delays in R2C and C2R directions whilst performing the horizontal circle trajectory. We first aligned the STDP learning mechanism to cope with the biological sensorimotor delay as well as the predictive temporal margin configured accordingly. We found that the predictive behavior of the SNN controller guaranteed a stable performance as long as time delays were kept within the established predictive time margin. Then, we faced the STDP learning mechanism to larger predictive temporal margins to test whether and to what extent the time delay tolerance of our SNN controller could be modified beyond the biological temporal imposition.

The PF-PC STDP mechanism allowed for motor learning by correlating the sensorimotor information recoded at granular layer into spike patterns with the teaching/error signal provided by CFs to the PC (*31*, *35*). A PF-PC synaptic weight change (Δ_W) occurred after an appropriate temporal sequence of PF-CF de/activations, involving two opposed processes of long-lasting modifications in synaptic strength: long-term potentiation (LTP) and long-term depression (LTD). LTP produced a fixed synaptic weight increment every time a spike arrived to a PC through the PF. Conversely, LTD synaptic weight decrement was triggered by the spikes arriving through the CF to the corresponding PC and depended on the previous activity of the afferent PF. The implementation of this temporal correlation between the teaching/error signal (CF activity) and the previous sensorimotor information (PF activity) followed a convolution kernel with an “eligibility trace” (*31*, *36*), similar to a convolved coincidence detection able to compensate for transmission delays (*37*). This implementation required a kernel “eligibility trace” peak (τ_{LTD}), which established the PF spike arrival time before a CF spike arrival for which the synaptic weight decrement was maximal. By changing τ_{LTD}, the predictive time margin could be accordingly modified (Fig. 4, A and B). Consequently, τ_{LTD} established the amount of time delay ($\delta_T + \delta_C$, transmission plus computation delays) that the SNN controller could tolerate. We found that establishing a τ_{LTD} value involved a fine trade-off between time delay tolerance and the performance accuracy obtained. As the predictive time margin increased, so did the time delay tolerance (Fig. 4B) but the performance error also increased (Fig. 4C).

Electrophysiological recordings (*36*, *38*) show a LTD contribution more acute for those PF spikes which occurred 50 to 150 ms before the CF activity, i.e., τ_{LTD} between 50 and 150 ms. We chose τ_{LTD} = 150 ms to increase the time delay tolerance whilst maintaining the SNN biological plausibility. We found that a kernel “eligibility trace” peak of 150 ms provided robustness against transmission delays up to 80 ms, thus requiring 70 ms for computation delays comprising analog information processing, neural activity computation, analog-to-spike and spike-to-analog conversion, and torque commands application by the robot actuators. Please, see Methods and Supplementary Materials (Annex S1) for a more in depth description of the temporal kernel operation.

****Figure 4 about here****

Benchmarking the non-deterministic time delays

The learning convergence of our SNN output against steady time delays was tested so far; convergence under non-deterministic time delays was still to be analyzed. Here, we characterized the response of our SNN to non-deterministic delays in a lab-controlled scenario. The delay range (from 0 to 80 ms) was covered with a set of gamma distributions from which non-deterministic time delays δ_T were randomly sampled ($\delta_T = \delta_{R2C} + \delta_{C2R}$; $\delta_{R2C} = \delta_{C2R} = \delta_T/2$), providing the following mean delays: 15 ± 5 ms, 25 ± 5 ms, 35 ± 5 ms, 45 ± 5ms, 55 ± 5ms, 65 ± 5ms, and 78 ± 4 ms (see Fig. 5, A and B for the probability density function, PDF, and cumulative distribution function, CDF, of the induced delays). Non-deterministic delays were induced using the set up described in Fig. 1A. For each delay distribution, 100 trials of the horizontal circle trajectory were performed, maintaining MAE values below the precision provided by the factory-default controller (Fig. 5C). Note that gamma distributions are proven to adequately model network delays (*39*, *40*).

Aiming at characterizing a more realistic scenario, we also tested asymmetrical (i.e., $\delta_{R2C} \neq \delta_{C2R}$), non-deterministic delays. Two scenarios were tested: i) δ_{R2C} = 8 ± 3 ms and δ_{C2R} = 40 ± 3 ms (Fig. 5D), ii) δ_{R2C} = 39 ± 2 ms and δ_{C2R} = 9 ± 4 ms (Fig. 5E). We found that the SNN was able to cope with both symmetric and asymmetric non-deterministic delays.

****Figure 5 about here****

Non-deterministic Wi-Fi and cloud-robotics time delays; cerebellar control use cases

We established a robot-controller Wi-Fi connection using a Raspberry Pi 3B+ (RPi) as gateway (Fig. 6, A and B) to circumvent Baxter's lack of wireless support (see Supplementary Materials). The non-deterministic delays inherent to a Wi-Fi connection (*41*) affected asymmetrically to both R2C and C2R directions whilst our SNN controller performed the horizontal circular trajectory. The established dialog between the robot and the controller had a bandwidth consumption of 15 Mbps that was further increased to worsen both sensory and motor delays (see Supplementary Materials). The initial 15 Mbps bandwidth consumption was gradually increased up to 3.6 times simulating control of up to three robots over the same wireless network. We found that the SNN torque controller performance accuracy was kept at the same level regardless of the asymmetrical and non-deterministic time delays (Fig. 6C); i.e., from bandwidth consumption of 15 to 54 Mbps, we obtained an average MAE of 0.025 ± 0.007 rad/s, comparable to the 0.024 ± 0.011 rad/s obtained at the artificial delays scenario with δ_T from of 0 to 80ms. The PD and ANN controllers could not be tested under these circumstances since 50% of the motor delay values were above 20 ms for all bandwidth consumptions (Fig. 6D), which added to the associated sensorial delay would set the PD and ANN controller in the instability zone (Fig. 2A), risking robot and personnel safety.

****Figure 6 about here****

Finally, we used our SNN torque controller in a cloud-robotics framework by establishing a long-distance controller-robot connection over the Internet. The controller was located in Madrid, whereas the robot was located 360 km south (i.e., 224 mi) in Granada (Spain). This remote connection involved 10 Internet hops (Fig. 7A). Two scenarios were tested: i) the robot connected to the Internet through an Ethernet connection via a gateway computer (Fig. 7B); ii) the robot connected to the Internet via Wi-Fi (Fig. 7C). In the first scenario, the sensorimotor time delay accounted for cloud-robotics inherent latency (*42*, *43*). The CDF of the sensorimotor time delays (Fig. 7D) confirmed the 50th, 90th, and 99th percentiles of the exchanged messages below 9, 10, and 12 ms respectively, for both sensorial (R2C direction) and motor (C2R direction) information; a total transmission delay below the 80 ms limit provided by the predictive time margin (Fig. 4B). The round-trip time (RTT) of the remote connection barely varied throughout the day (i.e., average RTT of 20.0 ± 1.3 ms, from 8:00 to 24:00). In the second scenario, the connection was additionally hampered by the Wi-Fi non-deterministic time delays. The CDF confirmed the 50th, 90th, and 99th percentiles below 12, 14, and 20 ms for the sensorial messages; and below 29, 32, and 36 ms for the motor messages (Fig. 7E); values below the 80 ms limit (Fig. 4B). The accuracy obtained in both cases (1st and 2nd scenario MAE = 0.020 ± 0.004 and 0.024 ± 0.007 rad/s) was kept at the same levels as in previous setups. Thus, our SNN torque controller was proven capable of operating in a cloud-robotics framework.

****Figure 7 about here****

Discussion

A well-timed response to stimuli is imperative for body-interaction with changing environments, thus causing human motor control to compensate for the significant time delay between the sensing of a stimuli and its response. In the CNS, these sensorimotor delays are caused by constraints in the neurophysiological substrate, which can be very efficient in computation due to massive parallel neural computing, but inefficient to communicate signals through long axons and slow chemical synapses. Physiologically, the transduction and transport of sensory inputs and motor commands involves: sensing delay, nerve conduction delay, synaptic delay, neuromuscular junction delay, electromechanical delay, and force generation delay (*44*). Consequently, the CNS needs to cope with the uncertainty aroused by these delays to provide accurate motor control. Besides these biologically inherent time delays, the CNS can self-adapt to additional external time delays (*45-47*). The CNS sensorimotor time delay compensation relies on state and sensory prediction; i.e., an estimation of the actions outcome before sensory feedback is available (*48*). The cerebellum plays a pivotal role in this prediction mechanism (*29*, *49-51*) due to its ability to acquire internal models of the human body and external tools through motor learning (*52-55*).

Consequently, cerebellum-inspired solutions have been proposed to different control problems: gaze stabilization (*56-58*), adaptive control of linear (*59*, *60*) and nonlinear (*6*, *61*, *62*) systems, acquisition of forward/inverse (*27*, *63*) dynamic models, or computation of inverse kinematics (*64*). Sensorimotor time delays were also considered by some analog-based cerebellum-inspired approaches recently suggested: i) An analog cerebellar-like functional model embedded with a Smith predictor was able to deal with the control loop inherent sensorimotor time delays, measured below 8 ms (*63*). ii) A cerebellum-inspired adaptive filter model was used to control saccadic eye movements with a delayed error signal temporally aligned at the PF-CF connection (*65*). In this analog solution, the temporal coding at granular layer was modeled as an echo-state network, thus simplifying the complex spatiotemporal processing of the cerebellar information to make the controller suitable for robotic application. iii) An adaptive filter based on the cerebellum and embedded with a reactive controller, implemented an eligibility trace that compensated for the 50 ms delay in the error feedback and the response lags intrinsic to the plant dynamics using different learning rules: a) forward model-based eligibility trace gradient descent (FM-ET); b) Widrow-Hoff (WH) algorithm with a delta-eligibility trace tuned to the error feedback delay (WH+50ms), and tuned to exceed that delay by 20 ms (WH+70ms) (*37*). iv) Control of fast limb movements (i.e., movements lasting less than the total duration of the sensorimotor pathway processing and transmission delays) was provided by a controller involving two fuzzy NNs representing each the cerebellar cortex and DCN (*66*). These solutions, although not of direct application to the present setup as they are constrained to more simple scenarios (simulation studies, numerical experiments, LTI systems, fixed delays, simple dynamics, dynamic-model-dependent), prove the efforts devoted to address the sensorimotor delay challenge from analog approaches. However, these cerebellum-inspired solutions removed the intrinsic temporal aspect naturally present in the spike coding found in biological networks. Compensating the temporal delay was more of a problem for motor control than a cerebellar virtue. Understanding the temporal compensation of the sensorimotor pathway delay within the cerebellum requires a different perspective starting from a more realistic replication of the biologically inherent temporal cerebellar features.

Cellular-level cerebellar-controllers offer an insight into cerebellar function at neuron level. Yet, the significant computational cost of these models (*67*) has traditionally prevented them from real robotic applications. Our SNN cerebellar model, which falls into this cellular-level category, was already tested in a real robotic application (*6*). We suggested and replicated the cerebellar acquisition of internal models as a solution to the non-linear dynamic modeling of elastic cobots; providing real-time, adaptive, and compliant torque control of a 6 DOF robot arm. The cellular-level nature of our cerebellar SNN controller enables the replication of the STDP mechanisms at neuron level. Consistently with the Marr-Albus-Ito cerebellar theory (*68*), we found that the LTD “eligibility trace” temporal margin at PF-PC cell synapses was key in estimating and shaping the cerebellar temporal output. LTD eligibility trace allowed for a temporal record of PF synapses past activity (i.e., the temporal sensorimotor patterns), so that the feedback error/teaching signal from CF arriving after that PF activity could make changes in the PF-PC synapses strength (*69*). A continued exposure to sensorimotor patterns allowed PF-PC synapses to acquire a temporal representation of the relation between the error/teaching signal and previous sensorimotor information (*70*). The precise time correlation between sensorimotor information at PF and the elicited error/teaching signal at CF of our SNN controller provided robustness to sensorimotor time delays.

The fourth industrial revolution, *Industry 4.0*, is leading industrial processes to be connected using Internet technologies (*71*). In robotics, this revolution is reflected in the growing field of cloud-robotics, which conjugates the benefits of Big Data, Cloud Computing and Collective Robot Learning (*42*). Nonetheless, cloud-robotics faces the technical challenge of dealing with communication latencies (*42*, *43*) between the cloud and edge nodes. Motor control can be highly sensitive to time delays as they drive the system towards instability and unmanageability (*5*), ultimately forcing some sort of strategy to address cloud communications latency. Efforts have been devoted to tackle cloud-robotics time delays by minimizing the latency of the existing architecture (*72*), modifying the communications paradigm (*73*) and protocols (*74*), or implementing new communication technologies (*75*). However, the application of these approaches is tied to specific communication architectures, technologies, or protocols. Conversely, a SNN controller able to provide robustness against time delays would solve the cloud-robotics latency challenge regardless of how the controller-robot connection is established. Not only cloud-robotics can benefit from our cerebellar SNN torque controller, but also other robot control schemes that carry inherent time delays such as teleoperation or wireless robot control; relevant to robotic applications such as remote control, factory automation, or HRI. HRI could especially benefit from the application of our SNN torque controller, as it meets the demand for adaptive, compliant robot behavior (*6*) even in the presence of sensorimotor delays.

In this work, we presented a neuroscience approach to a real-world robotic application, providing both lab-controlled setups with synthetic communication delays and real-world setups that fall under higher technology readiness levels (TRL) (*76*) with potential use in cloud-robotics and remote control with long latencies.

PF-PC STDP modeling considerations

Concerning the implemented STDP rule for PF-PC LTD, some considerations need to be noted. This STDP is pivotal in sorting out the PC output credit assignment problem (*77*), i.e., modeling how a change in the weight of PC synapses would impact the behavior of the final cerebellar output; however, it still remains open what occurs to PF-PC adaptation to either a specific delay or to a range of delays at the cerebellar intermediate zone, responsible for controlling the distal extremity muscles. Interestingly, the PF-PC STDP in other cerebellar regions (vermis vs flocculus) adapts differently to the specific delay at which CF error signals shall arrive with respect to MF sensorimotor signals during motor learning (*78*). The vermis receives proprioceptive information from the dorsal columns of the spinal cord and coordinates body posture and locomotion, whereas the flocculonodular lobe receives information from the vestibular nuclei and visual cortex and helps learning basic motor skills found within the vestibulo-ocular system (VOS).

LTD is induced in the flocculus when PFs activate 120 ms before the CFs, assuming a PF-PC LTD monokernel presumably tuned to a unique pathway delay (*78*). This plasticity at PF-to-PC synapses differs from plasticity found at PF-to-PC synapses in the vermis, in which LTD is induced by a range of PF-CF pairing interval (50 to 150 ms), assuming PF-PC LTD multikernels presumably tuned to a set of pathway delays (*78*). It is speculated that the wide range of delays between PF-CF activation inducing LTD may reflect the wide range of pathway delays in the error signals carried by the different CF inputs to the vermis, i.e., from spinal afferent signals with latencies between 10–30 ms (*79*) to cognitive signals with, a priori, longer latencies (*80-82*). Conversely, the flocculus responses to the PF-CF temporal interval are consistent at 120 ms in agreement with the specificity of the pathway delays in the CF error signals found in the VOS (*78*).

In looking for analogies between our robotic pathway delay and what occurs within either the vermis or flocculus pathway delays, we assumed PF-PC LTD monokernel configuration as in the latter. The robotic sensorimotor pathway was equally configured for each Baxter joint (motor and sensor), as it occurs in the VOS. Biology seems to have evolved a PF-PC LTD multikernel solution to meet the different sensory pathway delays converging in the vermis, however, industrial field buses/Ethernet in robotics avoid these problems by design. A PF-PC LTD multikernel approach would impose to configure a different robotic sensorimotor pathway per Baxter motor accordingly, e.g., sensory motor pathways configured with increasing levels of delay according to the corresponding Baxter joint distance to the central CPU, mimicking limbs distance to the cerebellum. However, Baxter motor and encoder data transmission are not meant to operate with these properties. These differences between the propagation of sensorimotor information in the human peripheral nervous system (PNS) and in its robotic counterpart (i.e., signals generated at robot joint sensors all propagated through the same pathway) drove us towards the monokernel solution.

For a widely distributed robotic platform with several ms delay differences between the interconnected elements, i.e., segmented in different sensorimotor pathways, together with RT capacity being granted despite the multikernel approach higher computational cost (see fig. S2), a multikernel solution adapted to a multipaired cerebellar architecture (*83*) could be a good approach to avoid the trade-off between accuracy and delay tolerance encountered in the monokernel solution.

Materials and Methods

Objective and study design

The objective of our study was to validate the robustness against time delays of a cerebellar-based SNN torque controller, thus applying CNS inherent features to robotic control. The SNN controller ran on an Intel® Core™ i7-5820K CPU at 3.30GHz with 12 cores, 32GB of RAM, and a GPU GeForce RTX 2080/PCIe/SSE2. The controlled front-end body was a Baxter robot (*84*): a two-armed collaborative robot equipped with both position and torque control capability. Our SNN torque controller together with Baxter's internal series elastic actuators (SEAs) ensure both active and passive compliance (*6*).

The cerebellar neural network

The cerebellar neural network consisted of 62040 Leaky Integrate and Fire (LIF) neurons and ~36.4M synapses (36M endowed with plasticity) mimicking the cerebellar structure. The network size was a trade-off between Baxter's working space coverage and RT working capability. The neurons were distributed across five different layers (see Fig. 1, B and C), and every layer was divided into six microcomplexes (*85*) to control each of the six DOF. The neural layer distribution was the following: mossy fibers (MFs, 240 neurons), granule cells (GCs, 60000 neurons), climbing fibers (CFs, 600 neurons), Purkinje cells (PCs, 600 neurons), and deep cerebellar nuclei (DCN, 600 neurons). The input sensorimotor information (actual and desired robot analog state translated into spiking patterns) was induced through the MF layer and transmitted through excitatory afferents toward the GC layer. The sensorimotor information was then recoded into somatosensory neural activity at the GC layer and then propagated toward the PC layer via the parallel fiber excitatory connections (PFs), i.e., GCs axons. The PC layer also received, via excitatory connections from the CF layer, the teaching/error signal, i.e., the mismatch between the actual and desired robot state translated into neural spikes. Finally, the DCN layer received inhibitory synapses from the PC and excitatory synapses from the CF and MF layers. The DCN neural activity was translated into an analog motor command which was sent to the robot, thus closing the loop. Note that each of the six microcomplexes comprising the CF-PC-DCN subcircuit was divided into two halves (agonist/antagonist), each half controlling the clock/anticlockwise movement of the robot joint actuator. This structure mimicked the physiological antagonistic muscle pairs located in opposite sides of each arm joint (*86*); i.e., one half of the microcomplex contracts the agonist muscle, the other half contracts the antagonist muscle.

The cerebellar input-output response was adjusted at the PF-PC connection, where the synaptic weight distribution was adapted through a STDP mechanism correlating both the sensorimotor information and the teaching/error signal. Thus, synaptic plasticity allowed error reduction through iterative trial and error motor task executions. The topology of the neural network is summarized in Table 1, and the overall depiction of the cerebellar neural network is shown in Fig. 1C.

LIF neurons (*87*) (see Supplementary Materials, Annex S2) were used to build the cerebellar neural network due to their minimal computational cost, thus enabling our real time computation requirement. See (*6*) for an in-depth review on the cerebellar neural layers, their connectivity, and neuron models.

****Table 1 about here****

The STDP mechanism

The STDP mechanism deployed at the PF-PC synapses conjugated two opposed processes of synaptic change: LTD and LTP. These two processes, compensating and complementing each other, allowed the regulation of the cerebellar output commands by temporally correlating the teaching/error signal (CF activity) and the previous sensorimotor information (PF activity). See Supplementary Materials, Annex S1, for a more in depth description of the temporal kernel operation.

The LTD process convolved the CF and PF activity as follows:

$$LTD\Delta w_{PF_j-PC_i}(t) = \beta \cdot \int_{-\infty}^{t_{CFspike}} k\left(t - t_{CFspike}\right) \cdot \delta_{PFspike}(t) \cdot dt \qquad (8)$$

$$k(x) = \begin{cases} \dfrac{-\left(x + d_k\right)}{\tau_{LTD} - d_k} \cdot e^{\frac{x+d_k}{\tau_{LTD}-d_k}+1} & if\ x < -d_k \\ 0 & if\ x \geq -d_k \end{cases} \qquad (9)$$

where $\Delta w_{PF_j-PC_i}$ is the synaptic weight change between the jth PF and the ith PC, β = -0.0008 nS is the synaptic weight decrement, δ_{PF} is the Dirac delta function of an afferent spike from a PF, *k(x)* defines the integrative kernel, d_k = 120 ms allowed the adjustment of the kernel width, and τ_{LTD} is the kernel "eligibility trace" peak. The kernel maximum value (*k(x)* = 1) is obtained when x = $-\tau_{LTD}$, that is, the synaptic weight decrement is maximum for those PF spikes received τ_{LTD} ms before the CF spike arrival. For our SNN torque controller we established τ_{LTD} = 150 ms.

The LTP process produced a fixed synaptic weight increment every time a spike arrived to a PC through the PF as defined by:

$$LTP\Delta w_{PF_j-PC_i}(t) = \alpha \cdot \delta_{PFspike}(t) \cdot dt \qquad (10)$$

where $\Delta w_{PF_j-PC_i}$ is the synaptic weight change between the jth PF and the ith PC, α = 0.002 nS is the synaptic efficacy increment, and δ_{PF} is the Dirac delta function of an afferent spike from a PF.

These two processes regulated the PF-PC synaptic weight and, therefore, shaped the SNN torque controller output commands. A PF-PC synaptic weight decrement would be translated into a reduction of the DCN inhibition caused by the PC, therefore increasing the DCN output activity. Conversely, a PF-PC synaptic weight increase, due to a low error signal and, therefore, a scarce CF-PC activity, would decrease the DCN output activity. A well synchronized sequence of increased/decreased DCN activity tuned the cerebellar output motor commands reducing the overall performance error.

Translation from analog sensorial states to neural activity

The SNN sensorial input information, originated in analog form at Baxter's sensors (Q_a, $\dot{Q}_a$) and the trajectory generator (Q_d, $\dot{Q}_d$), had to be translated into neural activity (MF activity) that the SNN could process. The 240 MFs were divided into six microcomplexes (one per DOF) of 40 neurons each. Each microcomplex was again divided into four subgroups of 10 neurons each, devoted to coding Q_a, $\dot{Q}_a$, Q_d, $\dot{Q}_d$ respectively. Each of the 10 neurons of the subgroup acted as a sensory receptor for a specific interval within the analog signal joint range; i.e., a neuron fired a spike ($\delta_{MFspike}(t)$) when the analog value (Q) was within its receptor interval (R_n), described as follows:

$$\delta_{MFspike}(t) \leftrightarrow Q(t) \in R_n \tag{11}$$

$$R_n = c_n \pm w_n \tag{12}$$

$$c_n = r_{\min} + \left(\frac{r_{\max} - r_{\min}}{S-1} \right) \cdot n \tag{13}$$

$$w_n = \frac{1}{2} \cdot \left(\frac{r_{\max} - r_{\min}}{S-1} \right) \tag{14}$$

where $\delta_{MFspike}(t)$ defines the Dirac delta function of an afferent spike from a MF, n = [0, 9] stands for the neuron index within the subgroup, c_n and w_n define the center and width of the interval, [r_{min}, r_{max}] denotes the joint range in radians of the analog signal, and S = 10 stands for the total number of the subgroup neurons. Since the receptor intervals within the subgroup were non-overlapping, only four MFs per microcomplex were active at each time step. Thus, the current sensorial state was univocally coded into neural activity. Please see fig. S3 for a representation of the analog input signals coding at MF layer using the timing of spikes.

The teaching/error signal $\varepsilon(t)$, obtained by comparing the desired (Q_d, $\dot{Q}_d$) and actual robot state (Q_a, $\dot{Q}_a$), was also translated from the analog to the spike domain (CFs activity). The 600 CFs were divided into six microcomplexes (one per DOF) of 100 neurons each, the first/last 50 cells were devoted to the agonist/antagonist sensed error, i.e., positive/negative joint error. Electrophysiological recordings of the CFs show a chaotic and low firing rate, between 1 and 10 Hz per neuron (*88*). The low firing rate could hamper capturing the high-frequency information of the teaching/error signal; however, the chaotic firing allows the statistical sampling of the entire signal range over multiple trials (*88, 89*). We replicated this behavior using a Poisson model: given the error signal $\varepsilon(t)$ and a random number $\eta(t)$ ϵ [0, 1], the given CF fired a spike $\delta_{CFspike}(t) \leftrightarrow \varepsilon(t) > \eta(t)$, remaining silent otherwise (*27, 38*).

Translation from neural activity to torque commands

The DCN neural activity, i.e., output cerebellar activity, was translated into analog torque commands (τ_j) before being sent to Baxter's actuators. There were six DCN microcomplexes, one per DOF. The spike-to-analog translation of each microcomplex activity was performed at every time step (2 ms) as follows:

$$DCN_{j,i}(t) = \int_{t-t_{step}}^{t} \delta_{DCN_{j,i}}(t) \cdot dt \tag{15}$$

$$DCN_{output,j}(t) = \alpha_j \cdot \sum_{i=1}^{N=50} DCN_{j,i}(t) - \sum_{i=51}^{N=100} DCN_{j,i}(t) \tag{16}$$

where j = [1, 6] for each of the six DOF; i = [1, 100] defines the DCN index within the microcomplex, the first/last 50 DCN cells were devoted to the agonist/antagonist joint movement; $\delta(t)$ is the Dirac delta function of a spike arrival; α_j = (0.75, 1.1, 0.375, 0.63, 0.078, 0.078) is a factor to weight the DCN output according to the relative position, orientation, and mass of each joint.

At the robot side, the DCN output torque values entered a mean filter, whose size varied at each time step depending on the number of predicted torque samples available (x) to generate a torque command. A torque command sample generated at time t with a prediction of δ_e ms shall be applied by the robot actuators at time $t + \delta_e$. When the time delay affecting that torque command sample was shorter than δ_e, the torque command sample was received at the robot side before its application time. In that event, that torque command sample would operate as a future torque command sample at the mean filter. Past torque command samples were also used to normalize the mean filter to the current time step (t), as follows:

$$\tau_j(t) = \frac{1}{2x+1} \cdot \left(\sum_{i=0}^{x} DCN_{output,j}\left(t + i \cdot t_{step}\right) + \sum_{i=1}^{x} DCN_{output,j}\left(t - i \cdot t_{step}\right) \right) \tag{17}$$

where $x \in [2, 10]$. This filter mimicked the low-pass filter behavior of muscles before sending torque commands to Baxter's actuators. When x was less than 2 (i.e., one or less than one available future torque command samples), we applied the previous time step torque command with 99.8% reduction. In the event of x being less than 2 for successive time steps, the applied torque command was gradually reduced to 0 Nm to provide a safe stopping. x equals 10 meant best case scenario, i.e., 10 predicted, 10 past and the current torque samples for a total of about 42 ms temporal window. This was in agreement with the upper motor neuron maximal discharge rates during slow isometric ramp contractions (*90*). Predicted, past and current torque samples were placed within the mean filter based on their application time.

Desired trajectories definition

We designed three motor tasks to be performed by the SNN torque controller under the described time delay conditions. The motor tasks were fast movements in smooth trajectories consisting of sinusoidal-like position and velocity profiles per joint; involving the complex dynamics of a 6 DOF robotic arm, including interaction forces between joints (*91-93*). These motor tasks depicted three different desired trajectories to be followed by Baxter's left arm end-effector: a horizontal (*xy* plane) circle trajectory, an inclined (*xyz* plane) circle trajectory, and a Lissajous trajectory (δ = π/2, a = 1, b = 2), i.e., eight-like Cartesian trajectory in the horizontal plane (*xy* plane) (*34*, *91*). Please see Supplementary Materials, Annex S3, for the mathematical description of the trajectories.

Performance accuracy and learning convergence measurement

To evaluate the SNN torque controller performance, we compared the desired and actual trajectory; i.e., desired (Q_d) compared to actual (Q_a) joint position at each time step. The average difference of all joints provided the MAE, serving as the performance accuracy metric:

$$MAE_{\text{joint}} = \frac{t_{step}}{T} \sum_{t=0s}^{T} \left(Q(t)_{desired} - Q(t)_{actual} \right) \tag{21}$$

$$MAE = \frac{1}{N} \sum_{j=1}^{N} MAE_j \tag{22}$$

where N = 6 stands for the six DOF. For each of the tested time delay conditions we used 100 consecutive trials of the trajectories to obtain the mean and standard deviation of the MAE for that given time delay condition.

To evaluate the learning convergence of the SNN and ANN torque controllers output response, we studied the average joint torque variability ($\Delta\tau$). Since the SNN and ANN torque controllers provided a non-deterministic output, first we obtained the 100 iterations average torque per joint as follows:

$$\tau_j = \frac{1}{100} \sum_{i=1}^{100} \tau_i(t) \tag{23}$$

where i = [1,100] stands for the iteration number, each iteration having a duration of 2 s, i.e., t = [0, 2]. Then, we found the average joint torque variability as described by:

$$\Delta\tau_j = \frac{t_{step}}{T} \sum_{t=t_{step}}^{T} \frac{\tau_j(t) - \tau_j(t - t_{step})}{t_{step}} \tag{24}$$

$$\Delta\tau = \frac{1}{N} \sum_{j=1}^{N} \Delta\tau_j \tag{25}$$

Since the PD inner computation was deterministic, we did not need the 100 iterations average torque, we used these last two equations applied to one iteration output torque to obtain the PD controller output torque variability.

Modules implementation

A Robot Operating System (ROS) framework allowed the processing and transmission of information between the control loop modules, and the spike-to-analog and analog-to-spike translation. For reproducibility purposes, the source code for the PD, ANN and SNN controllers as well as the experimental setup are available at https://github.com/EduardoRosLab/EDLUT_BAXTER_DELAYS.

Supplementary Materials

Annex S1. The “eligibility trace” and how it enters the learning rule equation.
Annex S2. The Leaky integrate-and-fire neuron model (LIF).
Annex S3. Mathematical description of the trajectories.
Annex S4. Robot-Controller Wi-Fi gateway.
Annex S5. Induction of additional Wi-Fi bandwidth.
Annex S6. The ANN cerebellar model.
Fig. S1. Trajectory learning convergence curves.
Fig. S2. Cerebellar SNN multikernel vs monokernel solution coping with time delays.
Fig. S3. Spike coding at the input MF layer.
Table S1. Neuron parameter values.
Movie S1. Remote Wi-Fi cobot control.

References and Notes

1. A. Vysocky, P. Novak, Human-Robot collaboration in industry. *MM Science Journal* **9**, 903-906 (2016).
2. H. M. Van der Loos, D. J. Reinkensmeyer, E. Guglielmelli, in *Springer handbook of robotics*. (Springer, 2016), pp. 1685-1728.
3. W. Bluethmann, R. Ambrose, M. Diftler, S. Askew, E. Huber, M. Goza, F. Rehnmark, C. Lovchik, D. Magruder, Robonaut: A robot designed to work with humans in space. *Autonomous robots* **14**, 179-197 (2003).
4. A. De Santis, B. Siciliano, A. De Luca, A. Bicchi, An atlas of physical human–robot interaction. *Mechanism and Machine Theory* **43**, 253-270 (2008).
5. M. Hernando, E. Gambao, in *Advances in Telerobotics*. (Springer, 2007), pp. 303-320.
6. I. Abadía, F. Naveros, J. A. Garrido, E. Ros, N. R. Luque, On Robot Compliance: A Cerebellar Control Approach. *IEEE transactions on cybernetics*, (2019).
7. K. G. Shin, X. Cui, Computing time delay and its effects on real-time control systems. *IEEE Transactions on control systems technology* **3**, 218-224 (1995).
8. G. Niemeyer, C. Preusche, S. Stramigioli, D. Lee, in *Springer handbook of robotics*. (Springer, 2016), pp. 1085-1108.
9. S. Shen, A. Song, T. Li, Predictor-based motion tracking control for cloud robotic systems with delayed measurements. *Electronics* **8**, 398 (2019).
10. N. J. Ploplys, P. A. Kawka, A. G. Alleyne, Closed-loop control over wireless networks. *IEEE Control Systems Magazine* **24**, 58-71 (2004).
11. A. Sorribes, Tesis doctoral, UPV, Valencia, Espana, (2011).
12. V. A. Oliveira, L. V. Cossi, M. C. Teixeira, A. M. Silva, Synthesis of PID controllers for a class of time delay systems. *Automatica* **45**, 1778-1782 (2009).
13. M. Bozorg, F. Termeh, Domains of PID controller coefficients which guarantee stability and performance for LTI time-delay systems. *Automatica* **47**, 2122-2125 (2011).
14. H. Xu, A. Datta, S. Bhattacharyya, in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*. (IEEE, 2003), vol. 4, pp. 4038-4043.
15. J. E. Normey-Rico, E. F. Camacho, Dead-time compensators: A survey. *Control engineering practice* **16**, 407-428 (2008).
16. O. Smith, Closer Control of Loops with Dead Time, Chem. Engng. Progr. *53, S. 217* **219**, (1957).
17. M. Bowthorpe, M. Tavakoli, H. Becher, R. Howe, Smith predictor-based robot control for ultrasound-guided teleoperated beating-heart surgery. *IEEE Journal of biomedical and Health Informatics* **18**, 157-166 (2013).
18. A. Manitius, A. Olbrot, Finite spectrum assignment problem for systems with delays. *IEEE transactions on Automatic Control* **24**, 541-552 (1979).

19. I. Vörös, B. Várszegi, D. Takács, in *Dynamic Systems and Control Conference*. (American Society of Mechanical Engineers, 2019), vol. 59162, pp. V003T018A002.
20. D. Rus, M. T. Tolley, Design, fabrication and control of soft robots. *Nature* **521**, 467-475 (2015).
21. A. Diamond, R. Knight, D. Devereux, O. Holland, Anthropomimetic robots: concept, construction and modelling. *International Journal of Advanced Robotic Systems* **9**, 209 (2012).
22. H. Chaoui, P. Sicard, W. Gueaieb, ANN-based adaptive control of robotic manipulators with friction and joint elasticity. *IEEE Transactions on Industrial Electronics* **56**, 3174-3187 (2009).
23. A. Sargolzaei, M. Abdelghani, K. K. Yen, S. Sargolzaei, Sensorimotor control: computing the immediate future from the delayed present. *BMC bioinformatics* **17**, 245 (2016).
24. J. Eyre, S. Miller, V. Ramesh, Constancy of central conduction delays during development in man: investigation of motor and somatosensory pathways. *The Journal of physiology* **434**, 441-452 (1991).
25. M. Gerwig, K. Hajjar, A. Dimitrova, M. Maschke, F. P. Kolb, M. Frings, A. F. Thilmann, M. Forsting, H. C. Diener, D. Timmann, Timing of conditioned eyeblink responses is impaired in cerebellar patients. *Journal of Neuroscience* **25**, 3919-3931 (2005).
26. J. Stein, Cerebellar forward models to control movement. *The Journal of physiology* **587**, 299 (2009).
27. N. R. Luque, J. A. Garrido, R. R. Carrillo, J.-M. C. Olivier, E. Ros, Cerebellarlike corrective model inference engine for manipulation tasks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **41**, 1299-1312 (2011).
28. S. Ghosh-Dastidar, H. Adeli, in *Advances in Computational Intelligence*. (Springer, 2009), pp. 167-178.
29. A. J. Bastian, Learning to predict the future: the cerebellum adapts feedforward movement control. *Current opinion in neurobiology* **16**, 645-649 (2006).
30. R. R. Carrillo, F. Naveros, E. Ros, N. R. Luque, A Metric for Evaluating Neural Input Representation in Supervised Learning Networks. *Frontiers in neuroscience* **12**, 913 (2018).
31. N. R. Luque, J. A. Garrido, F. Naveros, R. R. Carrillo, E. D'Angelo, E. Ros, Distributed cerebellar motor learning: a spike-timing-dependent plasticity model. *Frontiers in computational neuroscience* **10**, 17 (2016).
32. J. G. Ziegler, N. B. Nichols, Optimum settings for automatic controllers. *trans. ASME* **64**, (1942).
33. J. A. Garrido, N. R. Luque, E. D'Angelo, E. Ros, Distributed cerebellar plasticity implements adaptable gain control in a manipulation task: a closed-loop robotic simulation. *Front. Neural Circuits* **7**, (2013).
34. N. R. Luque, J. A. Garrido, R. R. Carrillo, E. D'Angelo, E. Ros, Fast convergence of learning requires plasticity between inferior olive and deep cerebellar nuclei in a manipulation task: a closed-loop robotic simulation. *Frontiers in computational neuroscience* **8**, 97 (2014).
35. N. R. Luque, F. Naveros, R. R. Carrillo, E. Ros, A. Arleo, Spike burst-pause dynamics of Purkinje cells regulate sensorimotor adaptation. *PLoS computational biology* **15**, e1006298 (2019).
36. R. Kettner, S. Mahamud, H.-C. Leung, N. Sitkoff, J. Houk, B. Peterson, A. Barto, Prediction of complex two-dimensional trajectories by a cerebellar model of smooth pursuit eye movement. *Journal of neurophysiology* **77**, 2115-2130 (1997).
37. I. Herreros, X. Arsiwalla, P. Verschure, in *Advances in Neural Information Processing Systems*. (2016), pp. 3828-3836.

38. C. Boucheny, R. Carrillo, E. Ros, J.-M. C. Olivier, in *International Work-Conference on Artificial Neural Networks*. (Springer, 2005), pp. 136-144.

39. A. Mukherjee, On the dynamics and significance of low frequency components of Internet load. (1992).

40. N. A. Ali, E. Ekram, A. Eljasmy, K. Shuaib, in *2008 IEEE/ACS International Conference on Computer Systems and Applications*. (IEEE, 2008), pp. 236-240.

41. C. Pei, Y. Zhao, G. Chen, R. Tang, Y. Meng, M. Ma, K. Ling, D. Pei, in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. (IEEE, 2016), pp. 1-9.

42. B. Kehoe, S. Patil, P. Abbeel, K. Goldberg, A survey of research on cloud robotics and automation. *IEEE Transactions on automation science and engineering* **12**, 398-409 (2015).

43. G. Hu, W. P. Tay, Y. Wen, Cloud robotics: architecture, challenges and applications. *IEEE network* **26**, 21-28 (2012).

44. H. L. More, J. M. Donelan, Scaling of sensorimotor delays in terrestrial mammals. *Proceedings of the Royal Society B: Biological Sciences* **285**, 20180613 (2018).

45. M. Rohde, M. O. Ernst, Time, agency, and sensory feedback delays during action. *Current Opinion in Behavioral Sciences* **8**, 193-199 (2016).

46. M. Rohde, G. Altan, M. O. Ernst, When vision lags, motor prediction follows. *bioRxiv*, (2020).

47. S. Franklin, J. Česonis, R. Leib, D. W. Franklin, in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. (IEEE, 2019), pp. 1517-1520.

48. D. M. Wolpert, Z. Ghahramani, Computational principles of movement neuroscience. *Nature neuroscience* **3**, 1212-1217 (2000).

49. M. G. Paulin, The role of the cerebellum in motor control and perception. *Brain, behavior and evolution* **41**, 39-50 (1993).

50. R. Shadmehr, M. A. Smith, J. W. Krakauer, Error correction, sensory prediction, and adaptation in motor control. *Annual review of neuroscience* **33**, 89-108 (2010).

51. A. Giovannucci, A. Badura, B. Deverett, F. Najafi, T. D. Pereira, Z. Gao, I. Ozden, A. D. Kloth, E. Pnevmatikakis, L. Paninski, Cerebellar granule cells acquire a widespread predictive feedback signal during motor learning. *Nature neuroscience* **20**, 727-734 (2017).

52. M. Kawato, H. Gomi, A computational model of four regions of the cerebellum based on feedback-error learning. *Biological cybernetics* **68**, 95-103 (1992).

53. M. Shidara, K. Kawano, H. Gomi, M. Kawato, Inverse-dynamics model eye movement control by Purkinje cells in the cerebellum. *Nature* **365**, 50-52 (1993).

54. D. M. Wolpert, R. C. Miall, M. Kawato, Internal models in the cerebellum. *Trends in cognitive sciences* **2**, 338-347 (1998).

55. H. Imamizu, S. Miyauchi, T. Tamada, Y. Sasaki, R. Takino, B. PuÈtz, T. Yoshioka, M. Kawato, Human cerebellar activity reflecting an acquired internal model of a new tool. *Nature* **403**, 192-195 (2000).

56. A. Lenz, S. R. Anderson, A. G. Pipe, C. Melhuish, P. Dean, J. 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**, 1420-1433 (2009).

57. F. Naveros, N. R. Luque, E. Ros, A. Arleo, VOR adaptation on a humanoid iCub robot using a spiking cerebellar model. *IEEE transactions on cybernetics*, (2019).

58. C. Casellato, A. Antonietti, J. A. Garrido, A. Pedrocchi, E. D'Angelo, in *5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*. (IEEE, 2014), pp. 813-818.

59. E. D. Wilson, T. Assaf, M. J. Pearson, J. M. Rossiter, P. Dean, S. R. Anderson, J. Porrill, Biohybrid control of general linear systems using the adaptive filter model of cerebellum. *Frontiers in neurorobotics* **9**, 5 (2015).

60. A. Antonietti, D. Martina, C. Casellato, E. D'Angelo, A. Pedrocchi, Control of a humanoid nao robot by an adaptive bioinspired cerebellar module in 3d motion tasks. *Computational intelligence and neuroscience* **2019**, (2019).

61. E. D. Wilson, T. Assaf, M. J. Pearson, J. M. Rossiter, S. R. Anderson, J. Porrill, P. Dean, Cerebellar-inspired algorithm for adaptive control of nonlinear dielectric elastomer-based artificial muscle. *Journal of The Royal Society Interface* **13**, 20160547 (2016).

62. J. Porrill, P. Dean, Recurrent cerebellar loops simplify adaptive control of redundant and nonlinear motor systems. *Neural computation* **19**, 170-193 (2007).

63. S. Tolu, M. C. Capolei, L. Vannucci, C. Laschi, E. Falotico, M. V. Hernandez, A Cerebellum-Inspired Learning Approach for Adaptive and Anticipatory Control. *International Journal of Neural Systems* **30**, 1950028 (2020).

64. M. Asadi-Eydivand, M. M. Ebadzadeh, M. Solati-Hashjin, C. Darlot, N. A. A. Osman, Cerebellum-inspired neural network solution of the inverse kinematics problem. *Biological cybernetics* **109**, 561-574 (2015).

65. H. T. Kalidindi, L. Vannucci, C. Laschi, E. Falotico, Cerebellar adaptive mechanisms explain the optimal control of saccadic eye movements. *Bioinspiration & Biomimetics*, (2020).

66. A. Salimi-Badr, M. M. Ebadzadeh, C. Darlot, Fuzzy neuronal model of motor control inspired by cerebellar pathways to online and gradually learn inverse biomechanical functions in the presence of delay. *Biological cybernetics* **111**, 421-438 (2017).

67. P. van der Smagt, M. A. Arbib, G. Metta, in *Springer handbook of robotics*. (Springer, 2016), pp. 2069-2094.

68. T. Yamazaki, W. Lennon, Revisiting a theory of cerebellar cortex. *Neuroscience research* **148**, 1-8 (2019).

69. R. S. Sutton, A. G. Barto, *Reinforcement learning: An introduction*. (MIT press, 2018).

70. D. Narain, E. D. Remington, C. I. De Zeeuw, M. Jazayeri, A cerebellar mechanism for learning prior distributions of time intervals. *Nature communications* **9**, 1-12 (2018).

71. H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, M. Hoffmann, Industry 4.0. *Business & information systems engineering* **6**, 239-242 (2014).

72. D. M. Lofaro, A. Asokan, E. M. Roderick, in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. (IEEE, 2015), pp. 519-526.

73. R. Imai, R. Kubo, in *2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE)*. (IEEE, 2014), pp. 565-566.

74. N. Tian, A. K. Tanwani, K. Goldberg, S. Sojoudi, in *International Symposium on Robotics Research (ISRR)*. (2019), pp. 1-16.

75. F. Voigtländer, A. Ramadan, J. Eichinger, C. Lenz, D. Pensky, A. Knoll, in *2017 International Symposium on Computer Science and Intelligent Controls (ISCSIC)*. (IEEE, 2017), pp. 69-72.

76. J. C. Mankins, Technology readiness levels. *White Paper, April* **6**, 1995 (1995).

77. B. A. Richards, T. P. Lillicrap, Dendritic solutions to the credit assignment problem. *Current opinion in neurobiology* **54**, 28-36 (2019).

78. A. Suvrathan, H. L. Payne, J. L. Raymond, Timing rules for synaptic plasticity matched to behavioral function. *Neuron* **92**, 959-967 (2016).

79. A. Berthoz, R. Llinás, Afferent neck projection to the cat cerebellar cortex. *Experimental brain research* **20**, 385-401 (1974).

80. D. Menghini, M. Di Paola, R. Murri, F. Costanzo, C. Caltagirone, S. Vicari, L. Petrosini, Cerebellar vermis abnormalities and cognitive functions in individuals with Williams syndrome. *Research in developmental disabilities* **34**, 2118-2126 (2013).

81. C. J. Stoodley, J. D. Schmahmann, Functional topography of the human cerebellum. *Handbook of Clinical Neurology* **154**, 59-70 (2018).

82. C. J. Stoodley, E. M. Valera, J. D. Schmahmann, Functional topography of the cerebellum for motor and cognitive tasks: an fMRI study. *Neuroimage* **59**, 1560-1570 (2012).

83. D. M. Wolpert, M. Kawato, Multiple paired forward and inverse models for motor control. *Neural networks* **11**, 1317-1329 (1998).

84. C. Fitzgerald, in *2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*. (IEEE, 2013), pp. 1-6.

85. M. Ito, Cerebellar microcomplexes. *International review of neurobiology* **41**, 475 (1997).

86. M. L. Latash, Muscle coactivation: definitions, mechanisms, and functions. *Journal of neurophysiology* **120**, 88-104 (2018).

87. W. Gerstner, W. M. Kistler, *Spiking neuron models: Single neurons, populations, plasticity*. (Cambridge university press, 2002).

88. S. Kuroda, K. Yamamoto, H. Miyamoto, K. Doya, M. Kawato, Statistical characteristics of climbing fiber spikes necessary for efficient cerebellar learning. *Biological cybernetics* **84**, 183-192 (2001).

89. J. Keating, W. Thach, Nonclock behavior of inferior olive neurons: interspike interval of Purkinje cell complex spike discharge in the awake behaving monkey is random. *Journal of neurophysiology* **73**, 1329-1340 (1995).

90. J. Duchateau, S. Baudry, Maximal discharge rate of motor units determines the maximal rate of force development during ballistic contractions in human. *Frontiers in Human Neuroscience* **8**, (2014); published online Epub2014-April-22 (10.3389/fnhum.2014.00234).

91. P. van der Smagt, Benchmarking cerebellar control. *Robotics and Autonomous Systems* **32**, 237-251 (2000).

92. N. Schweighofer, M. A. Arbib, M. Kawato, Role of the cerebellum in reaching movements in humans. I. Distributed inverse dynamics control. *European Journal of Neuroscience* **10**, 86-94 (1998).

93. H. Hoffmann, G. Petkos, S. Bitzer, S. Vijayakumar, in *ICINCO-ICSO*. (2007), pp. 262-269.

94. T. Hirano, Long-term depression and other synaptic plasticity in the cerebellum. *Proc Jpn Acad Ser B Phys Biol Sci* **89**, 183-195 (2013)10.2183/pjab.89.183).

95. R. S. Sutton, A. G. Barto, Toward a modern theory of adaptive networks: expectation and prediction. *Psychological review* **88**, 135 (1981).

96. A. G. Barto, R. S. Sutton, C. W. Anderson, Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE transactions on systems, man, and cybernetics,* 834-846 (1983).

97. J. L. Raymond, S. G. Lisberger, Neural learning rules for the vestibulo-ocular reflex. *Journal of Neuroscience* **18**, 9112-9129 (1998).

98. V. Lev-Ram, S. B. Mehta, D. Kleinfeld, R. Y. Tsien, Reversing cerebellar long-term depression. *Proceedings of the National Academy of Sciences* **100**, 15989-15993 (2003).

99. E. Ros, R. R. Carrillo, E. M. Ortigosa, B. Barbour, R. Agís, Event-driven simulation scheme for spiking neural networks using lookup tables to characterize neuronal dynamics. *Neural Comput.* **18**, 2959-2993 (2006).

100. S. Chitta, I. Sucan, S. Cousins, Moveit![ros topics]. *IEEE Robotics & Automation Magazine* **19**, 18-19 (2012).

101. J. Dugan, S. Elliott, B. A. Mah, J. Poskanzer, K. Prabhu, iPerf-The ultimate speed test tool for TCP, UDP and SCTP. *línea]. Available: https://iperf. fr.[Último acceso: 23 Mayo 2018]*, (2014).

102. S. Ostojic, N. Brunel, From spiking neuron models to linear-nonlinear models. *PLoS Comput Biol* **7**, e1001056 (2011).

103. T. Yamazaki, S. Tanaka, Neural modeling of an internal clock. *Neural computation* **17**, 1032-1058 (2005).

104. T. Honda, T. Yamazaki, S. Tanaka, S. Nagao, T. Nishino, Stimulus-dependent state transition between synchronized oscillation and randomly repetitive burst in a model cerebellar granular layer. *PLoS Comput Biol* **7**, e1002087 (2011).

105. T. Yamazaki, S. Tanaka, The cerebellum as a liquid state machine. *Neural Networks* **20**, 290-297 (2007).

106. T. Yamazaki, S. Tanaka, A spiking network model for passage- of- time representation in the cerebellum. *European Journal of Neuroscience* **26**, 2279-2292 (2007).

107. T. Yamazaki, S. Tanaka, Computational models of timing mechanisms in the cerebellar granular layer. *The cerebellum* **8**, 423-432 (2009).

108. R. Apps, M. Garwicz, Anatomical and physiological foundations of cerebellar information processing. *Nature Reviews Neuroscience* **6**, 297-311 (2005).

Acknowledgments

Acknowledgments: The authors would like to thank Dr. Jesús A. Garrido, Dr. Francisco Barranco, and Dr. Jorge Navarro for their research inputs. **Funding:** This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 891774. European Union Human Brain Project Specific Grant Agreement 3 (H2020-RIA. 945539). European Union and Junta de Andalucía, CEREBIO (P18-FR-2378). FEDER-Junta de Andalucía (A-TIC-276-UGR18) and the National Grant INTSENSO (MICINN-FEDER-PID2019-109991GB-I00). **Author contributions:** I.A. and F.N. developed the experimental setup and extracted the experimental results. I.A., F.N, and N.R.L. conducted the analysis of the results. R.R.C. and N.R.L. conceived the proposed approach and the working hypothesis studied in this work. E.R. contributed to the development of the experimental setup. All authors worked on writing and revising the manuscript. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** All data needed to evaluate the conclusions in this paper are present in the paper or the Supplementary Materials. The source code is available at: https://github.com/EduardoRosLab/EDLUT_BAXTER_DELAYS.

Figure Captions

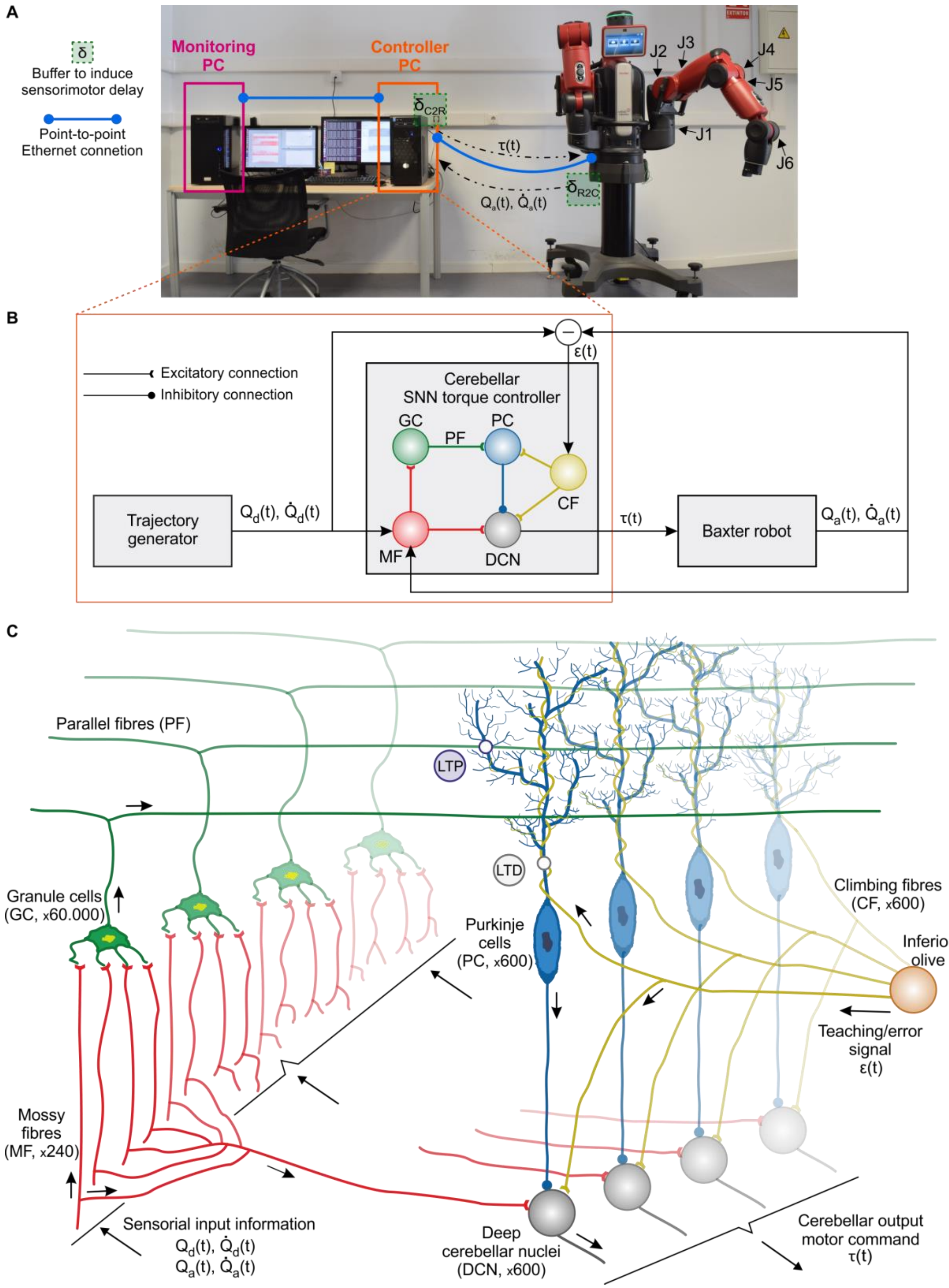

Fig. 1. Cerebellar control loop. (A) Experimental setup in which communication time delays were artificially induced within the cerebellar control loop. The computer allocating the cerebellar controller and the robot communicated through a point-to-point Ethernet connection, whilst time delays were induced at each end of the control loop (δ_{C2R} and δ_{R2C}). A second computer was added for monitoring purposes, connected to the controller through a point-to-point Ethernet connection. **(B)** Schematic of the cerebellar feedback control loop. **(C)** Depiction of the cells, neural layers, connections, and plasticity site of our cerebellar SNN torque controller. The inputs to the cerebellar network arrive through the MFs (sensorial signals) and CFs (teaching/error signal). MFs project the sensorial information onto GCs. GCs project, through the PFs, onto PCs, which also

receive excitatory inputs from the CFs. Finally, DCN drives the cerebellar output torque commands receiving excitatory inputs from MFs and CFs and inhibitory inputs from PCs, which shape the cerebellar output. The cerebellar model also implements a STDP at PF-PC connections.

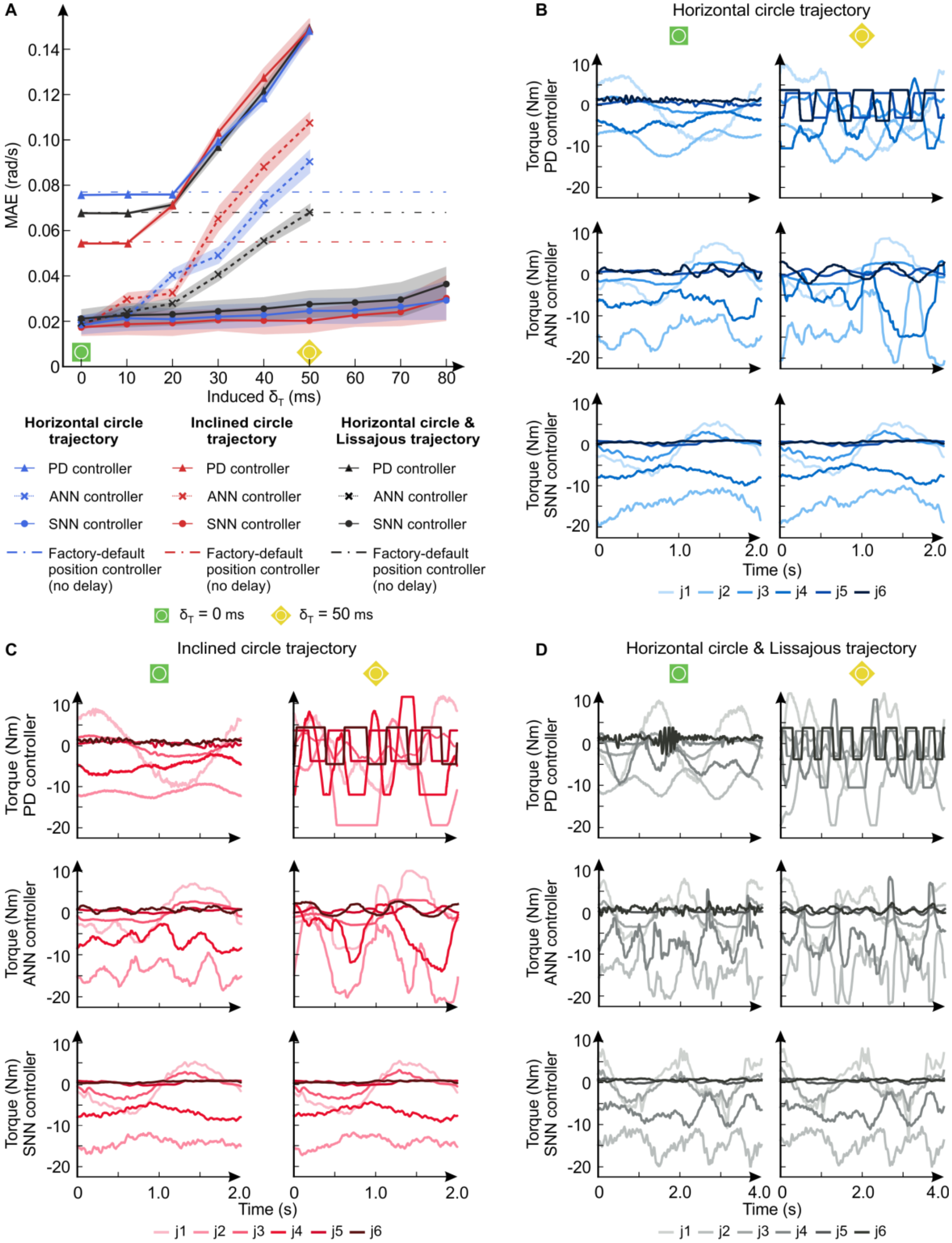

Fig. 2. PD and cerebellar ANN vs cerebellar SNN control response to steady time delays. The induced transmission delays (δ_T) comprised symmetrical R2C and C2R steady time delays ($\delta_T = \delta_{R2C} + \delta_{C2R}$; $\delta_{R2C} = \delta_{C2R}$). **(A)** As δ_T increased from 0 to 80 ms, mean MAE and standard deviation of 100 trials per δ_T value performed by the Ziegler-Nichols tuned PD, the ANN and the SNN torque controller solutions. After tuning the PD parameters, it performed similarly to the default factory position controller. ANN and SNN were both equipped with similar PF-PC synaptic

mechanisms although ANN lacked the learning temporal capability. Two circular trajectories in different planes and a sequence of a horizontal circle plus a Lissajous trajectory were used as benchmarks for revealing the robot arm dynamics (*91*, *93*). SNN controller MAE plateaued for values under δ_T = 80 ms, whereas both PD and ANN MAE should not operate above δ_T = 20 ms (for safety reasons, δ_T was kept below 50 ms for the PD and ANN controllers since the MAE was increasing dramatically). **(B)**, **(C)**, and **(D)** evolution of the output torque commands for the horizontal circle, inclined circle, and circle-Lissajous sequence respectively, for δ_T values from 0 to 50 ms (left and right column respectively).

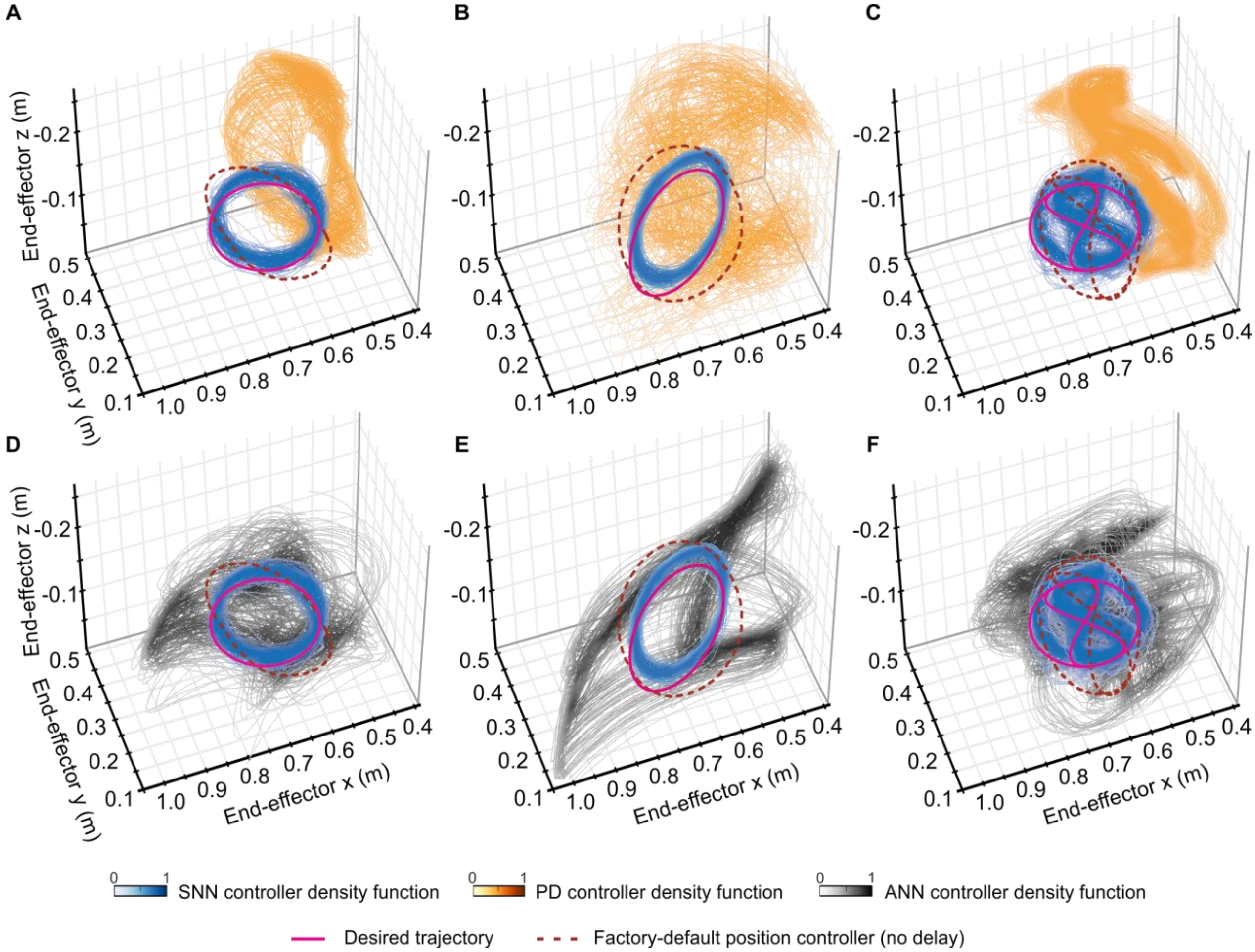

Fig. 3. Cartesian space representation of Baxter's end-effector under PD, ANN and SNN torque control. PD vs SNN performance for the horizontal circle **(A)**, inclined circle **(B)**, and circle-Lissajous sequence **(C)**. ANN vs SNN performance for the horizontal circle **(D)**, inclined circle **(E)**, and circle-Lissajous sequence **(F)**. The induced transmission delay was δ_T = 50 ms. The desired vs actual trajectory followed by the end-effector are displayed using the density function of 100 trials performed by each controller. The trajectory performed by the factory-default position controller with no delay is also displayed as a reference.

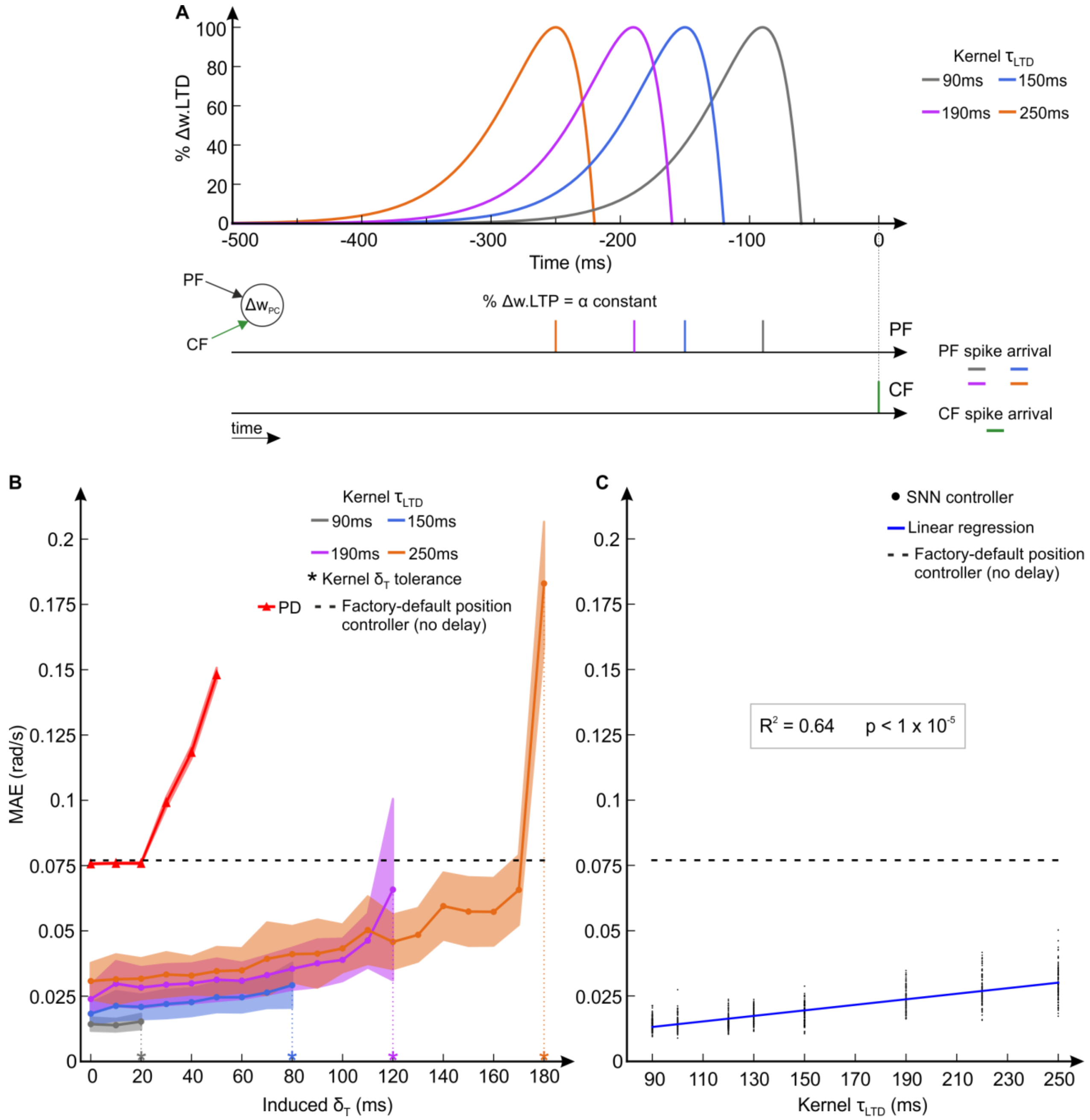

Fig. 4. Modifying the cerebellar predictive time margin by variating the STDP kernel. (A) Set of CF-PF convolution kernels with different "eligibility trace" peaks (τ_{LTD}) (*31*) and how the CF spike arrival is correlated to previous PF spike for each convolution kernel. **(B)** Performance accuracy (MAE) obtained by the SNN controller for each of the convolution kernels (τ_{LTD} peak varying from 90 to 250 ms), and PD controller reference. The transmission delay tolerance increased with τ_{LTD} peak at the cost of decreasing performance accuracy. The horizontal circle trajectory benchmark was used. The SNN technological approach overcame the τ_{LTD} = [50–150 ms] biological constraint. **(C)** Modeling the degradation of the performance accuracy as time delay tolerance increases along with the kernel τ_{LTD}. The transmission delays were set to zero, thus oversizing τ_{LTD}. A linear regression analysis was conducted on the MAE data of 100 horizontal circle trajectory trials per each of the different convolution kernels. MAE degradation seemed to linearly evolve as the τ_{LTD} peak increased (y = 0.000106x + 0.0036). Instability may arise under two possible scenarios: a) "eligibility trace" peaks shorter than transmission delays, b) oversized "eligibility trace" peaks, i.e., beyond 300 ms.

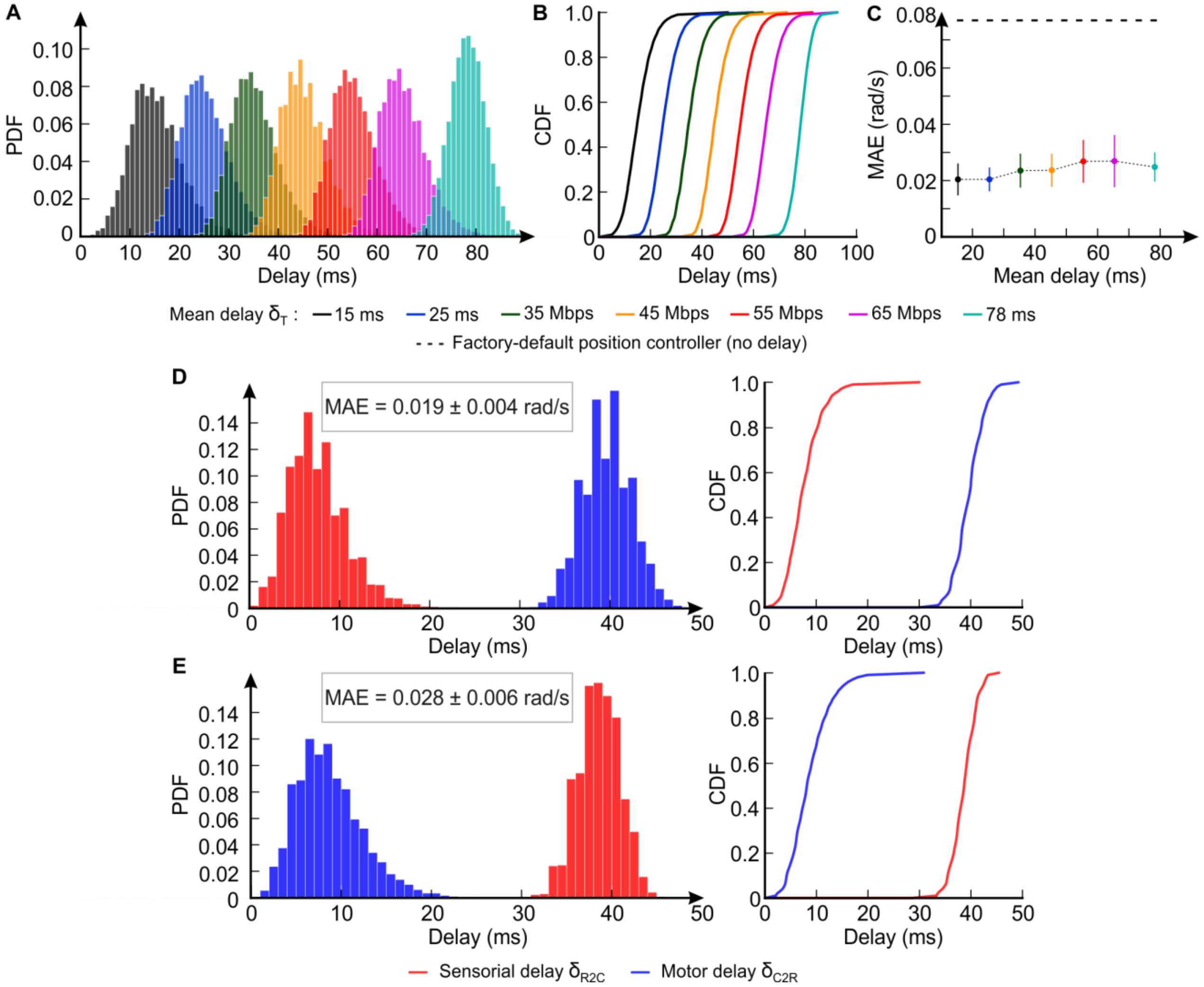

Fig. 5. Symmetric and asymmetric non-deterministic delays scenario. (A) Set of gamma distributions used to induce symmetrical ($\delta_T = \delta_{R2C} + \delta_{C2R}$; $\delta_{R2C} = \delta_{C2R} = \delta_T/2$) non-deterministic delays, **(B)** corresponding δ_T CDF, and **(C)** SNN MAE performance. **(D)** and **(E)** asymmetrical non-deterministic delays scenarios. The depicted data accounts for 100 trials of the horizontal circle trajectory per delay distribution.

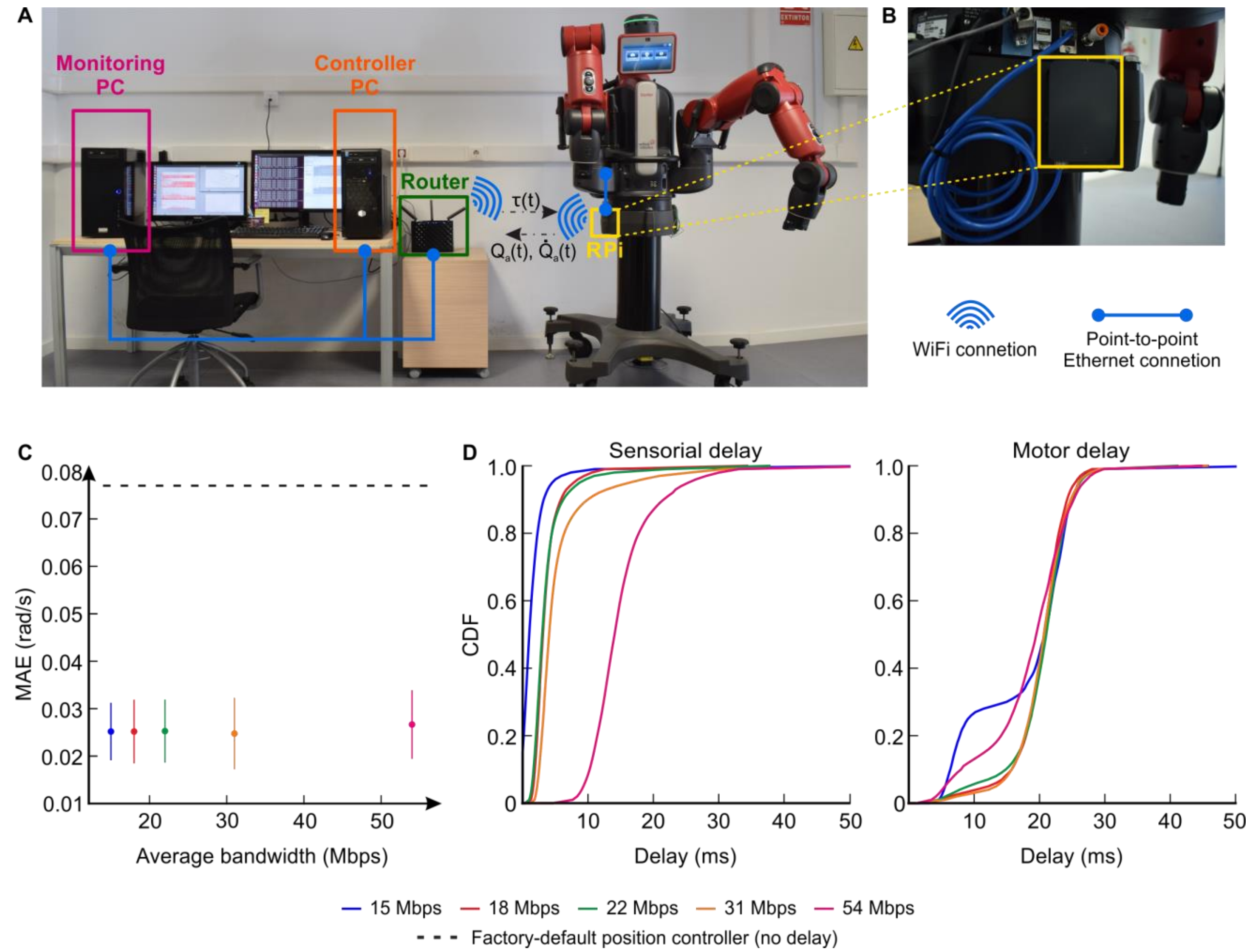

Fig. 6. Cerebellar response to non-deterministic Wi-Fi delays. (A) Experimental setup in which the computer allocating the cerebellar controller and the robot communicated through Wi-Fi. The Controller and Monitoring PCs were connected to a router, which established a Wi-Fi connection with the **(B)** RPi attached to the robot. **(C)** Performance accuracy, and **(D)** CDF of sensorial (R2C) and motor (C2R) time delays as the bandwidth consumption increased from 15 up to 54 Mbps (equivalent to three robots simultaneously connected). 100 horizontal circle trajectory trials were performed for each bandwidth value. The asymmetry between sensorial and motor delays followed the asymmetrical nature of the control loop hardware; on one end, the RPi gateway holds limited computational capacity compared to the PC on the other end. We induced the additional bandwidth in the R2C direction as the processing capacity of the RPi became saturated when additional bandwidth was induced in the C2R direction. Regardless of the asymmetrical and non-deterministic time delays, the SNN torque controller provided for compliance and accuracy.

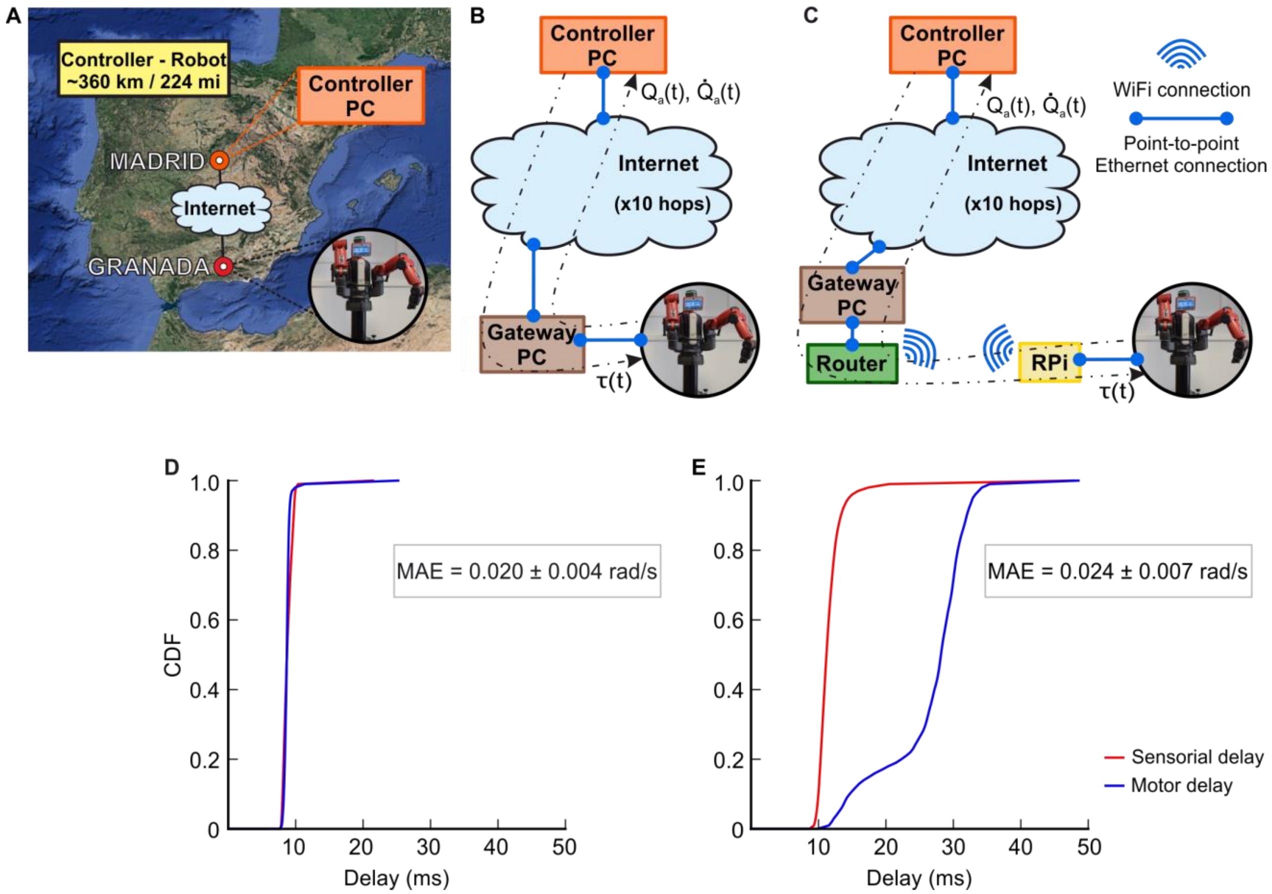

Fig. 7. Cerebellar response to remote control. (A) Experimental setup involving long-distance remote control. The robot was remotely operated over the Internet involving 10 network hops and a controller-robot distance of ~360 km (i.e., 224 mi). Two approaches were used: **(B)** the robot connected to the Internet using an Ethernet connection via a gateway PC; **(C)** the robot connected to the Internet via Wi-Fi. **(D)**, and **(E)** depict the CDF of the sensorimotor time delays associated to **(B)** and **(C)** respectively. 100 trials of the horizontal circle trajectory were performed for each approach.

Tables

Table 1. Cerebellar neural network topology.

Neurons		Synapses			
Pre-synaptic	Post-synaptic	Number	Type	Initial weight (nS)	Weight range (nS)
240 MFs	60K GCs	240K	AMPA	0.18	-
240 MFs	600 DCN	144K	AMPA	0.1	-
60K GCs	600 PCs	36M	AMPA	2.0	[0, 5]
600 PCs	600 DCN	600	GABA	1.0	-
600 CFs	600 PCs	600	AMPA	0.0	-
600 CFs	600 DCN	600	AMPA	0.5	-
600 CFs	600 DCN	600	NMDA	0.25	-

SUPPLEMENTARY MATERIALS

Annex S1. The "eligibility trace" and how it enters the learning rule equation

The most widely accepted hypothesis on motor learning cerebellar adaptation assumes that CFs spike discharges on PCs work as motor-error related signals able to drive synaptic adaptation on PFs-PCs connections. The spike-timing dependent plasticity (STDP) mechanism operating at this cerebellar layer combines a supervised long-term depression (LTD) mechanism driven by the motor-error related signal and an unsupervised long-term potentiation (LTP) mechanism that occurs even in the absence of such error signal (*94*).

LTD produces a synaptic efficacy decrease in PFs each time a PC receives a CF discharge. The amount of PFs-PCs synaptic weight decrement depends on the timing of the activity arriving through the PFs before the CF spike discharge on the same PC. This PF activity is convolved with the integrative kernel defined in Eq. 1, which only considers those PF spikes within the time-window before the CF spike discharge (*87*). The past activity of the afferent PF is evaluated similarly to a time-logged "eligibility trace," (*36*, *95*, *96*). This trace aims at correlating the relative timing between CF discharges (motor-error related activity) and the spike activity driven by the PFs (sensorimotor related activity). The eligibility trace idea stems from experimental evidence indicating the likelihood of a CF discharge to depress a PF–PC synapse when the corresponding PF fires between 50 and 150 ms before the CF discharge arrives at the same PC (*31*, *36*, *97*).

$$k\left(t-t_{CFspike}\right)=k\left(x\right)=\begin{cases}\dfrac{-\left(x+d_k\right)}{\tau_{LTD}-d_k}\cdot e^{\frac{x+d_k}{\tau_{LTD}-d_k}+1} & if\ x<-d_k\\ 0 & if\ x\geq -d_k\end{cases} \tag{1}$$

The amount of LTD produced is not constant (see the LTD kernel vs. time representation in Fig. 4), with a maximum occurring when the time difference between PFs and CFs spikes is aligned to the sensorimotor pathway delay (i.e., 150 ms). On the other hand, the inertia that results when operating a body (either a human or human-like robotic body) makes the body position and velocity at a specific moment dependent on a sequence of motor commands rather than on just the current motor command. The closer the temporal distance of a motor command in the sequence to the current time step, the greater its impact on the body state (bear in mind the propagation delay from the cerebellum to the muscle is also accounted for). The LTD kernel shapes this behavior applying the maximum LTD action in the PFs aligned with the sensorimotor delay (the ones propagating the sensorimotor information most tightly related with the "current" body state, therefore the most important for generating the necessary motion sequence), but also applying smaller LTD actions (using both kernel tails) in the PFs propagating sensorimotor information with longer and shorter sensorimotor delays, allowing the generation of a smooth movement.

Besides LTD, LTP produces a fixed increase in synaptic efficacy each time a spike arrives through a PF to the corresponding targeted PC. This mechanism aims to capture how the LTD process is reversed according to neurophysiologist studies (*98*). Both processes, LTP and LTD, are computed using Eq. 2 and 3, where $\Delta W_{PFj-PCi}(t)$ denotes the synaptic weight change between the jth PF and the target ith PC; $\alpha = 0.002$ nS is the synaptic efficacy increment; δ_{PF} is the Dirac delta function corresponding to an afferent spike from a PF; β

= -0.0008 nS is the synaptic efficacy decrement; and k(x) is the kernel function previously defined.

$$LTP\,\Delta w_{PF_j-PC_i}(t)=\alpha\cdot\delta_{PFspike}(t)\cdot dt \qquad (2)$$

$$LTD\,\Delta w_{PF_j-PC_i}(t)=\beta\cdot\int_{-\infty}^{t_{CFspike}}k\left(t-t_{CFspike}\right)\cdot\delta_{PFspike}(t)\cdot dt \qquad (3)$$

In summary, focusing on the functionality behind these mechanisms, LTD allows specifically decreasing the weights of the PC connections that received sensorimotor activity sometime before an error occurred. On the other hand, the non-specific LTP facilitates PCs to slowly recover connections from fibers carrying sensorimotor signals. Both mechanisms jointly allow reducing the error during a task as shown in the results.

Annex S2. The Leaky integrate-and-fire neuron model (LIF)

The LIF neuron model (*87*) was selected due to its minimal computational cost when generating and processing spikes, key for RT operation. The LIF neuron model elicited a single spike only when its membrane potential reached a certain threshold and, immediately after, its membrane potential was reset. The LIF neural dynamics was defined by its membrane potential and its excitatory (AMPA and NMDA) and inhibitory (GABA) chemical conductances as follows:

$$C\cdot\frac{dV}{dt}=I_{int}+I_{ext} \qquad (4)$$

$$I_{int}=-g_l\cdot\left(V+E_L\right) \qquad (5)$$

$$I_{ext}=-\left(g_{AMPA}(t)+g_{NMDA}(t)\cdot g_{NMDA\_inf}\right)\cdot\left(V-E_{AMPA}\right)-g_{GABA}(t)\cdot\left(V-E_{GABA}\right) \qquad (6)$$

$$g_{AMPA}(t)=g_{AMPA}(t_0)\cdot e^{\frac{t-t_0}{\tau_{AMPA}}}+\sum_{i=1}^{N}\delta_{AMPA_i}(t)\cdot w_i \qquad (7)$$

$$g_{NMDA}(t)=g_{NMDA}(t_0)\cdot e^{\frac{t-t_0}{\tau_{NMDA}}}+\sum_{i=1}^{N}\delta_{NMDA_i}(t)\cdot w_i \qquad (8)$$

$$g_{GABA}(t)=g_{GABA}(t_0)\cdot e^{\frac{t-t_0}{\tau_{GABA}}}+\sum_{i=1}^{N}\delta_{GABA_i}(t)\cdot w_i \qquad (9)$$

$$g_{NMDA\_inf}=1\Big/\left(1+exp\left(62\cdot V\right)\cdot\frac{1.2}{3.57}\right) \qquad (10)$$

where *C* denotes the membrane capacitance; *V* is the membrane potential; I_{int} is the internal current and I_{ext} is the external current. E_L is the resting potential and g_L the conductance responsible for the passive decay term towards the resting potential. Conductances g_{AMPA}, g_{NMDA} and g_{GABA} integrate all the contributions received by each receptor type (AMPA, NMDA, GABA) through individual synapses, being g_{NMDA_inf} the NMDA activation channel. These conductances were defined as decaying exponential functions (*87, 99*) where their values were directly incremented proportionally to the synaptic weights (w_i) upon each presynaptic spike arrival (Dirac delta functions). When the membrane potential reached a threshold (V_{thr}), it was then reset to E_L during the refractory period (T_{ref}). The configuration parameters for the neurons modeled are shown in Supplementary Materials table S1.

Annex S3. Mathematical description of the trajectories

The Cartesian space description of the horizontal circle trajectory is described by:

$$\left.\begin{aligned} x &= R\cdot\cos\left(2\cdot\pi\cdot\frac{t}{T}+\pi\right)\\ y &= R\cdot\sin\left(2\cdot\pi\cdot\frac{t}{T}+\pi\right)\\ z &= \alpha \end{aligned}\right\} t\in\left[0,2\right];\alpha = const \tag{11}$$

whilst the inclined circle trajectory is described by:

$$\left.\begin{aligned} x &= R\cdot\cos\left(2\cdot\pi\cdot\frac{t}{T}+\pi\right)\cdot\cos\left(\frac{\pi}{6}\right)\\ y &= R\cdot\sin\left(2\cdot\pi\cdot\frac{t}{T}+\pi\right)\\ z &= R\cdot\cos\left(2\cdot\pi\cdot\frac{t}{T}+\pi\right)\cdot\sin\left(\frac{\pi}{6}\right) \end{aligned}\right\} t\in\left[0,2\right] \tag{12}$$

The parametric equations of the Lissajous trajectory are:

$$\left.\begin{aligned} x &= B\cdot\sin\left(b\cdot 2\cdot\pi\cdot\frac{t}{T}\right)\\ y &= A\cdot\sin\left(a\cdot 2\cdot\pi\cdot\frac{t}{T}+\delta\right)\\ z &= \alpha \end{aligned}\right\} \left.\begin{aligned} x &= \frac{R}{2}\cdot\sin\left(4\cdot\pi\cdot\frac{t}{T}\right)\\ y &= R\cdot\cos\left(2\cdot\pi\cdot\frac{t}{T}\right)\\ z &= \alpha \end{aligned}\right\} t\in\left[0,2\right];\alpha = const \tag{13}$$

where R = 12 cm denotes the circle radius, T = 2 s stands for the trajectory duration. The Cartesian space trajectories were then translated to joint space using *Moveit!* software (*100*), thus obtaining the desired position (Q_d) for each of the six DOF. The desired joint velocity profiles ($\dot{Q}_d$) were obtained as the desired position derivative over time; thus completing the desired trajectory input signals (Q_d, $\dot{Q}_d$).

Annex S4. Robot-Controller Wi-Fi gateway

To establish the Wi-Fi connection between the robot and the controller we had to circumvent Baxter's lack of wireless support. We attached a Raspberry Pi 3B+ (RPi) to the robot using an Ethernet connection with negligible delay (δ_{Eth} ~ 0 ms). The RPi, in turn, connected with the controller via a Wi-Fi connection which carried inherent non-deterministic time delays ($\delta_{\text{Wi-Fi}}$). Thus, the RPi operated as a robot-controller gateway establishing an end-to-end Wi-Fi communication with non-deterministic time delays ($\delta = \delta_{\mathrm{Eth}}+ \delta_{\text{Wi-Fi}} \sim \delta_{\text{Wi-Fi}}$) (Fig. 4, A and B). The Wi-Fi connection was established using a Tenda® AC15 AC1900 Smart Dual-band Gigabit Wi-Fi Router.

Annex S5. Induction of additional Wi-Fi bandwidth

To modify the non-deterministic Wi-Fi time delays, we induced additional UDP traffic to the control loop end-to-end communication using the tool *Iperf* (*101*). We gradually increased the original bandwidth consumption from 15 to 54 Mbps in the R2C direction since the processing capacity of the RPi rapidly became saturated when additional traffic was induced in the C2R direction. The processing of additional incoming information jeopardized the RPi ability as robot-controller gateway. The controller PC processing capability, however, was not affected by the additional traffic. The asymmetrical hardware of the control loop forced us to induce the additional bandwidth in the R2C direction, which was reflected in asymmetrical Wi-Fi non-deterministic time delays (Fig. 4C). The cost of the RPi acting as a bottleneck could be saved if access to Baxter's onboard PC were granted or other more powerful nodes were used instead of the RPi.

Annex S6. The ANN cerebellar model

The ANN cerebellar model adopted a pure rate-based functional scheme. The focus was on maintaining the functional information processing features of the cerebellar micro-circuitry using analog activity values instead of an explicit spiking representation (*102*).

We implemented four main layers:

- Granular layer: implemented as a state-generator able to provide for different time stamps along the executed trajectory (*103, 104*) depending on the actual and desired joint positions and velocities. These time stamps emulate parallel fibers (PFs) activated in an unambiguous and sequential manner (producing an unambiguous state representation).

- Purkinje-cell layer: the activity at Purkinje cells (PCs) is defined in Eq.14:

$$PC_i\left(t\right)=f_i\left(PF\left(t\right)\right),\quad where\ i\in\left\{1,2,\ldots,number\ of\ motors\right\} \tag{14}$$

where *PC_i(t)* represents the average firing rate of the PCs associated with the i^{th} motor. f_i is the function that matches each granular layer state (active PF) with a particular output firing rate at each PC. This function was modified during the learning process. The output activity at different cell layers (PCs, MFs and CFs) was normalized between 0 (representing the absence of activity) and 1 (representing the maximum firing rate of the cell).

- Mossy fibers: the ANN cerebellar model assumes mossy fibers (MFs) transmitting a baseline neural activity during the trajectory execution according to studies of eyeblink conditioning experiments (*105-107*).

- DCN cells: the activity of these nuclei cells integrated the excitatory-activity coming from MFs and CFs and the inhibitory-activity from PCs. Due to the low number of MFs and CFs in comparison to granule cells (GrCs), the capacity of these fibers for generating a sparse representation of different cerebellar states seems to be very limited (i.e., MFs act as baseline global activity/term provider). Eq. 15 describes the DCN layer behavior:

$$\begin{gathered}DCN_i\left(t\right)=MF_i(t)\cdot w_{MF-DCN,i}-PC_i\left(t\right)\cdot w_{PC-DCN,i}+CF_i\left(t\right)\cdot w_{CF-DCN,i},\\ where\ i\in\left\{1,2,\ldots,Number\ of\ motors\right\}\end{gathered} \tag{15}$$

$DCN_i(t)$ represents the average firing rate of the DCN cells associated with the i^{th} motor, $MF_i(t)$ stands for the baseline activity of the MFs associated to the i^{th} motor, and $W_{MF\text{-}DCN,\mathrm{i}}$ the synaptic strength of the MF-DCN connection to the i^{th} motor. $W_{PC\text{-}DCNi}$ represents the synaptic strength of the PC-DCN connection of the i^{th} motor. Finally, $CF_i(t)$ represents the average firing rate of the CFs associated with the i^{th} motor, being $W_{CF\text{-}DCNi}$ the synaptic strength of the CF-DCN of the associated motor. $CF_i(t)$ carries the normalized current activity in the range [0, 1] that represents the actual motor error.

$$\varepsilon(t) = \left(q_{desired}(t) - q(t)\right) + \left(\dot{q}_{desired}(t) - \dot{q}(t)\right)$$
$$CF_i(t) = \frac{\varepsilon_i(t)}{\varepsilon_{\max_i}} \in [0,1] \quad where\ i \in \{1, 2, \ldots, Number\ of\ motors\} \qquad (16)$$

PF-PC long-term synaptic plasticity

Following on from our previous articles (*33*), the present model implements PF-PC synaptic plasticity as follows:

$$\Delta w_{PF_j - PC_i}(t) = \begin{cases} \dfrac{LTP_{Max}}{\left(CF_i(t) + 1\right)^{\alpha}} - LTD_{Max} \cdot CF_i(t), if\ PF_j\ is\ active\ at\ t \\ 0 \quad otherwise \end{cases} \qquad (17)$$

$$where\ i \in \{1,2,\ldots, Number\ of\ motors\}$$

where $\Delta w_{PF_j - PC_i}(t)$ represents the weight change between the j$^{\mathrm{th}}$ PF and the target PC associated with the i$^{\mathrm{th}}$ motor. $CF_i(t)$ stands for the current activity coming from the associated climbing fiber (which represents the normalized error along the executed arm plant movement), LTP_{Max} and LTD_{Max} are the maximum long term potentiation/long term depression (LTP/LTD) values, and α is the LTP decaying factor. In the experiments α is set to 1000 to ensure a fast LTP action decreasing (*33*).

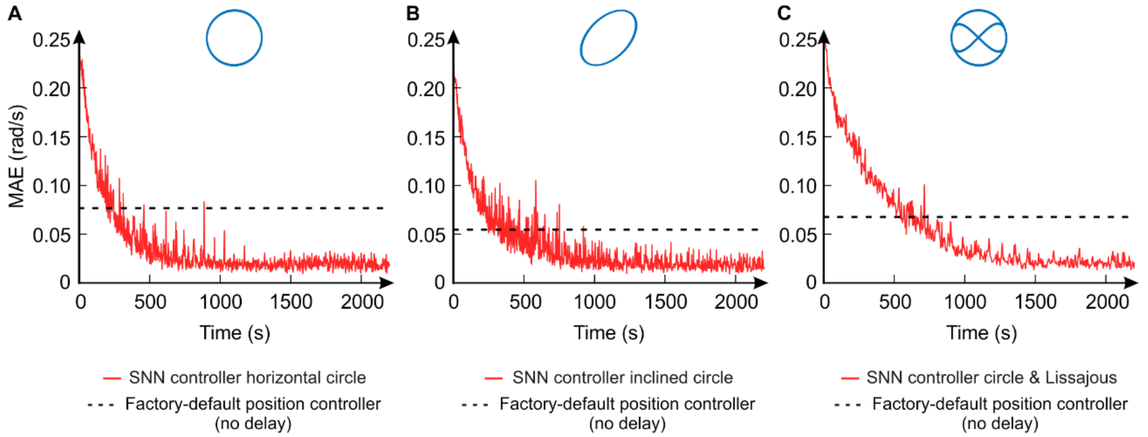

Fig. S1. Trajectory learning convergence curves. (A) Circle trajectory in xy plane, trajectory duration of 2 s. Learning stabilization achieved after about 1000 s (500 trials). (B) Inclined circle trajectory in xyz plane, trajectory duration of 2 s. Learning stabilization achieved after about 1000 s (500 trials). (C) Concatenated circle and Lissajous trajectory in xy plane, trajectory duration of 4 s (2 s circle + 2 s Lissajous). Learning stabilization achieved after about 2000 s (500 trials).

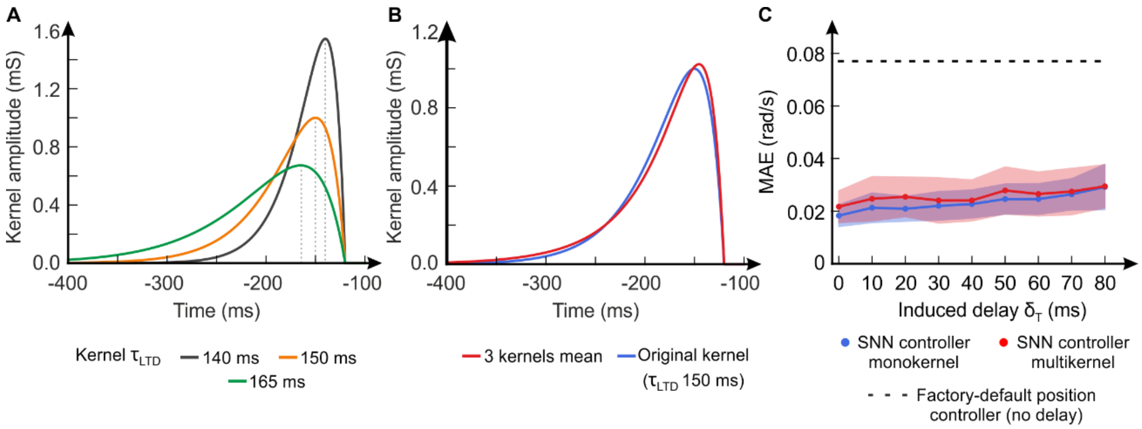

Fig. S2. Cerebellar SNN multikernel vs monokernel solution coping with time delays. **(A)** LTD eligibility traces of the multikernel solution. **(B)** Mean eligibility trace of the multikernel solution and eligibility trace of the monokernel solution. **(C)** Performance accuracy for the horizontal circle trajectory, both SNNs operated in the scenario depicted in Fig. 1A, with induced delays δ_T from 0 to 80 ms, using a unique robotic sensorimotor pathway delay (150ms). The multikernel solution required a larger cerebellar network (1800 PCs and 108M of plastic synapses) to maintain equivalent levels of output resolution when compared to the monokernel solution (600 PCs and 36M synapses), i.e., extra computational power hindering full capacity performance in RT. The EDLUT simulator ran at full capacity for 99.999% of the experiment time for the monokernel solution, reduced to 99.263% for the multikernel solution; i.e., since RT operation needs to be guaranteed in the control loop, EDLUT includes mechanisms to minimize the impact of higher computing intervals, such as temporarily disabling learning for the sake of RT operation (*57*). The multikernel overall response was configured to provide a PC output drive equivalent to the monokernel PC solution but preserving the enhanced delay sensitivity to its corresponding kernel peaks, similar to what it is found at the cerebellar vermis (*78*). The larger number of the multikernel PC outputs coalesced into the same number of DCNs for the monokernel network (*108*). Multikernel PC outputs hampered co-operation among themselves under time delays (range from 0 to 80 ms) thus decreasing instead of increasing the performance accuracy thanks to augmented delay sensitivity. A winner-take-all PC output behavior might take advantage of the increased delay sensitivity provided by the different kernel peaks of the multikernel solution but only if different robotic sensorimotor pathways were required to be conjointly used.

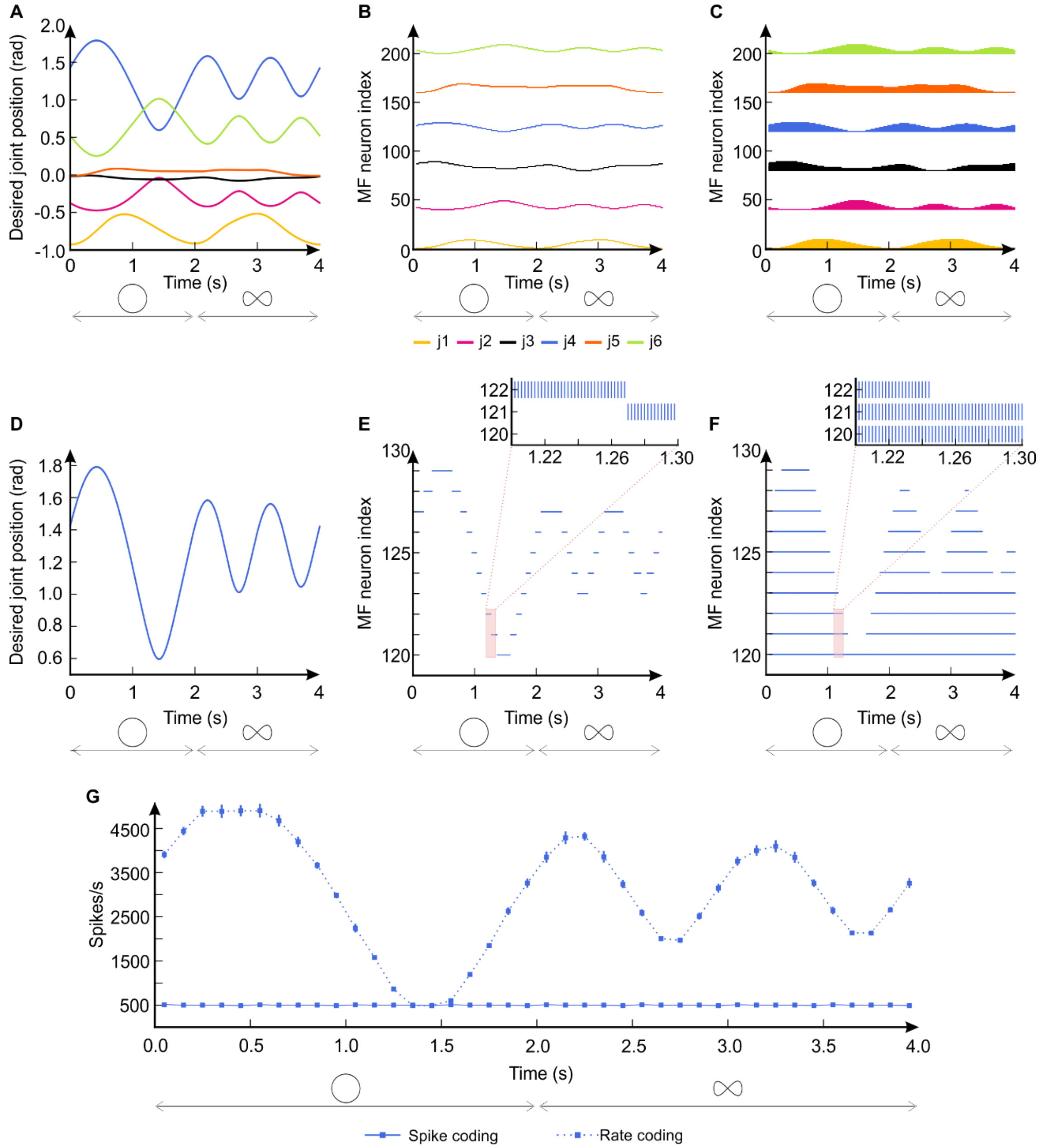

Fig. S3. Spike coding at the input MF layer. (A) All joints desired position for the circle-eight sequence, i.e., input analog signal. Corresponding spiking activity at the MFs implemented by the SNN model **(B)**, and by a possible rate based model **(C)**. **(D)**, **(E)**, and **(F)** depict a zoom in to the fourth joint (j4) information represented in **(A)**, **(B)**, and **(C),** respectively. **(G)** shows the population firing rate (MFs corresponding to j4) for time windows of 100 ms during the trajectory period (4s: 2s for circle + 2s for eight-like trajectory), both for our spike-coding SNN model (solid line), and a rate-based model (dashed line). The firing rate depicts the average of 100 trajectory trials.

Table S1. Neuron parameter values

Parameters	GC	PC	DCN
C_m (pF)	2.0	100	2.0
G_L (nS)	1.0	6.0	0.2
E_L (mV)	-65.0	-70	-70.0
E_{AMPA} (mV)	0.0	0.0	0.0
E_{GABA} (mV)	–	–	-80.0
τ_{AMPA} (ms)	1.0	1.2	0.5
τ_{NMDA} (ms)	–	–	14.0
τ_{GABA} (ms)	–	–	10.0
V_{thr} (mV)	-50.0	-52.0	-40.0
T_{ref} (ms)	1.0	2.0	1.0

Movie S1. Remote Wi-Fi cobot control. Experimental setup in which the controller PC and the Baxter robot communicate through a Wi-Fi connection. The controller PC (allocating the cerebellar controller) and the monitoring PC are connected via point-to-point Ethernet to a router, which establishes a Wi-Fi connection with a RPi (i.e., gateway) attached to the Baxter robot. The sensorial (R2C) and motor (C2R) time delays, as well as the desired and on-going trajectory are represented whilst the robot moves (i.e., horizontal circular trajectory). The recording depicts the adaptability, accuracy and the safe responsiveness to unexpected interactions and time delays of the cerebellar controller under several physical HRIs.