ARTIFICIAL INTELLIGENCE

# A cerebellar-based solution to the nondeterministic time delay problem in robotic control

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



**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. 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.**



## 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 contributing to diverse endeavors such as lightening human labor (*1*), providing medical assistance (*2*), and assisting humans in space exploration (*3*). 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 shown 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 nondeterministic 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 with 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*), and (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 nonlinear due to soft or elastic components (*4*). Regarding the second category, it includes the dead-time compensators (DTCs) (*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*), and (ii) the finite spectrum assignment approach (*18*, *19*). However, DTC solutions' strong dependence on the accuracy of the system model (*11*) makes them nonreliable for HRI control, because the growing use of flexible joints and elastic materials (*20*, *21*) makes the mathematical modeling of cobots nonlinear dynamics intractable (*22*).

These solutions prove the effort devoted to compensating for time delays in control systems. Here, we enlarge the family of solutions by taking inspiration from millions of years of biological evolution, through 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, there exists a variable delay accounting for the time spent since a motor command

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

is generated and propagated to the muscles (efferent delay $\delta_e$) until its effect is sensed back at the cerebellum (afferent delay $\delta_a$). These sensorimotor delays range from about 100 to 150 ms, with inter- and intraindividual variations (*25*). To compensate for them, the cerebellum acquires internal representations of the sensorimotor transformations needed to generate the motor commands required to achieve a desired movement (*26*) and generates predictive motor commands through 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 because they model the transfer and processing of information as it occurs in their biological counterparts, by means of the precise timing of spikes (*28*) that efficiently embed accurate timing. Thus, our cerebellar-like SNN controller adopts the biological delays and mimics the cerebellar STDP mechanism.

In the following sections, we evaluate the performance of our SNN controller under time delays of different nature: steady and nondeterministic delays in both laboratory-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 (DOFs) of the robot arm acting on a trial-and-error basis. An 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 a 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, *j*1 to *j*6), 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 ($\varepsilon$) per joint obtained comparing the actual robot state with 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, $\tau$, 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 ($\delta_T$) in the sensorimotor pathway to test whether our SNN inherits the cerebellar natural ability to deal with nondeterministic 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 ($\delta_C$) inherent to the SNN computation.

Our SNN controller reproduced the main properties of the cerebellar circuit and consisted of 62,040 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 60,000 granule cells (GCs). GCs expanded the coding space of MFs (*30*), and this GC 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 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.

### 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 $\delta_{R2C}$ before being sent to the controller, whereas at the controller side the buffer held the motor commands for a time $\delta_{C2R}$ 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 was 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 $\delta_T$ was maintained for 100 trials and then increased to the next value, that is, at least 200 s of experiment duration per $\delta_T$ value. The performance metric given by the position mean absolute error (MAE) illustrated the learning convergence of the SNN torque controller (see Materials and Methods) across a wide range of induced delays $\delta_T$ (Fig. 2A). Note that the SNN torque controller, regardless of the induced delay $\delta_T$, improved the performance accuracy (MAE) of the factory-default position controller given under no-delay circumstances.

The factory-default position controller could not be tested in a time delay framework; hence 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 versus factory-default MAE = 0.077 rad/s for the horizontal circle trajectory, PD MAE = 0.054 rad/s versus 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 rate-based artificial neural network (ANN) controller that was devoid of spatio-temporal resolution. We used the rate-based cerebellar solution from (*33*, *34*) conveniently adapted for Baxter's six DOFs 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 (LTD) was heterosynaptically driven by CF,

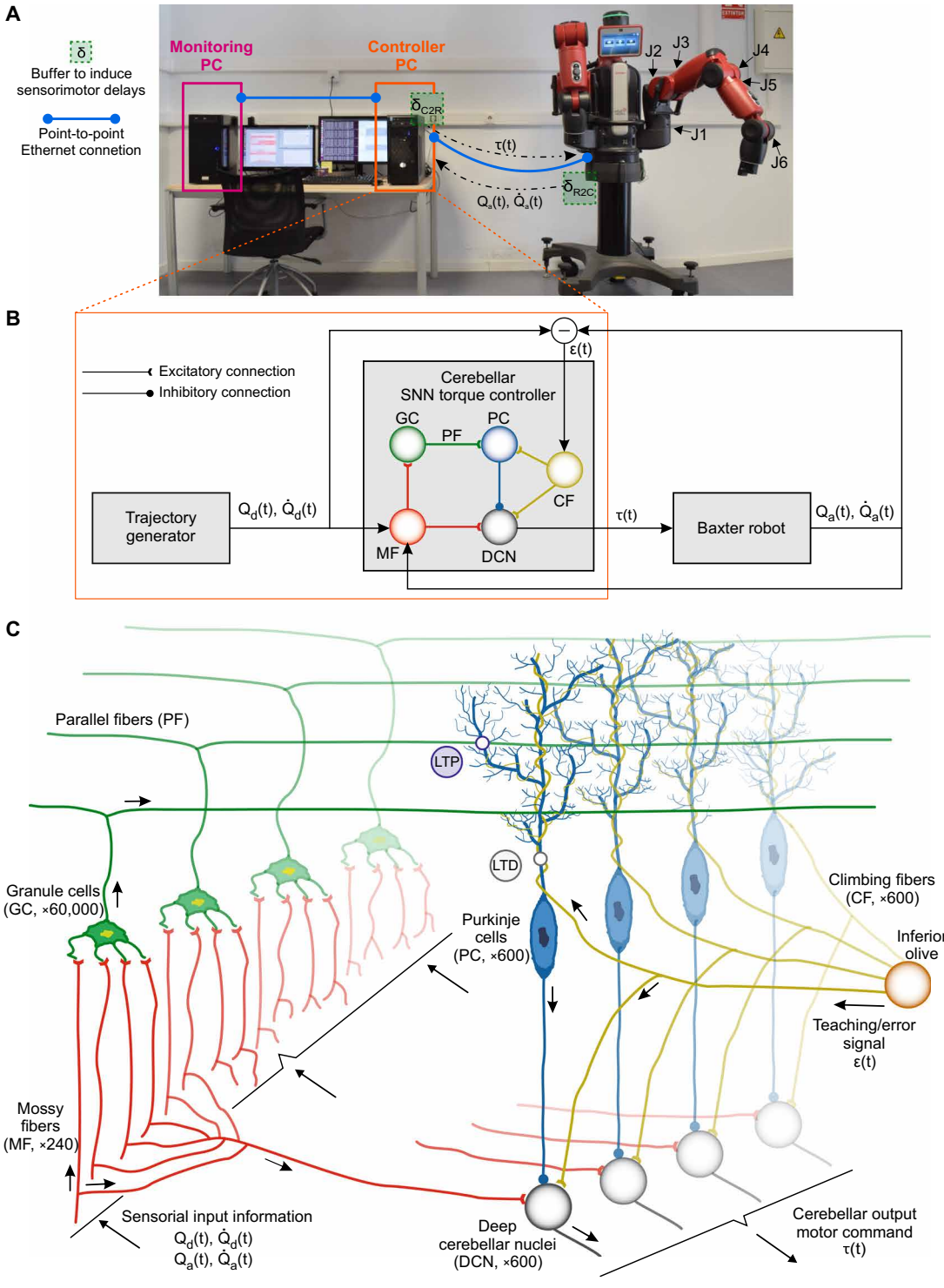


**Fig. 1. Cerebellar control loop.** (**A**) Experimental setup in which communication time delays were artificially induced within the cerebellar control loop. The computer allocating the cerebellar controller and the robot communicated through a point-to-point Ethernet connection, whereas time delays were induced at each end of the control loop ($\delta_{C2R}$ and $\delta_{R2C}$). 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. Last, DCN neurons drive 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 an STDP mechanism at PF-PC connections.

whereas PC long-term potentiation (LTP) 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 versus SNN MAE = 0.018 ± 0.004 rad/s for the horizontal circle trajectory, ANN MAE = 0.017 ± 0.002 rad/s versus SNN MAE = 0.017 ± 0.004 rad/s for the inclined circle trajectory, and ANN MAE = 0.019 ± 0.001 rad/s versus SNN MAE = 0.021 ± 0.004 rad/s for the circle-Lissajous sequence (Fig. 2A).

As the induced delay $\delta_T$ increased from 0 to 50 ms, the PD and ANN controllers performance degraded substantially (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, and 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 to D). PD control instability

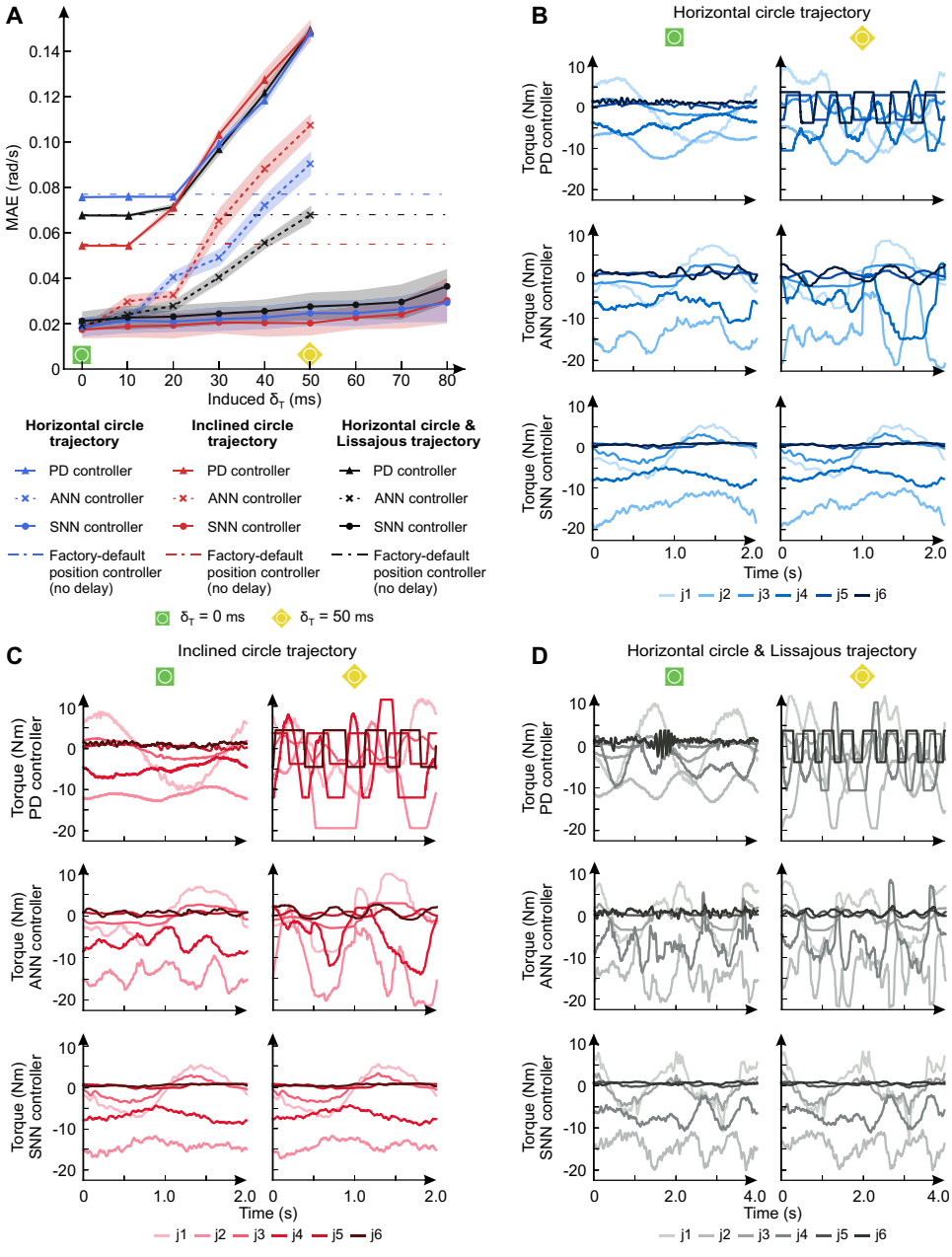


**Fig. 2. PD and rate-based cerebellar ANN versus cerebellar SNN control response to steady time delays.** The induced transmission delays ($\delta_T$) comprised symmetrical R2C and C2R steady time delays ($\delta_T = \delta_{R2C} + \delta_{C2R}$; $\delta_{R2C} = \delta_{C2R}$). (**A**) As $\delta_T$ increased from 0 to 80 ms, mean and SD of the position MAE for 100 trials per $\delta_T$ 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 $\delta_T = 80$ ms, whereas both PD and ANN MAE should not operate above $\delta_T = 20$ ms (for safety reasons, $\delta_T$ was kept below 50 ms for the PD and ANN controllers because the MAE was increasing markedly). (**B** to **D**) Evolution of the output torque commands for the horizontal circle, inclined circle, and circle-Lissajous sequence, respectively, for $\delta_T$ values from 0 to 50 ms (left and right column, respectively).

occurred from early stages: delays $\delta_T$ 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 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 while maintaining sensitivity low is required. An in crescendo sensitivity may ultimately induce instability (oscillatory PD responses) and compromise compliance with lower delay $\delta_T$ values. Similarly to the PD, the ANN controller was driven to instability with delays $\delta_T$ 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 $\delta_T$ = 50 ms because 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 $\delta_T$ 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, three to four 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, and 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 to 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 $\delta_T$ was limited to 80 ms according to the predictive time margin of the deployed learning mechanism (see Materials and Methods).

The SNN torque controller outperformed the PD and ANN controllers in the Cartesian space execution of the trajectories (Fig. 3). Comparative ANN versus SNN results indicated that the time-related capability of the SNN form of synaptic plasticity accounted for coping with the delay.

### Adjusting STDP at PF-PC allows increasing the cerebellar tolerance to time delays

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 while performing the horizontal circle trajectory. We first adjusted the STDP learning mechanism to cope with the biological sensorimotor delay and 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 evaluated the STDP learning mechanism against 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 ($\Delta_W$) occurred after an appropriate temporal sequence of PF-CF de/activations, involving two opposed processes of long-lasting modifications in synaptic strength: LTP and 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 ($\tau_{LTD}$), which established the PF spike arrival time before a CF spike arrival for which the synaptic weight decrement was maximal. By changing $\tau_{LTD}$, the predictive time margin could be accordingly modified (Fig. 4, A and B). Consequently, $\tau_{LTD}$ 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 $\tau_{LTD}$ 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*) showed an LTD contribution more acute for those PF spikes that occurred 50 to 150 ms before the CF activity, i.e., $\tau_{LTD}$ between 50 and 150 ms. We chose $\tau_{LTD}$ = 150 ms to increase the time delay tolerance while 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 command application by the robot actuators. Please see Materials and Methods and annex S1 for a more in-depth description of the temporal kernel operation.

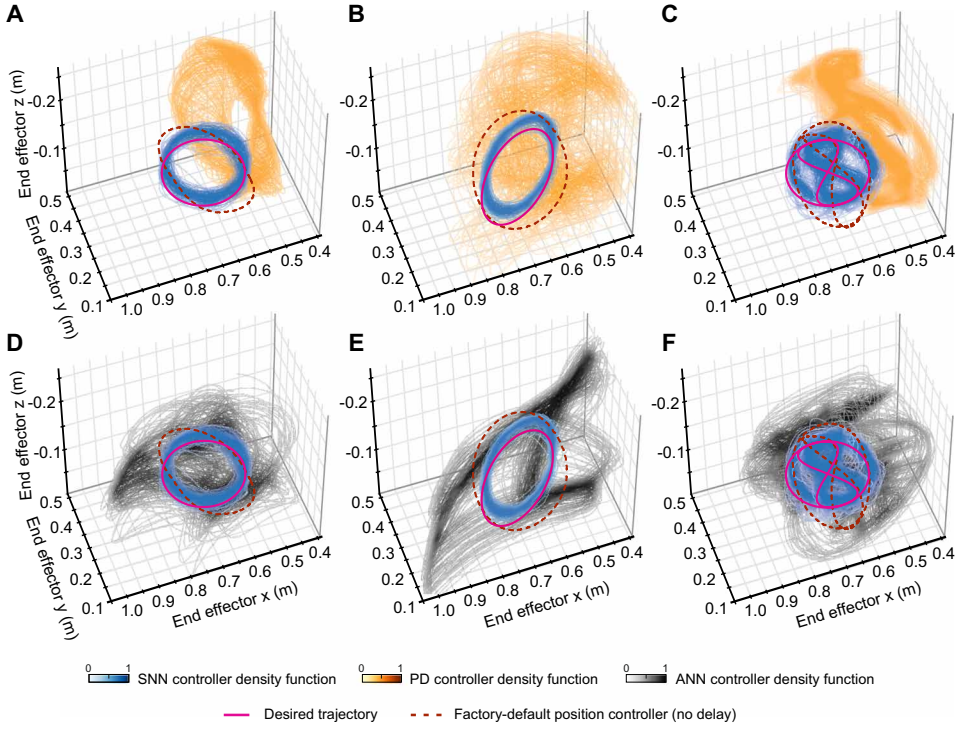


**Fig. 3. Cartesian space representation of Baxter's end effector under PD, rate-based ANN, and SNN torque controllers.** PD versus SNN performance for the horizontal circle (**A**), inclined circle (**B**), and circle-Lissajous sequence (**C**). ANN versus SNN performance for the horizontal circle (**D**), inclined circle (**E**), and circle-Lissajous sequence (**F**). The induced transmission delay was $\delta_T$ = 50 ms. The desired versus actual trajectory followed by the end effector is 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.

### Benchmarking the nondeterministic time delays

The learning convergence of our SNN output against steady time delays was tested so far; convergence under nondeterministic time delays was still to be analyzed. Here, we characterized the response of our SNN to nondeterministic delays in a laboratory-controlled scenario. The delay range (from 0 to 80 ms) was covered with a set of gamma distributions from which nondeterministic time delays $\delta_T$ were randomly sampled

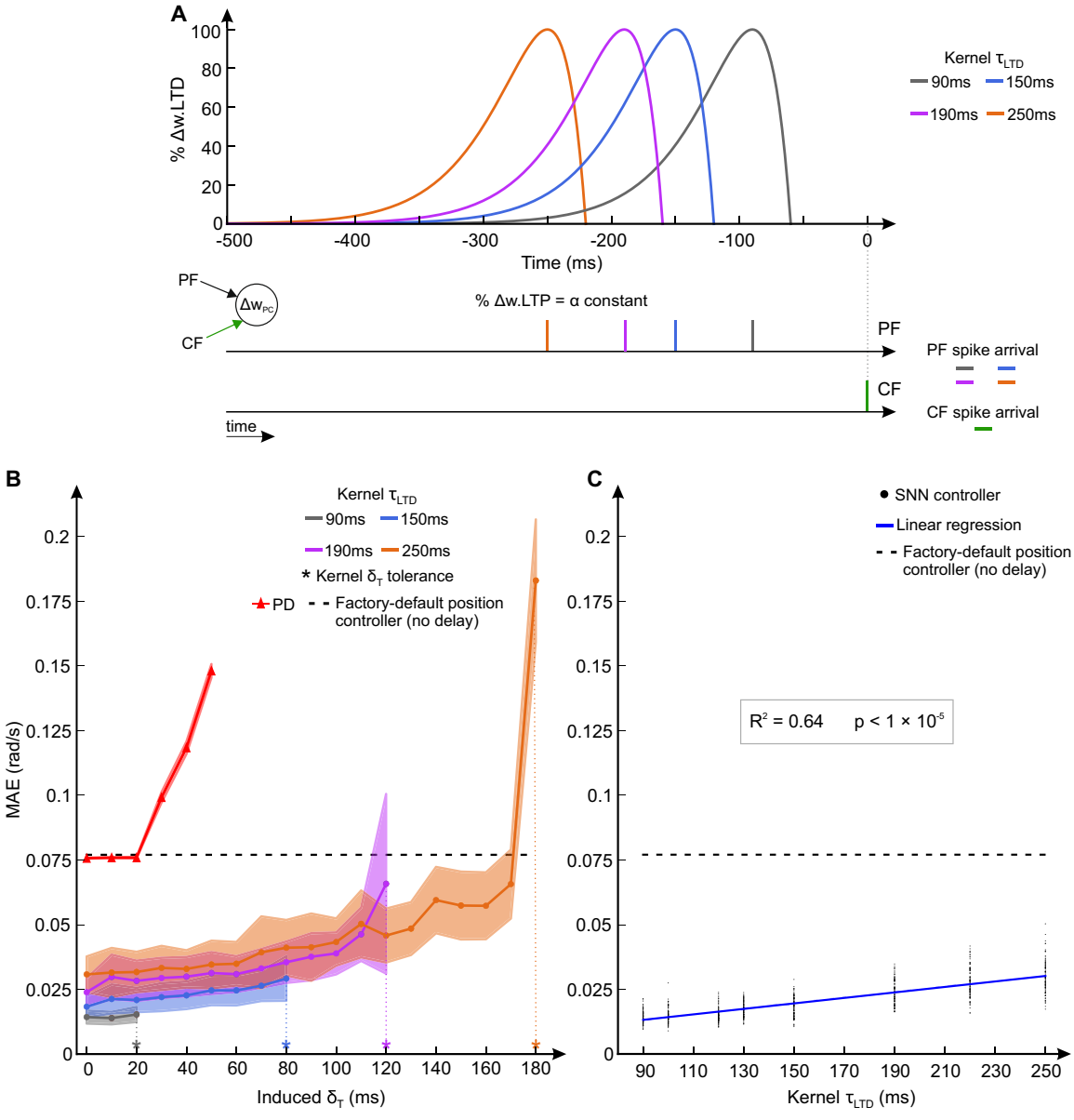


**Fig. 4. Modifying the cerebellar predictive time margin by adjusting the STDP kernel.** (**A**) Set of CF-PF convolution kernels with different eligibility trace peaks ($\tau_{LTD}$) (*31*) and how the CF spike arrival is correlated to previous PF spike for each convolution kernel. (**B**) Performance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

($\delta_T = \delta_{R2C} + \delta_{C2R}$; $\delta_{R2C} = \delta_{C2R} = \delta_T/2$), providing the following mean delays: 15 ± 5 ms, 25 ± 5 ms, 35 ± 5 ms, 45 ± 5 ms, 55 ± 5 ms, 65 ± 5 ms, 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]. Nondeterministic delays were induced using the setup 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 can adequately model network delays (*39*, *40*).

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

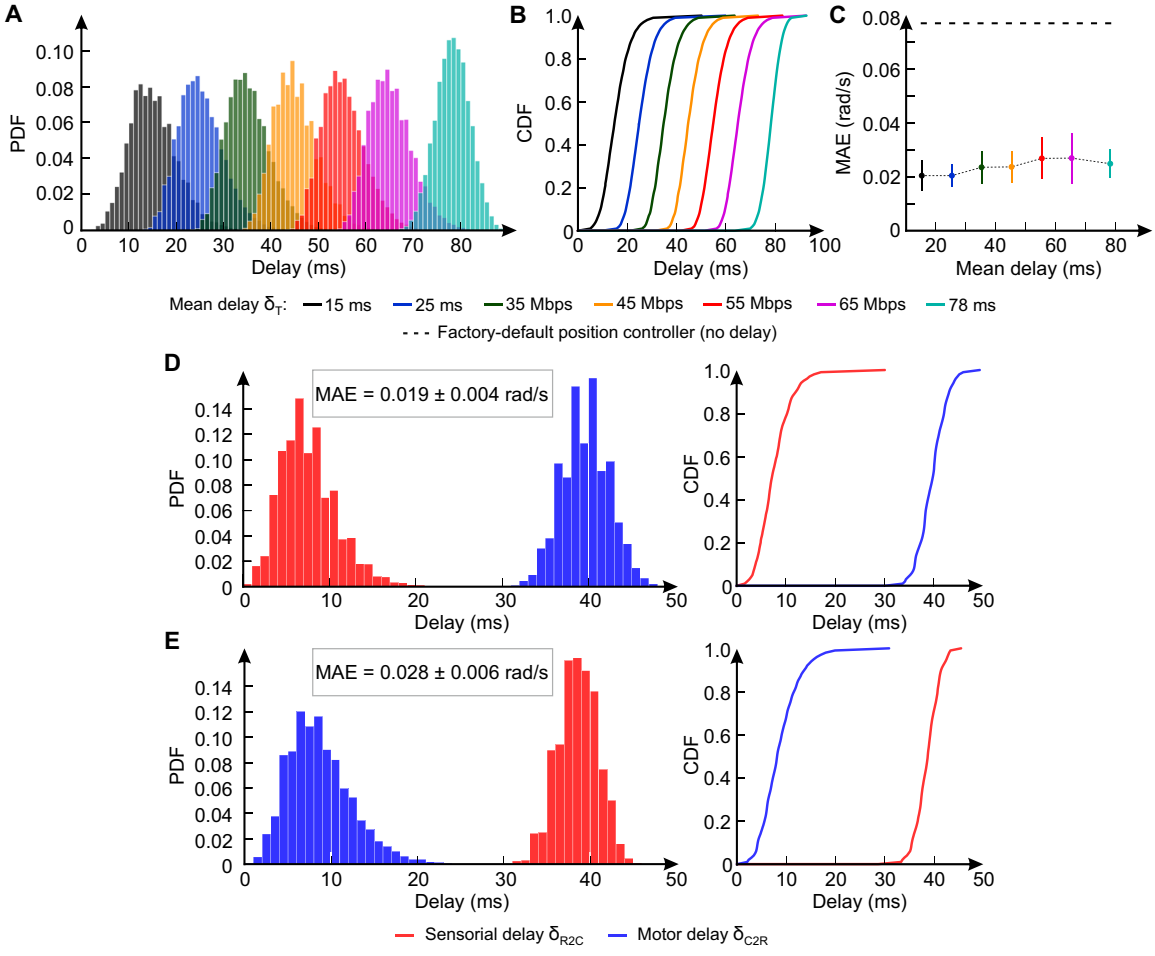


**Fig. 5. Symmetric and asymmetric nondeterministic delays scenario.** (**A**) Set of gamma distributions used to induce symmetrical ($\delta_T = \delta_{R2C} + \delta_{C2R}$; $\delta_{R2C} = \delta_{C2R} = \delta_T/2$) nondeterministic delays described by $\delta_T$ PDF and (**B**) $\delta_T$ CDF. (**C**) SNN position MAE performance. The depicted data account for 100 trials of the horizontal circle trajectory per delay distribution: mean and SD of the MAE for 100 trials. (**D** and **E**) Asymmetrical nondeterministic delay scenarios.

## Nondeterministic 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 the Supplementary Materials). The nondeterministic delays inherent to a Wi-Fi connection (*41*) affected asymmetrically to both R2C and C2R directions while 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 the 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 nondeterministic 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 with the 0.024 ± 0.011 rad/s obtained at the artificial delays scenario with $\delta_T$ from of 0 to 80 ms. The PD and ANN controllers could not be tested under these circumstances because 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.

Last, 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 miles) 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) and (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 roundtrip 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

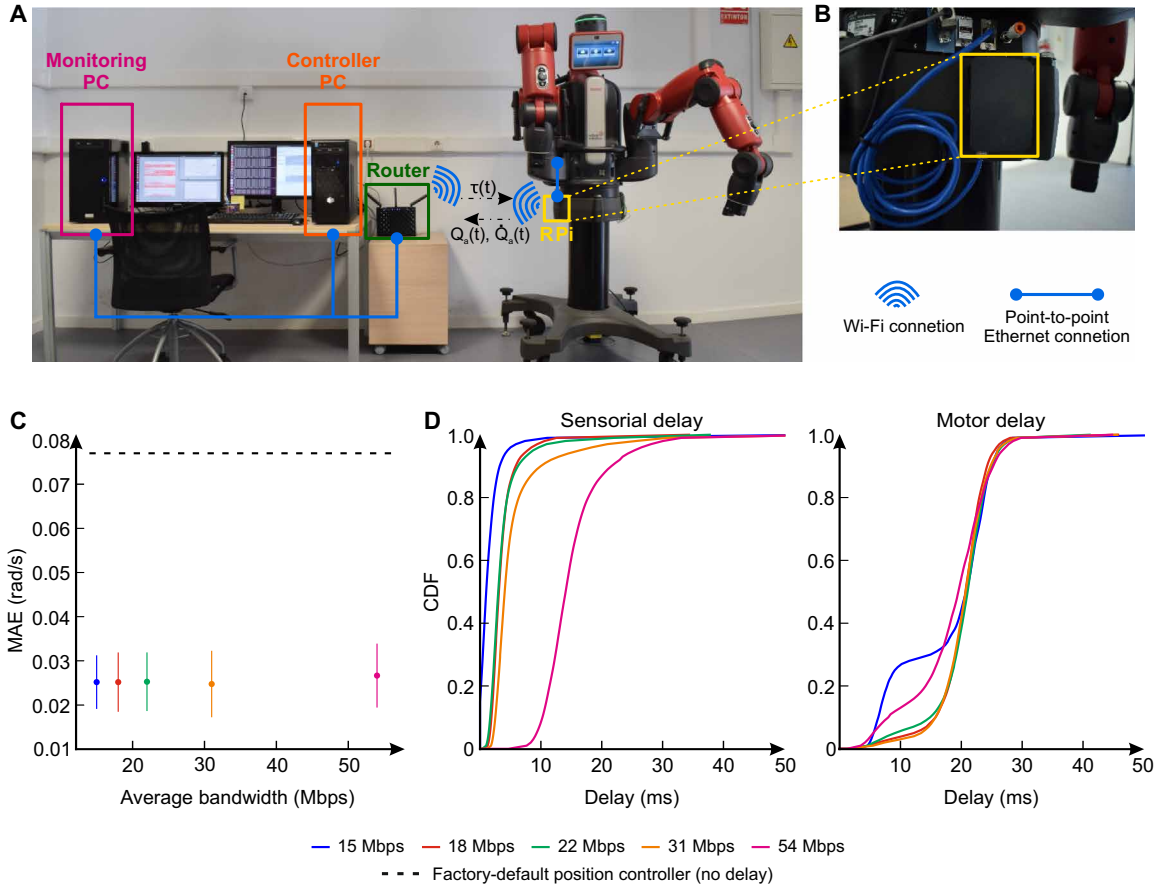


**Fig. 6. Cerebellar response to nondeterministic Wi-Fi delays.** (**A**) Experimental setup in which the computer allocating the cerebellar controller and the robot communicated through Wi-Fi. The controller and monitoring PCs were connected to a router, which established a Wi-Fi connection with the (**B**) RPi attached to the robot. (**C**) Performance accuracy: mean and SD of the position MAE for 100 trials, and (**D**) CDF of sensorial (R2C direction) and motor (C2R direction) time delays as the bandwidth consumption increased from 15 up to 54 Mbps (equivalent to three robots simultaneously connected). One hundred 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 with 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 nondeterministic time delays, the SNN torque controller provided for compliance and accuracy.

Wi-Fi nondeterministic 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 (first and second 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 shown to be capable of operating in a cloud-robotics framework.

## DISCUSSION

A well-timed response to stimuli is imperative for body interaction with changing environments, thus causing human motor control to compensate for the substantial time delay between the sensing of a stimulus 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 stimulated 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 rate-based cerebellum-inspired approaches recently suggested. (i) A rate-based cerebellar-like functional model embedded

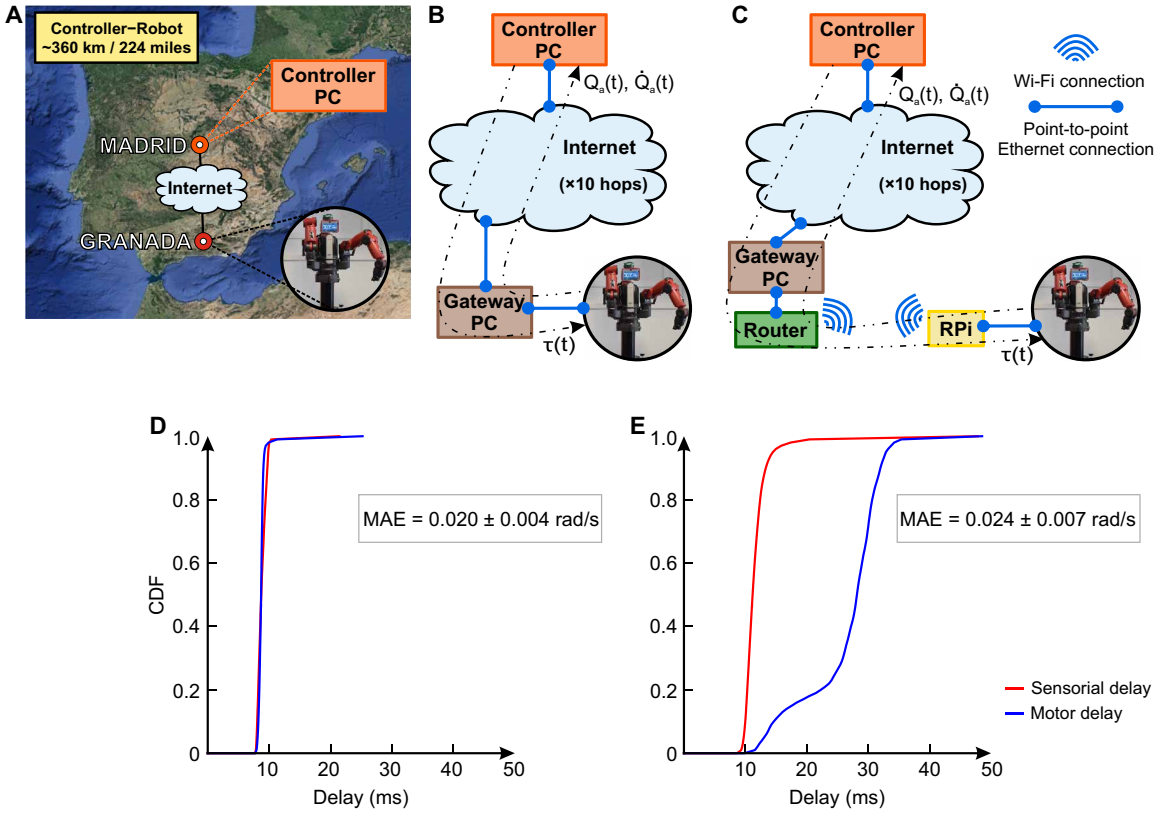


**Fig. 7. Cerebellar response to remote control.** (**A**) Experimental setup involving long-distance remote control. The robot was remotely operated over the Internet involving 10 network hops and a controller-robot distance of ~360 km (i.e., 224 miles). Two approaches were used: (**B**) the robot connected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

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 rate-based 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) and (b) Widrow-Hoff (WH) algorithm with a delta-eligibility trace tuned to the error feedback delay (WH + 50 ms) and tuned to exceed that delay by 20 ms (WH + 70 ms) (*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 neural networks representing the cerebellar cortex and DCN (*66*). These solutions, although not of direct application to the present setup because they are constrained to more simple scenarios (simulation studies, numerical experiments, LTI systems, fixed delays, simple dynamics, and dynamic model dependent), prove the efforts devoted to address the sensorimotor delay challenge from rate-based 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 the neuron level. Yet, the 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 nonlinear dynamic modeling of elastic cobots, providing real-time, adaptive, and compliant torque control of a six-DOF robot arm. The cellular-level nature of our cerebellar SNN controller enables the replication of the STDP mechanisms at the 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 toward 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 alternative communication technologies (*75*). However, the application of these approaches is tied to specific communication architectures, technologies, or protocols. Conversely, an 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 can cloud robotics 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, because 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 laboratory-controlled setups with synthetic communication delays and real-world setups that fall under higher technology readiness levels (*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 affect the behavior of the final cerebellar output; however, it still remains open what occurs to PF-PC adaptation either to a specific delay or to a range of delays at the cerebellar intermediate zone, responsible for controlling the distal extremity muscles. The PF-PC STDP in other cerebellar regions (vermis versus 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 and 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 the configuration of 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 processing unit (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 and in its robotic counterpart (i.e., signals generated at robot joint sensors all propagated through the same pathway) drove us toward the monokernel solution.

For a widely distributed robotic platform with several millisecond delay differences between the interconnected elements, i.e., segmented in different sensorimotor pathways, together with real-time 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 3.30 GHz with 12 cores, 32 GB 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 ensures both active and passive compliance (*6*).

### The cerebellar neural network

The cerebellar neural network consisted of 62,040 leaky integrate and fire (LIF) neurons and ~36.4 million synapses (36 million 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 DOFs. The neural layer distribution was the following: MFs (240 neurons), GCs (60,000 neurons), CFs (600 neurons), PCs (600 neurons), and DCNs (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 PF excitatory connections, i.e., GC 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. Last, 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 that 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, and 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 an 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 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.

## 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 annex S1 for a more in-depth description of the temporal kernel operation.

The LTD process convolved the CF and PF activity as follows

$$\mathrm{LTD}\ \Delta w_{\mathrm{PF}_j-\mathrm{PC}_i}(t) = \beta \cdot \int_{-\infty}^{t_{\mathrm{CFspike}}} k(t - t_{\mathrm{CFspike}}) \cdot \delta_{\mathrm{PFspike}}(t) \cdot \mathrm{dt}$$

$$k(x) = \begin{cases} \dfrac{-(x + d_{\mathrm{k}})}{\tau_{\mathrm{LTD}} - d_{\mathrm{k}}} \cdot e^{\frac{x+d_{\mathrm{k}}}{\tau_{\mathrm{LTD}}-d_{\mathrm{k}}}+1} & \text{if } x < -d_{\mathrm{k}} \\ 0 & \text{if } x \geq -d_{\mathrm{k}} \end{cases}$$

where $\Delta w_{\mathrm{PF}j\,-\,\mathrm{PC}i}$ is the synaptic weight change between the *j*th PF and the *i*th PC, $\beta = -0.0008$ nS is the synaptic weight decrement, $\delta_{\mathrm{PF}}$ is the Dirac delta function of an afferent spike from a PF, $k(x)$ defines the integrative kernel, $d_{\mathrm{k}} = 120$ ms allowed the adjustment of the kernel width, and $\tau_{\mathrm{LTD}}$ is the kernel eligibility trace peak. The kernel maximum value [$k(x) = 1$] is obtained when $x = -\tau_{\mathrm{LTD}}$, that is, the synaptic weight decrement is maximum for those PF spikes that were received $\tau_{\mathrm{LTD}}$ ms before the CF spike arrival. For our SNN torque controller, we established $\tau_{\mathrm{LTD}} = 150$ ms.

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

$$\mathrm{LTP}\ \Delta w_{\mathrm{PF}_j-\mathrm{PC}_i}(t) = \alpha \cdot \delta_{\mathrm{PFspike}}(t) \cdot \mathrm{dt}$$

where $\Delta w_{\mathrm{PF}j\,-\,\mathrm{PC}i}$ is the synaptic weight change between the *j*th PF and the *i*th PC, $\alpha = 0.002$ nS is the synaptic efficacy increment, and $\delta_{\mathrm{PF}}$ 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 ($Q_a$ and $\dot{Q}_a$) and the trajectory generator ($Q_d$ and $\dot{Q}_d$), had to be translated into neural activity (MF activity) that the

**Table 1. Cerebellar neural network topology.** Dash entries indicate not applicable.

| Neurons | | Synapses | | | |
|---|---|---|---|---|---|
| **Presynaptic** | **Postsynaptic** | **Number** | **Type** | **Initial weight (nS)** | **Weight range (nS)** |
| 240 MFs | $60 \times 10^3$ GCs | $240 \times 10^3$ | AMPA | 0.18 | – |
| 240 MFs | 600 DCN | $144 \times 10^3$ | AMPA | 0.1 | – |
| $60 \times 10^3$ GCs | 600 PCs | $36 \times 10^6$ | 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 | – |

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 $Q_a$, $\dot{Q}_a$, $Q_d$, and $\dot{Q}_d$, 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

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

$$R_n = c_n \pm w_n$$

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

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

where $\delta_{MFspike}(t)$ defines the Dirac delta function of an afferent spike from an 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. Because the receptor intervals within the subgroup were nonoverlapping, 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 the MF layer using the timing of spikes.

The teaching/error signal $\varepsilon(t)$, obtained by comparing the desired ($Q_d$ and $\dot{Q}_d$) and actual robot state ($Q_a$ and $\dot{Q}_a$), was also translated from the analog to the spike domain (CF 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 $\eta(t) \in [0, 1]$, 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 ($\tau_j$) 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

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

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

where $j$ = [1, 6] for each of the six DOFs; $i$ = [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; and $\alpha_j$ = (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 $\delta_e$ 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 $\delta_e$, 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

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

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 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 six-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 ($\delta = \pi/2$, $a = 1$, $b = 2$), i.e., eight-like Cartesian trajectory in the horizontal plane ($xy$ plane) (*34*, *91*). Please see 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 with actual ($Q_a$) joint position at each time step. The average difference of all joints provided the position MAE, serving as the performance accuracy metric

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

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

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

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

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

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

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

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

Because 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://doi.org/10.5281/zenodo.5171064.

## SUPPLEMENTARY MATERIALS

www.science.org/doi/10.1126/scirobotics.abf2756
Annex S1 to S6
Figs. S1 to S3
Table S1
Movie S1
References (*94*–*108*)

## REFERENCES AND NOTES


1. A. Vysocky, P. Novak, Human-robot collaboration in industry. *MM Sci. J.* **9**, 903–906 (2016).
2. H. M. Van der Loos, D. J. Reinkensmeyer, E. Guglielmelli, Rehabilitation and health care robotics, 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. *Auton. Robots* **14**, 179–197 (2003).
4. A. De Santis, B. Siciliano, A. De Luca, A. Bicchi, An atlas of physical human–robot interaction. *Mech. Mach. Theory* **43**, 253–270 (2008).
5. M. Hernando, E. Gambao, Teleprogramming: Capturing the intention of the human operator, 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 Trans. Cybern.* **51**, 2476–2489 (2021).
7. K. G. Shin, X. Cui, Computing time delay and its effects on real-time control systems. *IEEE Trans. Control Syst. Technol.* **3**, 218–224 (1995).
8. G. Niemeyer, C. Preusche, S. Stramigioli, D. Lee, Telerobotics, 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 Syst. Mag.* **24**, 58–71 (2004).
11. A. Sorribes, thesis, Universitat Politècnica de València, Valencia, Spain (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, PID stabilization of LTI plants with time-delay, 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 Eng. Pract.* **16**, 407–428 (2008).
16. O. Smith, Closer control of loops with dead time. *Chem. Engng. Progr.* **53**, 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 J. Biomed. Health Inform.* **18**, 157–166 (2014).
18. A. Manitius, A. Olbrot, Finite spectrum assignment problem for systems with delays. *IEEE Trans. Automat. Contr.* **24**, 541–552 (1979).
19. I. Vörös, B. Várszegi, D. Takács, Lane keeping control using finite spectrum assignment with modeling errors, in *Dynamic Systems and Control Conference* (American Society of Mechanical Engineers, 2019), vol. 59162, p. 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. *Int. J. Adv. Robot. Syst.* **9**, 209 (2012).
22. H. Chaoui, P. Sicard, W. Gueaieb, ANN-based adaptive control of robotic manipulators with friction and joint elasticity. *IEEE Trans. Ind. Electron.* **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. *J. Physiol.* **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. *J. Neurosci.* **25**, 3919–3931 (2005).
26. J. Stein, Cerebellar forward models to control movement. *J. Physiol.* **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 Trans. Syst. Man Cybern. B Cybern.* **41**, 1299–1312 (2011).
28. S. Ghosh-Dastidar, H. Adeli, Third generation neural networks: Spiking neural networks, 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. *Curr. Opin. Neurobiol.* **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. *Front. Neurosci.* **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. *Front. Comput. Neurosci.* **10**, 17 (2016).
32. J. G. Ziegler, N. B. Nichols, Optimum settings for automatic controllers. *J. Dyn. Syst. Trans. ASME* **115**, 220–222 (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**, 159 (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. *Front. Comput. Neurosci.* **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 Comput. Biol.* **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. *J. Neurophysiol.* **77**, 2115–2130 (1997).
37. I. Herreros, X. Arsiwalla, P. Verschure, A forward model at Purkinje cell synapses facilitates cerebellar anticipatory control, in *Advances in Neural Information Processing Systems* (NIPS, 2016), pp. 3828–3836.
38. C. Boucheny, R. Carrillo, E. Ros, J.-M. C. Olivier, Real-time spiking neural network: An adaptive cerebellar model, 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); https://repository.upenn.edu/cis_reports/300/.

40. N. A. Ali, E. Ekram, A. Eljasmy, K. Shuaib, Measured delay distribution in a wireless mesh network test-bed, 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, WiFi can be the weakest link of round trip network latency in the wild, 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 Trans. Autom. Sci. Eng.* **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. *Proc. Biol. Sci.* **285**, 20180613 (2018).
45. M. Rohde, M. O. Ernst, Time, agency, and sensory feedback delays during action. *Curr. Opin. Behav. Sci.* **8**, 193–199 (2016).
46. M. Rohde, G. Altan, M. O. Ernst, When vision lags, motor prediction follows. bioRxiv 2020.02.13.937235 [Preprint]. 14 February 2020. https://doi.org/10.1101/2020.02.13.937235.
47. S. Franklin, J. Česonis, R. Leib, D. W. Franklin, Feedback delay changes the control of an inverted pendulum, 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. *Nat. Neurosci.* **3**, 1212–1217 (2000).
49. M. G. Paulin, The role of the cerebellum in motor control and perception. *Brain Behav. Evol.* **41**, 39–50 (1993).
50. R. Shadmehr, M. A. Smith, J. W. Krakauer, Error correction, sensory prediction, and adaptation in motor control. *Annu. Rev. Neurosci.* **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, C. I. De Zeeuw, J. F. Medina, S. S.-H. Wang, Cerebellar granule cells acquire a widespread predictive feedback signal during motor learning. *Nat. Neurosci.* **20**, 727–734 (2017).
52. M. Kawato, H. Gomi, A computational model of four regions of the cerebellum based on feedback-error learning. *Biol. Cybern.* **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 Cogn. Sci.* **2**, 338–347 (1998).
55. H. Imamizu, S. Miyauchi, T. Tamada, Y. Sasaki, R. Takino, B. Pütz, 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 Trans. Syst. Man Cybern. B Cybern.* **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 Trans. Cybern.* **50**, 4744–4757 (2020).
58. C. Casellato, A. Antonietti, J. A. Garrido, A. Pedrocchi, E. D'Angelo, Distributed cerebellar plasticity implements multiple-scale memory components of Vestibulo-Ocular Reflex in real-robots, 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. *Front. Neurorobot.* **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. *Comput. Intell. Neurosci.* **2019**, 1–15 (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. *J. R. Soc. Interface* **13**, 20160547 (2016).
62. J. Porrill, P. Dean, Recurrent cerebellar loops simplify adaptive control of redundant and nonlinear motor systems. *Neural Comput.* **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. *Int. J. Neural Syst.* **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. *Biol. Cybern.* **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. *Bioinspir. Biomim.* **16**, 016004 (2021).
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. *Biol. Cybern.* **111**, 421–438 (2017).
67. P. van der Smagt, M. A. Arbib, G. Metta, Neurorobotics: From vision to action, in *Springer Handbook of Robotics* (Springer, 2016), pp. 2069–2094.
68. T. Yamazaki, W. Lennon, Revisiting a theory of cerebellar cortex. *Neurosci. Res.* **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. *Nat. Commun.* **9**, 469 (2018).
71. H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, M. Hoffmann, Industry 4.0. *Bus. Inf. Syst. Eng.* **6**, 239–242 (2014).
72. D. M. Lofaro, A. Asokan, E. M. Roderick, Feasibility of cloud enabled humanoid robots: Development of low latency geographically adjacent real-time cloud control, in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)* (IEEE, 2015), pp. 519–526.
73. R. Imai, R. Kubo, Cloud-based remote motion control over FTTH networks for home robotics, 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, Mitigating network latency in cloud-based teleoperation using motion segmentation and synthesis, in *International Symposium on Robotics Research (ISRR)* (NSF, 2019), pp. 1–16.
75. F. Voigtländer, A. Ramadan, J. Eichinger, C. Lenz, D. Pensky, A. Knoll, 5G for robotics: Ultra-low latency control of distributed robotic systems, 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 (1995).
77. B. A. Richards, T. P. Lillicrap, Dendritic solutions to the credit assignment problem. *Curr. Opin. Neurobiol.* **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. *Exp. Brain Res.* **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. *Res. Dev. Disabil.* **34**, 2118–2126 (2013).
81. C. J. Stoodley, J. D. Schmahmann, Functional topography of the human cerebellum. *Handb. Clin. Neurol.* **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 Netw.* **11**, 1317–1329 (1998).
84. C. Fitzgerald, Developing baxter, in *2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA)* (IEEE, 2013), pp. 1–6.
85. M. Ito, Cerebellar microcomplexes. *Int. Rev. Neurobiol.* **41**, 475–487 (1997).
86. M. L. Latash, Muscle coactivation: Definitions, mechanisms, and functions. *J. Neurophysiol.* **120**, 88–104 (2018).
87. W. Gerstner, W. M. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity* (Cambridge Univ. Press, 2002).
88. S. Kuroda, K. Yamamoto, H. Miyamoto, K. Doya, M. Kawato, Statistical characteristics of climbing fiber spikes necessary for efficient cerebellar learning. *Biol. Cybern.* **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. *J. Neurophysiol.* **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. *Front. Hum. Neurosci.* **8**, 234 (2014).
91. P. van der Smagt, Benchmarking cerebellar control. *Rob. Auton. Syst.* **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. *Eur. J. Neurosci.* **10**, 86–94 (1998).
93. H. Hoffmann, G. Petkos, S. Bitzer, S. Vijayakumar, Sensor-assisted adaptive motor control under continuously varying context, in *ICINCO-ICSO* (SciTePress 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).
95. R. S. Sutton, A. G. Barto, Toward a modern theory of adaptive networks: Expectation and prediction. *Psychol. Rev.* **88**, 135–170 (1981).
96. A. G. Barto, R. S. Sutton, C. W. Anderson, Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Trans. Syst. Man Cybern.* 834–846 (1983).
97. J. L. Raymond, S. G. Lisberger, Neural learning rules for the vestibulo-ocular reflex. *J. Neurosci.* **18**, 9112–9129 (1998).
98. V. Lev-Ram, S. B. Mehta, D. Kleinfeld, R. Y. Tsien, Reversing cerebellar long-term depression. *Proc. Natl. Acad. Sci. U.S.A.* **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 Robot. Autom. Mag.* **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 (2014); https://iperf.fr.
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 Comput.* **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 Netw.* **20**, 290–297 (2007).
106. T. Yamazaki, S. Tanaka, A spiking network model for passage-of-time representation in the cerebellum. *Eur. J. Neurosci.* **26**, 2279–2292 (2007).
107. T. Yamazaki, S. Tanaka, Computational models of timing mechanisms in the cerebellar granular layer. *Cerebellum* **8**, 423–432 (2009).
108. R. Apps, M. Garwicz, Anatomical and physiological foundations of cerebellar information processing. *Nat. Rev. Neurosci.* **6**, 297–311 (2005).


**Acknowledgments:** We thank J. A. Garrido, F. Barranco, and J. Navarro for fruitful discussions. **Funding:** This work was supported by European Union Human Brain Project Specific Grant Agreement 3 (H2020-RIA. 945539); European Union Neuro Cerebellar Recurrent Network for Motor Sequence Learning in Neurorobotics: NEUSEQBOT (891774); European Union and Junta de Andalucía, CEREBIO (P18-FR-2378); FEDER-Junta de Andalucía (A-TIC-276-UGR18); and the National Grants INTSENSO & SPIKEAGE (MICINN-FEDER-PID2019-109991GB-I00 & PID2020-113422GA-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 and/or the Supplementary Materials. The source code is available at https://doi.org/10.5281/zenodo.5171064.

Submitted 16 October 2020
Accepted 12 August 2021
Published 8 September 2021
10.1126/scirobotics.abf2756

**Citation:** I. Abadía, F. Naveros, E. Ros, R. R. Carrillo, N. R. Luque, A cerebellar-based solution to the nondeterministic time delay problem in robotic control. *Sci. Robot.* **6**, eabf2756 (2021).

Science Robotics

# A cerebellar-based solution to the nondeterministic time delay problem in robotic control

Ignacio Abadía, Francisco Naveros, Eduardo Ros, Richard R. Carrillo, and Niceto R. Luque

*Sci. Robot.* **6** (58), eabf2756. DOI: 10.1126/scirobotics.abf2756







*Science Robotics* (ISSN 2470-9476) is published by the American Association for the Advancement of Science. 1200 New York Avenue NW, Washington, DC 20005. The title *Science Robotics* is a registered trademark of AAAS.