

# Compliant Robot Control Using Cerebellar Spiking Neural Networks, A Biologically Inspired Approach

Ignacio Abadía Tercedor

Supervisors

Dr. Niceto R. Luque

Dr. Francisco Naveros

Advisor

Prof. Eduardo Ros



UNIVERSIDAD  
DE GRANADA

Doctoral Programme in Information and  
Communication Technologies



**COMPLIANT ROBOT CONTROL USING CEREBELLAR  
SPIKING NEURAL NETWORKS, A BIOLOGICALLY  
INSPIRED APPROACH**



**UNIVERSIDAD  
DE GRANADA**

**Ignacio Abadía Tercedor**

Thesis supervisors: Dr. Niceto Rafael Luque Sola, Dr. Francisco Naveros Arrabal

Department of Computer Architecture and Technology  
University of Granada

A thesis submitted for the degree of

Philosophiae Doctor, PhD, from the University of Granada  
Doctoral Programme in Information and Communication Technologies

2022

Editor: Universidad de Granada. Tesis Doctorales  
Autor: Ignacio Abadía Tercedor  
ISBN: 978-84-1117-567-8  
URI: <https://hdl.handle.net/10481/77687>



## Acknowledgments

This dissertation has been made possible thanks to the efforts and support of many people.

I am deeply grateful to Prof. Eduardo Ros for opening the door to academic research for me and providing guidance over these years, he welcomed me into his lab and made me feel comfortable from the very first day. I am very fortunate to have had the support of two brilliant supervisors: Dr. Niceto Luque and Dr. Francisco Naveros. Their hard work, rigor, and generosity have made possible this thesis. They have provided advice and promoted personal initiative. For their friendship I am extremely grateful. I would like to extend my gratitude to Dr. Jesús Garrido, Dr. Richard Carrillo, and Dr. Francisco Barranco for willingly sharing their valuable knowledge and experienced perspective. I am also thankful to everyone in the Applied Computational Neuroscience & Neurorobotics research group, for sharing lab hours, meetings, and concerns. To my colleagues and friends from D1-2 & D1-3, thanks for making this journey easier.

Thanks to Prof. Auke Ijspeert for hosting me at the Biorobotics Laboratory (EPFL). It was my pleasure to spend three months there and get a sense of their brilliant research. Thanks to everyone at BioRob who made those three months a refreshing and truly enjoyable experience. Special thanks to Alice Bruel for her direct implication in this thesis.

To my family and friends, thanks for the unconditional, endless love. To my brother and sister. To my mom and dad. I owe you everything I am.



# TABLE OF CONTENTS

Abstract.....	11
List of Figures.....	13
List of Tables.....	15
Glossary.....	17
1. Introduction.....	19
1.1 New robotics paradigm: collaborative robots.....	20
1.2 Neurorobotics.....	21
1.3 Cerebellar motor control theory.....	23
1.4 Spiking neural networks.....	25
1.5 Motivation.....	26
1.6 Objectives.....	27
1.7 Project framework.....	29
2. Cerebellar adaptive motor control.....	31
2.1 Introduction.....	31
2.2 Methods.....	33
2.2.1 Compliant robot – Baxter.....	33
2.2.2 Cerebellar control loop.....	34
2.2.3 Cerebellar controller – The neural network.....	36
2.2.4 Spiking neuron models.....	39
2.2.5 Synaptic plasticity.....	40
2.2.6 Analog to spike translation.....	41
2.2.7 Spike to analog translation.....	42
2.2.8 ROS control loop implementation.....	43
2.2.9 Benchmarking the cerebellar controller – Behavioral tasks.....	44
2.3 Results.....	46
2.3.1 Circular trajectory.....	46
2.3.2 Eight-like trajectory.....	49
2.3.3 Target reaching.....	51
2.3.1 Unstructured interactions.....	52

2.4	Discussion.....	54
2.5	Conclusion.....	54
3.	Motor control under nondeterministic time delays.....	57
3.1	Introduction .....	57
3.2	Methods .....	59
3.2.1	The cerebellar control loop with induced transmission delays.....	59
3.2.2	The cerebellar neural network .....	61
3.2.3	The STDP mechanism - The “eligibility trace” and how it enters the learning rule equation .....	62
3.2.4	Translation from neural activity to torque commands.....	64
3.2.5	Desired trajectories definition .....	65
3.2.6	The ANN cerebellar model.....	66
3.2.7	Performance accuracy and learning convergence measurement .....	68
3.2.8	Modules implementation .....	68
3.3	Results .....	69
3.3.1	Cerebellar torque control provides learning convergence in the presence of time delays.....	69
3.3.2	STDP at PF-PC copes with the delay. Overcoming the 150 ms delay biological limitation.....	73
3.3.3	Benchmarking the nondeterministic time delays.....	75
3.3.4	Nondeterministic Wi-Fi and cloud-robotics time delays; cerebellar control use cases .....	76
3.4	Discussion.....	78
3.4.1	PF-PC STDP modeling considerations .....	81
4.	The central nervous system: a distributed neural controller – Beyond cerebellar control.....	83
4.1	Introduction .....	83
4.2	Spinal cord and cerebellum integration.....	84
4.3	Methods .....	85
4.3.1	Front-end body: musculoskeletal upper limb .....	85
4.3.2	Spinal cord circuits .....	86
4.3.3	Connecting the cerebellum and the spinal cord.....	87
4.3.4	Motor tasks .....	88
4.4	Results .....	89

4.5	Application to robotics – Adding muscles to cerebellar torque control .....	90
4.6	Methods .....	91
4.6.1	Muscle model .....	91
4.6.2	Baxter cerebellar control loop with muscles .....	92
4.7	Results .....	92
4.8	Discussion and future work .....	93
5.	Discussion.....	95
5.1.1	Summary of accomplished objectives .....	97
6.	Conclusion.....	99
	Research publications .....	101
	Resumen .....	103
	Bibliography .....	113



# ABSTRACT

In the last decades, a new robotics paradigm has been introduced due to physical human-robot interaction (HRI) and the use of collaborative robots (cobots) equipped with low-power actuators and elastic components. This scenario requires the use of cobot controllers able to operate in unstructured environments and that do not depend on the accurate mathematical modeling of the nonlinear dynamics introduced by elastic elements. Robot behavior in this context is required to emulate the adaptability and flexibility of human behavior as much as possible.

The cerebellum, pivotal for human motor control, has long been proposed as an adaptive controller, and its regular neural structure has allowed the development of computational models which replicate, to some extent, its structural and functional properties. Here, we propose a cerebellar-based adaptive controller able to provide torque control of a cobot with nonlinear dynamics. Using spiking neural networks we replicate the cerebellum neural topology and synaptic plasticity mechanisms. We then embed the biologically plausible cerebellar network at the core of a cobot control loop.

The spike-processing computational cost of biologically plausible cerebellar models has prevented their real-world applicability, thus relegating them to mere theoretical or simulated models. Within this dissertation, we prove the applicability of our biologically plausible cerebellar controller in real-world control problems. We present a cerebellar spiking neural network which is large enough to provide the required resolution for torque control of six degrees of freedom in real-time, and hence can operate real cobots. The cerebellar controller provides fine accuracy in the execution of different motor tasks thanks to the deployed cerebellar learning mechanisms. Besides, the controller is also able to adapt the cobot behavior to unstructured changes directly affecting the cobot dynamics. Furthermore, the aforementioned cerebellar control learning mechanisms can also cope with sensorimotor delays affecting the robot-controller communication, a well-known source of control loop instability. Sensorimotor latency is unavoidable in the central nervous system (CNS), however, it does not jeopardize the stability of motor control thanks to, among others, cerebellar predictive behavior. We prove the cerebellar controller robust against sensorimotor delays of different nature, thus applying to robotics another intrinsic feature of the cerebellum.

In addition to cerebellar control, we expand the biologically inspired approach with other key elements of the CNS and musculoskeletal system. We present some first results of adding spinal cord circuits to the cerebellar controller. The spinal cord, using

direct muscle feedback to allow fast-stretch reflexes and muscle activity regulation, is found to improve cerebellar learning and robustness against perturbations. As next step we will integrate the spinal cord circuits and the cerebellar controller operating the cobot, for which muscle dynamics will need to be added to the control loop. Here we present a preliminary approach for the integration of muscle dynamics within the cobot control loop, which is shown capable of modifying the motion stiffness of the cobot by changing the cocontraction degree of antagonistic muscle pairs. Different stiffness profiles would allow the robot behavior to cover different degrees of admittance and impedance control, of interest to physical HRI as those control modes directly impact how the robot reacts to external interactions (admittance control performs better in soft environments, while impedance control favors stiff environments).

For collaborative robotics to succeed, robot performance must emulate the adaptability and flexibility of human behavior. Hence, the biological substrate behind human conduct could be used as inspiration to bring robot behavior closer to our inherent motor capabilities. Human behavior is sustained by both *hardware* and *software*: the biomechanics of the musculoskeletal system together with the control provided by the CNS allow us to interact with others and the environment. On the hardware side, robot design is increasingly mimicking the dynamics of living beings. On the software side, the study and understanding of the different CNS areas and their computational replication can expand the family of controllers able to provide adaptive, compliant robot control. Here, we benefit from decades of neuroscience studies about the cerebellum structure and functioning, and apply those findings to current robotic challenges.

# LIST OF FIGURES

Figure 1.1. Neurorobotics .....	22
Figure 1.2. Details from the cerebellar structure .....	23
Figure 2.1. Baxter robot .....	33
Figure 2.2. Dynamics of the real robot vs. simulated robot .....	34
Figure 2.3. Schematic of the cerebellar closed control loop .....	35
Figure 2.4. Cerebellar network scheme .....	36
Figure 2.5. Detailed cerebellar closed control loop scheme .....	43
Figure 2.6. Behavioral evolution through circle trajectory trials (2 s) .....	48
Figure 2.7. Behavioral evolution through eight-like trajectory trials (2 s) .....	50
Figure 2.8. Baxter's end-effector position and linear velocity when performing the eight-like trajectory .....	51
Figure 2.9. Behavioral evolution through target reaching trials (2 s) .....	52
Figure 2.10. Performance in an unstructured environment .....	53
Figure 3.1. Cerebellar control loop .....	60
Figure 3.2. Spike coding at the input MF layer .....	66
Figure 3.3. Trajectory learning convergence curves .....	69

Figure 3.4. PD and cerebellar ANN vs cerebellar SNN control response to steady time delays .....	71
Figure 3.5. Cartesian space representation of Baxter's end-effector under PD, ANN and SNN torque control .....	72
Figure 3.6. Modifying the cerebellar predictive time margin by varying the STDP kernel .....	74
Figure 3.7. Symmetric and asymmetric nondeterministic delays scenario .....	75
Figure 3.8. Cerebellar response to nondeterministic Wi-Fi delays .....	77
Figure 3.9. Cerebellar response to remote control .....	78
Figure 3.10. Cerebellar SNN multikernel vs monokernel solution coping with time delays .....	82
Figure 4.1. Musculoskeletal upper limb model .....	85
Figure 4.2. SC circuits and muscle pairs .....	86
Figure 4.3. Cerebellum-SC control loop .....	88
Figure 4.4. Motor tasks .....	89
Figure 4.5. Performance with and without SC circuits (SC-Cerebellum vs Cerebellum controller) .....	90
Figure 4.6. Performance of the different cocontraction profiles .....	93

# LIST OF TABLES

Table 2.1. Cerebellar neural network topology .....	38
Table 2.2. Neuron models parameters .....	40
Table 2.3. Circular and eight-like trajectories: position MAE (mean and std) at different learning stages .....	47
Table 2.4. Target reaching: position MAE (mean and std) at different learning stages	51
Table 3.1. Cerebellar neural network topology .....	62
Table 4.1. Muscle model parameters.....	92



# GLOSSARY

HRI: human-robot interaction

CNS: central nervous system

AI: artificial intelligence

ANN: artificial neural network

SNN: spiking neural network

RT: real time

HBP: Human Brain Project

STDP: spike-timing-dependent plasticity

LIF: leaky integrate and fire

ROS: Robot Operating System

MF: mossy fiber

GC: granule cell

PC: Purkinje cell

CF: climbing fiber

DCN: deep cerebellar nuclei

PF: parallel fiber

IO: inferior olive

LTP: long-term potentiation

LTD: long-term depression

NTP: Network Time Protocol

MAE: mean absolute error

PID: proportional-integral-derivative

LTI: linear time invariant

DTC: dead-time compensator

R2C: robot-to-controller

C2R: controller-to-robot

PD: proportional-derivative

PDF: probability density function

CDF: cumulative distribution function

RTT: round-trip time

VOS: vestibulo-ocular system

SC: spinal cord

EMG: electromyography



# 1. INTRODUCTION

Humanity has used technology as an answer to diverse questions and needs. From the wheel to airplanes or computers, we can give form to ideas that did not exist before and develop solutions for a wide variety of challenges. Robots first appeared in the last century as tools for physical labor, and their use has extended since then from automated factories to every-day tasks and new application domains. Technological development in the robotics field is driven by an increasing demand for robots provided with ever more human-like behavior.

Human behavior is mediated by the brain and rest of the nervous system; the most magnificent tool ever built. This wonder of nature and evolution provides us with the cognitive, emotional, and motor skills required to successfully interact with others and the environment. Robotics faces the challenge of developing machines with high-level cognitive and motor skills. On the cognitive side, the increase in computation capacity has already allowed developing machines that outperform human beings in certain tasks; the already classic example of computers beating chess grandmasters. But on the motor side, there is no robot yet able to move and adapt to the environment with the same levels of complexity and accuracy as humans and animals can. There are different engineering approaches trying to solve what evolution already did; hence, another logical approach would be to take inspiration from evolution itself and biological systems. Any researcher would dream about having the millions of years and resources used by evolution to find the most suitable solutions to its challenges. Here, we propose a robot controller developed by taking inspiration from how humans and animals can move and adapt to different contexts. Specifically, a spiking neural network that replicates, in a biologically plausible way, the main neural layers and synaptic plasticity forms of the cerebellum, a nervous region that is pivotal for motor learning and coordination.

The embodiment of biologically plausible neural networks can inspire the development of more advanced robots. Besides, these robots equipped with replicas of our nervous system also appear as a tool for neuroscience to study the behavior of those neural circuits under different contexts and tasks. Thus, a promising symbiosis can be established between neuroscience and robotics: next generations of robots can be inspired by neuroscience, which can in turn use those robots to gain a better understanding about the functioning of the brain and the rest of the nervous system.

## 1. Introduction

### 1.1 New robotics paradigm: collaborative robots

The word *robot* was originally coined in 1920 at the theatre play R.U.R. (Rossum's Universal Robots), by Czech playwright Karel Čapek [1], after a suggestion from his brother Josef Čapek [2]. The word described the artificial workers conceived to be employed in factories.

Since their first, fictional appearance, robots have been long used in industry to lighten human labor by performing demanding physical jobs, such as manipulating heavy payloads or toxic substances, or executing monotonous tasks. For the sake of performance accuracy and efficiency, industrial robots are rigid-bodied equipped with high-power actuators and high-ratio gearboxes, allowing the implementation of traditional position control solutions based on kinematic model availability. These conditions, both hardware and software, prevent any physical human-robot interaction (HRI), forcing the use of safety fences to define the robot workspace. Thus, in these traditional, well-structured, industrial scenarios, any physical interaction between robot and human is avoided as safety cannot be guaranteed.

Nevertheless, new robotic application domains (e.g., search and rescue missions [3], medical assistance and rehabilitation therapies [4], social interaction [5], education [6]) demand safe physical HRI [7]. Even in the industrial domain there is a drive towards safe physical HRI to harness the strengths of both human (e.g., adaptability, creativity) and robot (e.g., precise manipulation, lifting heavier payloads) [8]. Hence, the family of robots solely conformed by traditional, industrial robots, needs to be enlarged with a new member: compliant robots able to provide for safe physical HRI, i.e., collaborative robots (cobots). For the development of cobots, both hardware and software solutions are to be considered.

On the hardware side, emulating the kinematics and dynamics of animals is pushing toward the development of more versatile robots able to provide robust operation in a wider range of environments [9], [10], [11]. Apart from robotic applications, these robots are also platforms for the study and further understanding of the biomechanics of living beings [12], [13], and even extinct ones [14]. Regarding the requirement for compliance and safety in cobots, low-power actuators and elastic components are used to ease the consequences of possible impacts. This approach, taking inspiration from biological soft tissues (e.g., muscles, tendons and ligaments), offers passive intrinsic compliance in contrast to classic rigid-bodied robots. Besides compliance, deformable bodies can potentially provide more adaptation, sensitivity and agility [15], [16]. But the use of these hardware components comes at a cost: they hinder the applicability of traditional torque control methods, as the nonlinearity introduced makes the dynamic model mathematically intractable [17]. Hence, on the software side, new control strategies are demanded [15], [18]; adaptive torque controllers that do not depend on dynamic model availability and that are able to cope with physical nonlinearities. In addition to the impositions made by the hardware approach, these new controllers are also expected to operate in unstructured scenarios, derived from the unpredictable

nature of human interaction, and to cope with environmental changes across time and tasks, as a result of operating in the real world [19].

Another element differentiating classic robotics scenarios and the new application domains is the physical link and communication technologies connecting robot and controller. The traditional point-to-point wired connections now coexist with wireless communications [20], cloud-robotics networks [21], and teleoperation architectures [22], which benefits come at the cost of inducing stochastic transmission delays in the control loop. The presence of nondeterministic transmission delays contributes to the unstructured nature of these new robot scenarios, and needs to be taken into account when designing torque control solutions.

From the traditional, well-structured, industrial applications, robotics has evolved towards a new paradigm. The distinctive characteristics of this new collaborative scenario can be summarized as: low-power actuators, elastic components, and adaptive torque control able to cope with both nonlinear dynamics and unstructured scenarios. Robotics research is devoting efforts toward achieving these features, which are all innate in human and animal motor control. Therefore, the new torque control strategies to be developed could directly benefit and take inspiration from a better understanding of biological motor control.

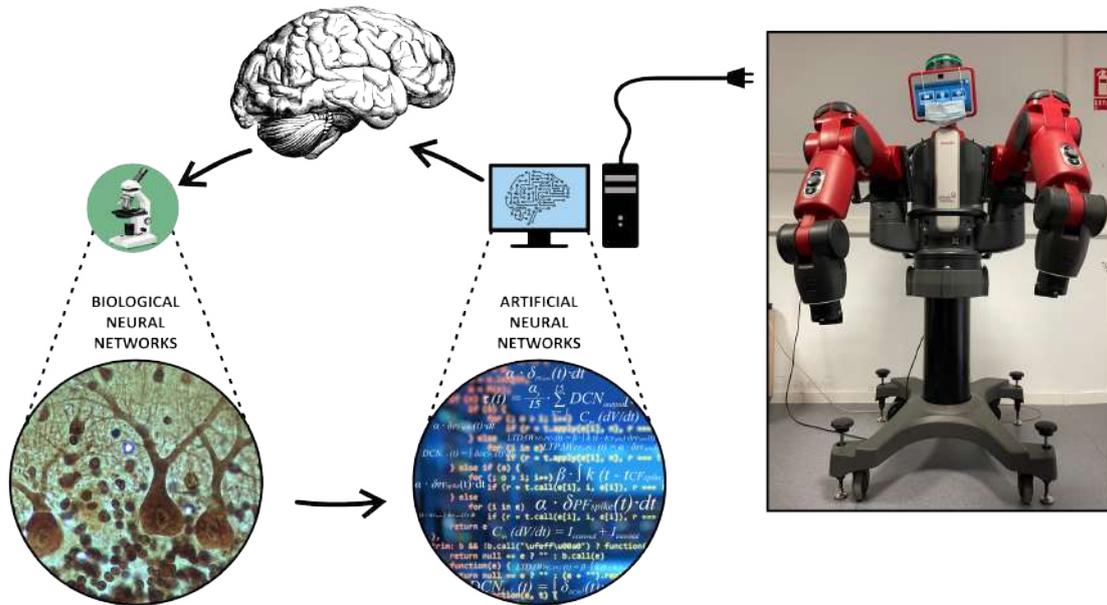
## 1.2 Neurorobotics

Physical HRI and the integration of robots in society require robots to express complex behaviors similar to those exhibited by living creatures. To develop the engineering solutions that shall allow robots to interact with others and the environment, we should harness the experience gained by nature in its search for the most suitable solutions to the very same challenge. Millions of years of evolution have developed nervous systems that allow us, and animals, to efficiently plan our actions as a response to sensory inputs, to predict their consequences, and to adapt them to an unstructured world. Perception, information processing, and the generation and execution of motor commands, constitute a well-orchestrated process involving different areas of the nervous system. The study and understanding of the different parts of the nervous system can inspire the development of new solutions for compliant robot behavior, which involves the whole perception-cognition-action loop.

The human visual system, driven by events, has inspired computer vision technologies using event-driven sensors [23]; the biological afferent nerves, i.e., interface between sensorial information and the central nervous system (CNS), have been mirrored by *in silico* spiking afferent nerves with energetic efficiency and computation capacity suitable for processing sensory signals from robot sensors [24]; head direction cells, found in the limbic system, have been computationally modeled and applied for robot navigation tasks [25]; spinal cord circuits and central pattern generators have been replicated to equip robots with robust locomotion capabilities [26], [27], [28]. These are just some examples of how neuroscience knowledge translated into computational

## 1. Introduction

models is already providing solutions for the development of more advanced robots. Besides, the embodiment of computational neuroscience models can be a powerful mechanism to test the fidelity of the simulated neural networks [29], providing a framework to study neurological behavior, pathologies, and treatments. This symbiosis established between neuroscience and robotics is what we call *neurorobotics* (Fig. 1.1).



**Figure 1.1. Neurorobotics.** Biological neural networks can be replicated using artificial neural networks, which can be used as robot controllers taking inspiration from biological motor control. The embodiment of these artificial neural networks provides a framework for testing the fidelity of the simulated neural networks, which can also be used to gain a better understanding of the biological substrate.

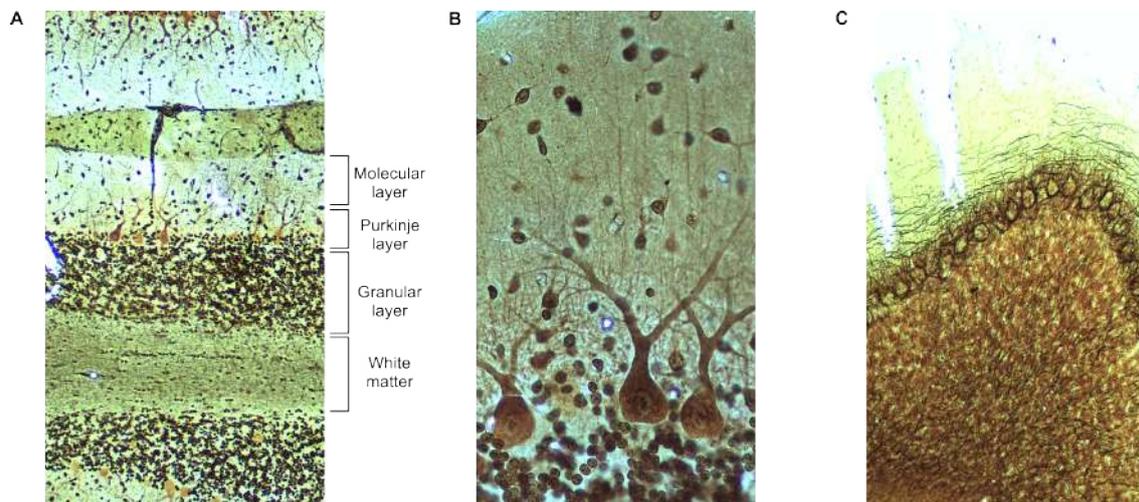
Regarding motor control of the musculoskeletal system, several areas of the CNS are involved: the premotor cortex plays a role in movement selection and organization [30], the primary motor cortex is directly involved in control of voluntary movements, segregated in neural sub-regions for different body parts [31], the basal ganglia and brainstem are associated with control of muscle tone and locomotion [32], the cerebellum is known for its crucial role in supervised sensorimotor learning [33], the spinal cord integrates descending signals from higher brain areas and controls motor neurons activation, as well as spinal circuits being able to generate rhythmic locomotor movements and provide fast-reflex responses [34]. These examples provide just a glimpse of the complex interaction of different CNS regions, with overlapping functions yet to be further clarified, shaping the widely distributed neurobiological control process.

Among all the CNS regions involved in motor control, the cerebellum appears as especially suitable for robot control thanks to its structure and function. The regular neural structure of the cerebellum favors its computational replication, while its recognized crucial role in motor learning makes it appealing for adaptive robot control solutions [35].

### 1.3 Cerebellar motor control theory

The human brain is estimated to have ~86 billion neurons, ~69 billion contained by the cerebellum; representing only 10% of the human brain mass, the cerebellum holds 80% of the neurons [36]. Cerebellar anatomy, characterized by a high neural density, has historically aroused a large body of research, from Aristotle already distinguishing it from the rest of the brain anatomy [37], to Ramón y Cajal masterful description of the cerebellar cortex layered structure [38]. The cerebellar regular structure has allowed detailed experimental research for centuries [39], leading to certain consensus on the anatomy of the cerebellum [40].

The cerebellar topology is structured in two well differentiated areas: the cerebellar cortex and the deep cerebellar nuclei. The cerebellar cortex is divided in three layers: granular layer, Purkinje layer, and molecular layer (Fig. 1.2). Granule cells densely populate the granular layer, where also Golgi and Lugaro cells are found. Purkinje cells give name and populate the Purkinje layer, and are the sole output of the cerebellar cortex projecting inhibitory connections to neurons in the deep cerebellar nuclei. At the molecular layer, basket and stellate cells are found, together with the axons of granule cells, the parallel fibers. Two major sensorimotor afferents reach the cerebellum, mossy fibers and climbing fibers, both forming excitatory synapses. Mossy fibers form connections with granule and Golgi cells in the cerebellar cortex, and also with neurons in the deep cerebellar nuclei. Mossy fibers convey sensorimotor signals from different sites of the CNS (cerebral cortex, vestibular nuclei, spinal cord, etc.). Climbing fibers arise from the inferior olive and form excitatory connections with Purkinje cells, interneurons at the molecular layer, and the deep cerebellar nuclei.



**Figure 1.2. Details from the cerebellar structure.** (A) Cerebellar cortex layered structure. (B) Purkinje cells. (C) Axons innervating Purkinje cells. Cat cerebellum histological slides from Prof. F. Abadía-Fenoll's archive, provided by Prof. F. Abadía-Molina.

On the functional side, the connection between cerebellum and motor control has been long established. In the first decades of the nineteenth century, Luigi Rolando linked cerebellar lesions to voluntary movement and posture impairment, Marie Jean-Pierre

## 1. Introduction

Flourens associated it to muscle coordination, and François Magendie proposed it as the equilibrium center [39]. Further studies contributed to settle the cerebellar involvement in the integration, regulation and coordination of motor processes; but it was in the late 1960s and early 1970s that a major step was achieved in cerebellar motor control theory: the Marr-Albus-Ito theory on cerebellar motor learning, based on David Marr's theory (1969) [41], extended and further supported by the work of Masao Ito (1970) [42], and followed by the independent work of James Albus (1971) [43]. Although some differences can be found among the three proposals, they shared the basis for motor learning in the cerebellum: i) parallel fibers to Purkinje cells synapses exhibit plasticity; ii) synaptic plasticity is regulated by the climbing fibers activity reaching Purkinje cells; iii) granule cells conduct an expansion recoding of mossy fibers sensory inputs. Besides a milestone in the study of cerebellar function, the Marr-Albus-Ito theory was also an example of the powerful contribution that computational neuroscience can make to brain studies [44]. The main sticking point between the models was later addressed by Masao Ito's experimental discovery of long-term depression (LTD) at parallel fibers – Purkinje cells induced by climbing fibers [45]; i.e., the climbing fibers activity does not facilitate parallel fibers – Purkinje cells synapses (Marr's suggestion), but rather depresses them (Albus' postulate). Thus, the Marr-Albus-Ito model was established and funded the computational principles of supervised learning in the cerebellum [33].

In supervised learning, the response of a system to a given stimuli is compared to the desired outcome, resulting in a teaching signal used as feedback to readjust some of the system's internal parameters. This iterative process tends to minimize the distance between the system's performance and the desired behavior. The cerebellar machinery allows supervised learning by adapting internal parameters (i.e., synaptic plasticity) as a function of the sensorimotor inputs (i.e., mossy fibers input activity, recoded at granule cells) and the evaluation of the corresponding motor response outcome through a teaching signal (i.e., climbing fibers convey the instructive signal generated at the inferior olive) [33]. Thus, the cerebellum enables the modification of our actions so their consequences match our expectations [46].

Nonetheless, accurate control of our body cannot solely rely on feedback availability, which is affected by biological sensorimotor delays. The physiological transport of sensory inputs and motor commands inevitably involves sensing delays, nerve conduction delays, synaptic delays, neuromuscular junction delays, electromechanical delays, and force generation delays [47]. The development of fine motor control requires mastering the sensorimotor delays inherent to the CNS perception-cognition-action loop. Delay compensation is achieved by state and sensory prediction, i.e., an estimation of the motor commands outcome before feedback is available [48], a prediction mechanism in which the cerebellum, equipped with motor learning ability, plays a pivotal role [49], [50]. This predictive behavior is synthesized in the idea of the cerebellum storing internal models of the body and external tools [51], i.e., a representation of the input-output properties defining the motor apparatus. There are two types of internal models, forward and inverse models. Forward models predict the next state based on the current state and the motor command. Inverse models provide

the motor command required to achieve a desired change in state. Both types of internal models have been hypothesized to be contained and combined in the cerebellum [51], [52].

The Marr-Albus-Ito model constitutes a well-accepted common ground for cerebellar research. Nonetheless, since it was first established, the model has evolved facing new arguments and findings [53]. Regarding motor control theory, it is noteworthy that synaptic plasticity, initially described at parallel fibers – Purkinje cells, has been experimentally found to be distributed across several connections in the cerebellar cortex and deep cerebellar nuclei; cerebellar learning appears as the integration of plasticity at several sites and in different forms [54]. These findings have allowed to deepen the classical view of the cerebellum as a supervised learning machine [33], and also to propose new theories suggesting the cerebellar cortex as a reinforcement learning machine [46], [55], allowing the cerebellum not only to learn how to perform an action but also to select the most rewarding action in a given situation.

Cerebellar research will keep evolving with further experimental studies together with more complex cerebellar computational models. These efforts will help to draw an increasingly refined cerebellar theory for motor control, together with uncovering the cerebellar role in cognitive functions [56] and how it is integrated in the whole brain network.

## 1.4 Spiking neural networks

From an engineering perspective, a definition of intelligence could be the efficient generation of goal-oriented outputs as a response to information inputs. For it to be applicable to a constantly changing world, intelligence must allow the adaptation of its responses to the surrounding changes; thus, learning ability is a basic tenet for intelligent behavior. This feature, naturally present in living creatures, can now be given to machines thanks to the technological development of the ever more ubiquitous field of Artificial intelligence (AI), in which artificial neural networks (ANNs) learn how to solve different tasks.

ANNs have been applied to a wide range of challenges: from speech recognition [57], to cancer detection [58], or autonomous driving [59]. In these networks, an artificial neuron, called unit, receives inputs from other units or external sources, processes the weighted sum of the inputs and transmits the output through the neuron connections, adjusting the connections as learning proceeds [60] in a continuous manner. ANNs are designed to address challenges by means of well-structured data and typically using standard analog representations for neural activity. On the one hand, as ANNs are not constrained to mimic biologically observable learning processes, the possible technological developments are endless. But on the other hand, as these networks are vaguely representative of their biological counterparts, drawing any analogies between them is hampered. Hence, another approach is needed when one intends to: i) harness the experience of millions of years gained by evolution in its search for the most

## 1. Introduction

suitable solution to its challenges, ii) develop computational models that contribute to the understanding of biological networks. Spiking neural networks (SNNs) naturally cope with these requirements, as they constitute the most biologically plausible approach of neural networks [61].

SNNs inherit the properties of biological neural networks by replicating their dynamics at a neuron level. These artificial networks transfer and process information as occurs in biological neurons, via the precise timing of spikes, i.e., membrane action potentials at discrete points in time. The spikes of a given neuron look all alike, hence, it is the precise timing of the spike, or sequence of spikes, rather than the form of the event which carries the information [61], [62]. The temporal dimension is hence intrinsically added for information encoding using SNNs, turning them a suitable solution for capturing the temporal evolution of analog signals, a pivotal feature in motor control and movement coordination [63]. Besides, neuromorphic computing based on SNNs has also been recognized for its energetic efficiency and computational power [24], [64], [65]. Therefore, the value of SNNs not only relies on their biological plausibility but also on their temporal nature, energy consumption and processing capacity.

SNNs enable the deployment of cellular-level cerebellar models, which offer an insight into the cerebellar function at a neuron level. However, this more biologically plausible approach comes at the cost of a greater computational cost when compared to functional models [35]; the application of cellular-level models to robot control has been usually limited. To perform accurate torque control of a nonlinear cobot, the SNN sizing must be large enough to capture the complex dynamics of the robot and provide high-resolution output signals, but at the same time it is also required to operate in real time (RT). In this thesis, we have fulfilled these requirements using the SNN simulator EDLUT [66], [67], [68], specially designed for RT operation.

### 1.5 Motivation

Robotics has evolved from isolated robots in structured, industrial scenarios, to robots and humans interacting in unstructured environments. Collaborative robots are the answer of the robotics field to allow HRI to be safe for both human and robot. Cobots are distinguished from traditional robots due to the use of low-power actuators and elastic components; on the hardware side, cobots are designed to provide passive compliance. But these hardware solutions hamper the implementation on the software side; traditional controllers relying on accurate analytical models are no longer applicable due to the nonlinearities and mathematical intractability introduced. Therefore, there is a demand for adaptive controllers that do not depend on prior analytical knowledge of the cobot dynamics.

HRI demands the robot to perform with accuracy but also to have control of how the movement is executed, i.e., the torque/force profiles causing the movement [69]. Therefore, in contrast to classic position controllers, torque control is presented as a more suitable solution for HRI. This control strategy is closer to human-like behavior,

as we do not command our joints to positions in space, but rather control the activation of our muscles to perform certain movements.

Hence, the roadmap is clear. There is a need for adaptive torque controllers able to operate cobots without dependence on analytical solutions. The development of the engineering solutions to this challenge can inspire from how biological motor control addresses the very same problem.

Different areas of the CNS collaborate to provide us with accurate and adaptive control of our body. From these nervous regions, the cerebellum stands out among them thanks to its regular neural structure and its recognized contribution to the integration and coordination of motor processes, but most importantly, motor learning. The computational replication of the cerebellum is favored by its structure, while its key role in motor learning proposes it as a promising candidate for robot control.

The computational replication of the cerebellar structure and functioning in the most biologically plausible way comes from the use of spiking neural networks. This kind of ANN replicates, at a neuron level, the dynamics of biological neural networks. Hence, SNNs can directly benefit from the knowledge provided by neuroscience on how the cerebellum learns motor commands and adapts to changes in the environment. Besides, the temporal dimension intrinsically present in SNNs makes them appealing for capturing the evolution through time of a physical system, the core of motor control. However, to successfully apply biologically plausible SNNs to robot control, the SNN is required to operate in RT while replicating the complex neural behavior encountered in biology. This condition has traditionally constrained SNNs to simple, reduced networks or simulation frameworks, or turned SNNs fully dependable on specific neuromorphic hardware. This scenario has hampered the deployment of biologically realistic SNNs applied to real-world cobot control.

In this dissertation, we present a biologically plausible SNN that replicates the main cerebellar structure and motor learning mechanisms, running in a conventional computer with RT operation guaranteed thanks to the EDLUT simulator [68], and performing torque control of 6 degrees-of-freedom (DOF) of a nonlinear cobot (Baxter robot [70]). The benefits of the presented biologically inspired approach are discussed in the following chapters.

## 1.6 Objectives

The main aim of this thesis is to develop a biologically plausible, cerebellar-like, torque controller for a cobot equipped with low-power actuators and elastic components (i.e., nonlinear dynamics). The controller is required to provide adaptive torque control, nondependent of prior analytical knowledge of the robot; and the performance is required to be accurate and compliant so it facilitates safe HRI.

To study the feasibility of the aforementioned, a cerebellar SNN was embedded in the control loop of a Baxter robot [70], performing torque control of 6 DOF in RT. Once

## 1. Introduction

the control loop was established, the cerebellar SNN controller was tested doing different tasks and in different contexts, assessing the implications and benefits of this bio-inspired approach.

Besides, this dissertation also explores expanding the biologically inspired, cerebellum-based approach, by adding to the cerebellar control loop other key elements of biological motor control: spinal cord circuits and muscle dynamics. Integration of the cerebellar SNN model and spinal cord circuits will provide a framework to study the complementary roles of these two nervous regions, while exploring the possible benefits of adding spinal cord circuits to cobot control. Muscle dynamics are proposed as a way to provide a control layer over the degree of motion stiffness, allowing different cobot behaviors depending on the HRI requirements. A low stiffness profile would benefit soft interactions, while high stiffness profiles would perform better in stiff environments.

To achieve the main goal, this thesis can be divided in the following specific objectives:

- Development of the cerebellar SNN adapted for 6 DOF using EDLUT simulator.
- Integration of the cerebellar SNN in the control loop of a Baxter robot, in RT.
- Validation of the motor learning capability of the cerebellar SNN torque controller. Learning torque commands to perform different trajectories involving 6 nonlinear DOF of one arm of the Baxter robot.
- Validation of the performance accuracy of the cerebellar SNN torque controller. The achieved motor learning must provide fine accuracy in the execution of the motor tasks.
- Validation of the adaptability of the cerebellar SNN torque controller to unstructured scenarios. The SNN must be able to adapt to dynamic changes induced by context variations
- Validation of the compliant behavior of the cerebellar SNN torque controller. The SNN must allow safe HRI.
- Validation of the robustness of the cerebellar SNN torque controller against nondeterministic time delays. The SNN must be able to control the cobot in scenarios affected by nondeterministic transmission delays between the robot and controller (e.g., wireless communications, remote control, and cloud-robotics).
- Integration of the cerebellar SNN and spinal cord circuits.
- Validation of the performance of the cerebellum – spinal cord integration controlling a musculoskeletal upper limb model.

- Validation of the contribution of spinal cord circuits in handling motor perturbations.
- Integration of muscle dynamics in Baxter control loop.
- Validation of the performance of different muscle cocontraction profiles and their response against perturbations.

## 1.7 Project framework

The work included in this thesis has been developed in the context of the Human Brain Project (HBP). The HBP is a European Commission Future and Emerging Technology Flagship project, a ten-year initiative started in 2013. This research project puts together scientists and engineers from more than 140 universities, hospitals, and research centers, to address the fascinating challenge of unraveling the human brain. The HBP constitutes a multidisciplinary framework with contributions from a wide range of research groups, including experimental wet lab studies or computational models and theories on the mechanisms involved in cognition, learning, or information processing at neuron level and large-scale networks. The project intends to translate the acquired knowledge to make an impact in health and technological innovation.

To address such a demanding challenge, the HBP is organized in different cross-disciplinary work packages (WP). The work covered in this thesis is integrated in WP3: “Adaptive networks for cognitive architectures: from advanced learning to neurorobotics and neuromorphic applications”. WP3 focuses on understanding how biological neural networks enable human visuo-motor and cognitive functions, with a special emphasis on embodiment on real-world systems. Within WP3, our work forms part of Task 3.4 *Closed-loop dynamic task performance: cognitive neurorobotics*.

The work presented in this thesis has been developed during two phases of the HBP: Specific Grant Agreement 2 and 3 (SGA2 and SGA3). SGA2 ran from April 2018 to March 2020, while SGA3 covers from April 2020 to September 2023, constituting the last phase of the HBP. Our research group contributed to SGA2 with the application of a cerebellar SNN model to the closed-loop control of a cobot, while at SGA3 the focus is at the integration of the cerebellar model with a spinal cord model to perform motor control of an upper limb musculoskeletal model. The cerebellum and spinal cord integration intends to study how these two nervous areas collaborate in motor control tasks, and it also carries possible applications to robot control by further expanding the aforementioned cerebellar cobot controller with the addition of spinal circuits and muscle dynamics.

The integration of the cerebellar SNN with the spinal cord model has been possible in the cooperation framework provided by the HBP, which has allowed an international collaboration with the Biorobotics Laboratory, led by Professor Auke Ijspeert, at École Polytechnique Fédérale de Lausanne (EPFL), Switzerland. The Biorobotics Laboratory research interests cover the computational aspects of locomotion control, sensorimotor

## 1. Introduction

coordination, and learning, with recognized expertise in spinal circuits. Their work takes inspiration from animals to design new robots and control methods, as well as using robots to study the neural mechanisms involved in animal movement control and learning. This collaboration allowed the doctoral candidate to do a research stay at the Biorobotics Laboratory, from September to December 2021.

## 2. CEREBELLAR ADAPTIVE MOTOR CONTROL

The work presented through this chapter can be found in:

*Abadía, I., Naveros, F., Garrido, J. A., Ros, E., & Luque, N. R. (2019). On robot compliance: a cerebellar control approach. IEEE Transactions on Cybernetics, 51(5), 2476-2489.*

*DOI: 10.1109/TCYB.2019.2945498*

### 2.1 Introduction

Physical HRI implies robots operating in complex unstructured environments in which human actions cannot be modeled, demanding robots to adapt their behavior to unpredicted situations in a safe and autonomous manner, i.e., compliant human-like behavior [71]. Compliance demands control over the torque values put in place to achieve a given movement, but traditional torque control methods based on dynamics modeling cannot be efficiently applied to HRI involving cobots, as the nonlinearities of elastic components make detailed modeling extremely complex [17]. New torque control strategies are required.

Controlling biologically inspired robots (i.e., elastic components and low-power actuators emulate biological actuation carried by muscles, tendons and ligaments) shall directly benefit from understanding biological motor control itself. The control mechanisms encountered in biology are involved in a continuous learning process to cope with the complexity and variations in body structure and dynamics. AI can be used to replicate this learning process. In particular, widely used ANNs have been proposed and tested as a solution for the control of these compliant robots without requiring prior knowledge of the robot dynamics [17], [72]. ANNs are vaguely inspired in the functioning of biological neural networks. Their lack of ability for carrying neural information via well-timed sequences of spikes prevents them from serving as the linkage between biological neural coding and movement coordination, thus sidelining any attempt to draw biological analogies. SNNs, on the other hand, offer a more biologically plausible approach [61], and their intrinsic temporal characteristics [64] make them a suitable solution for dealing with robot inner dynamics in torque control.

## 2. Cerebellar adaptive motor control

Among the several areas of the CNS contributing to motor control of the musculoskeletal system, the cerebellum stands out among them by its role in the integration, regulation, and coordination of motor processes, and more importantly, motor learning [40], [43], [73], [74]. Diverse computational models of the cerebellum have been proposed in robot control: CMAC [75], APG [76], MPFIM [77], or the Schweighofer-Arbib model [78] are some examples. Among these models, the so-called *cellular-level* models [35] (SNN based), account for the most detailed approach as they capture the biophysical features of cerebellar neuronal processing. These models enable a realistic implementation of the biological understanding of the cerebellum. The cerebellar controller presented in this thesis belongs to this cellular-level family and its very nature allows taking advantage of previous neurocomputational and *in vivo* studies, which deepen in diverse aspects of cerebellar structure and functionality:

- Spike-timing-dependent plasticity (STDP) mechanisms can correlate the actual and desired motor states toward the generation of accurate corrective commands, even in the presence of sensorimotor delays [79].
- Synaptic adaptation at the cerebellar cortex: granular [80], [81], and molecular layer [82].
- Suitability of cerebellar control under dynamic and kinematic perturbations [52], [83], [84].
- Existence of structural and functional cerebellar microcomplexes [85].
- Effective representation of neural input states for a supervised-learning cerebellar network [86].
- Granular layer timing mechanisms [87], information processing [88], and multimodal sensory inputs [89].

Here, we conjugate the following elements to address the need for adaptive torque control in cobots: i) a hardware compliant robot (i.e., Baxter); ii) an SNN modeling the highly regular neural structure and learning mechanisms of the cerebellum. Addressing this problem implies facing the state-of-the-art challenges described below.

First, to control a real cobot (i.e., not a simulated one), the cerebellar-like SNN is required to operate in RT. Spiking neural processing in RT is a highly demanding computational task. Considering that our computational resources are limited, there must be a trade-off between network size, neuron complexity, network topology, and temporal output resolution, which will determine the RT capability and also, to some extent, the motor control accuracy. We used EDLUT simulator to accommodate, for the first time, a RT cerebellar SNN consisting of ~62K leaky integrate and fire (LIF) neurons with ~36.4M synapses, 36M of which endowed with plasticity.

Second, we need to implement an effective RT dialogue between the cerebellar spike domain and the analog sensorimotor domain [29], [90], [91]. In closed-loop, the cerebellar SNN must generate the adequate output motor commands to control the robot

movement as a response to the input stimuli. The spike and analog domains are brought together using a set of analog to spike, and spike to analog translations (see 2.2.6 and 2.2.7), operating in RT without compromising motor accuracy.

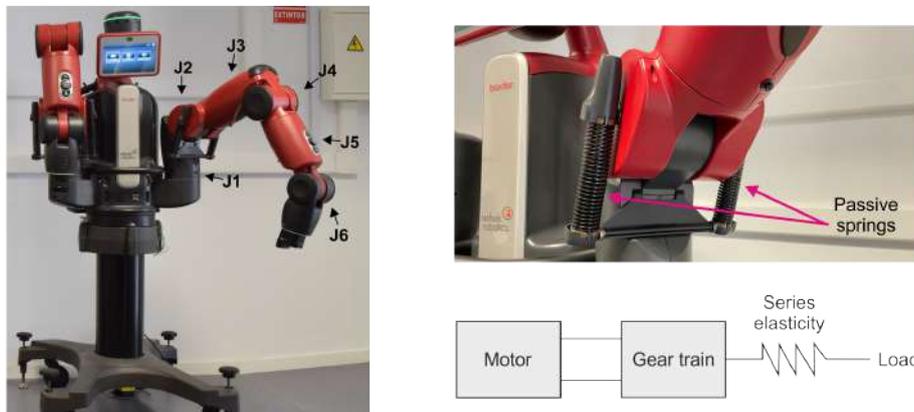
Third, the cerebellar SNN torque controller must cope with hardware compliance impositions. We used the Baxter robot, equipped with passive compliance, and performed direct torque control to address compliant interactions with an unstructured environment [69]. The cerebellar controller must be able to compensate for Baxter’s loss in precision and lower capacity to exert a force due to its hardware passive compliance. The SNN must be able to continuously learn the minimal torque values needed to execute several motor tasks in RT under changing operational and environmental conditions, i.e., perturbation forces that continuously modify their module and direction, human collisions, and interactions.

Finally, we need to validate the performance of the implemented solution. We have provided a cerebellar SNN able to learn the adequate torque values in a safe manner, outperforming the control accuracy provided by the factory-default position controller.

## 2.2 Methods

### 2.2.1 Compliant robot – Baxter

Baxter robot, manufactured by Rethink Robotics [70], is a two-armed collaborative robot, each arm counting with seven DOFs. Baxter is inherently compliant thanks to its series elastic actuators (SEAs) [92]. SEAs interpose a passive mechanical spring between the actuator and the load, low-pass filtering possible impacts, hence increasing shock tolerance [92], and providing force sensing capabilities. SEAs constitute a built-in mechanism that inherently allows for safe physical HRI. Besides the elasticity provided by SEAs, Baxter is also equipped with two external passive springs located at joint  $J2$  (see Fig. 2.1).

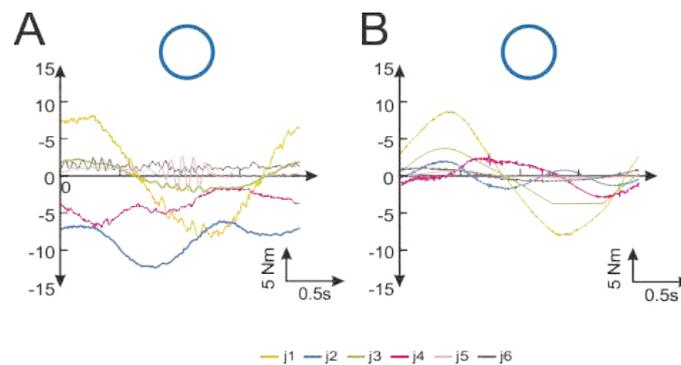


**Figure 2.1. Baxter robot.** Full body of the Baxter robot, the tags indicate the six joints of the left arm used within this dissertation. The right column shows the elastic components of Baxter: passive springs and a schematic representation of Series Elastic Actuators (SEAs), reproduced from [92].

## 2. Cerebellar adaptive motor control

Baxter implements a factory-default position controller, as well as allowing for torque control of the different joints. The factory-default position controller was used as reference for motor accuracy to evaluate the performance of the cerebellar SNN torque controller. The nonlinearities introduced by elastic components and the torque control capability make Baxter perfectly suitable for the research questions addressed within this dissertation.

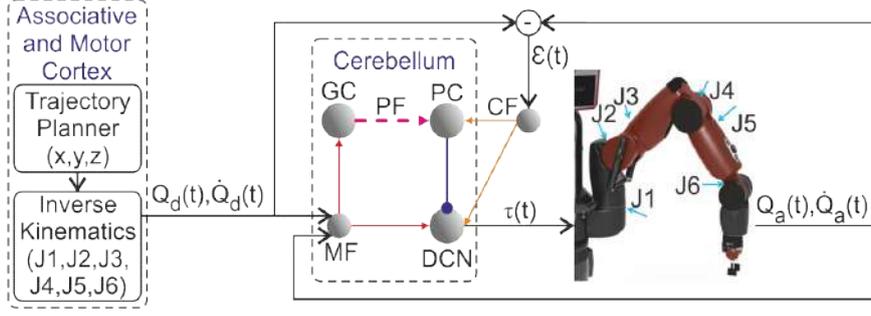
Prior to Baxter’s hands-on testing, we used the simulated version of Baxter available in Gazebo [93] as a safe environment to develop and test the robot-cerebellum communication interface, which allowed sending motor commands to the robot (a torque value per joint) and receiving sensorimotor information (position and velocity per joint). This interface was developed using Robot Operating System (ROS) [94] and effectively allowed for control of both the simulated and real robot. Gazebo was only used to test the communication setup, as the dynamics of the simulated and the real robot are significantly different (Fig. 2.2). The implemented ROS frameworks used in the different chapters of this thesis were made available after publication (see each chapter’s corresponding paper).



**Figure 2.2. Dynamics of the real robot vs. simulated robot.** The figure depicts the torque commands to perform a circular trajectory generated by the same PD controller for both the real robot (A), and the Gazebo simulator version (B). Besides series elastic actuators present at every joint, Baxter also holds a couple of passive springs at joint  $j2$ ; i.e., nonlinear elements which the model fails to accurately represent. The larger mismatch in  $j2$  is extended to other joints via inertial forces.

### 2.2.2 Cerebellar control loop

The implemented ROS framework allowed establishing an effective dialog between Baxter and the cerebellar network, exchanging sensorimotor information. This dialog was framed within a closed control loop with negative feedback, with a working frequency of 500 Hz (Fig. 2.3). The cerebellar SNN acted as the controller and computed a torque command at each time step (2 ms) to achieve the goal behavior, which was defined as desired position and velocity per joint at each time step.



**Figure 2.3. Schematic of the cerebellar closed control loop.** The cerebellar controller received the following input signals: desired trajectory (position and velocity per joint,  $Q_d$  and  $\dot{Q}_d$ ); actual state of the joints (position and velocity,  $Q_a$ , and  $\dot{Q}_a$ ), and a teaching/error signal per joint ( $\epsilon$ ). The cerebellar controller consisted of five different neural populations: mossy fibers (MFs), granule cells (GCs), Purkinje cells (PCs), climbing fibers (CFs), and deep cerebellar nuclei cells (DCN). The activity of the DCN cells (output cerebellar layer) was translated into a torque command per joint ( $\tau$ ) sent to the robot. The control loop operated at 500 Hz.

The SNN torque controller computed the neural activity using the following input information: desired robot state (desired trajectory position,  $Q_d$ , and velocity,  $\dot{Q}_d$ , per joint), actual robot state (actual position,  $Q_a$ , and velocity,  $\dot{Q}_a$ , per joint), and a teaching/error signal. The desired trajectory input signals were provided by a trajectory generator module representing the role of the motor cortex and other motor areas [95], [96], [97]. The desired trajectories were first designed in Cartesian space defining 3D position and orientation for the end-effector, and then translated into joint space using *Moveit!* software [98]. This inverse kinematics process was conducted offline, allowing the pre-computation of joint space trajectories later on used by the trajectory generator module, which provided the online controller input signals. The actual robot sensorimotor state was provided by Baxter’s sensors and then mapped into actual robot state input signals for the controller. The teaching/error signal ( $\epsilon$  per joint) was obtained by comparison of the desired trajectory and the actual robot state. A single teaching/error signal per joint comprised the weighted sum of the corresponding position and velocity joint errors, as follows:

$$\epsilon(t) = k_{pos} [Q_d(t) - Q_a(t)] + k_{vel} [\dot{Q}_d(t) - \dot{Q}_a(t)] \quad (2.1)$$

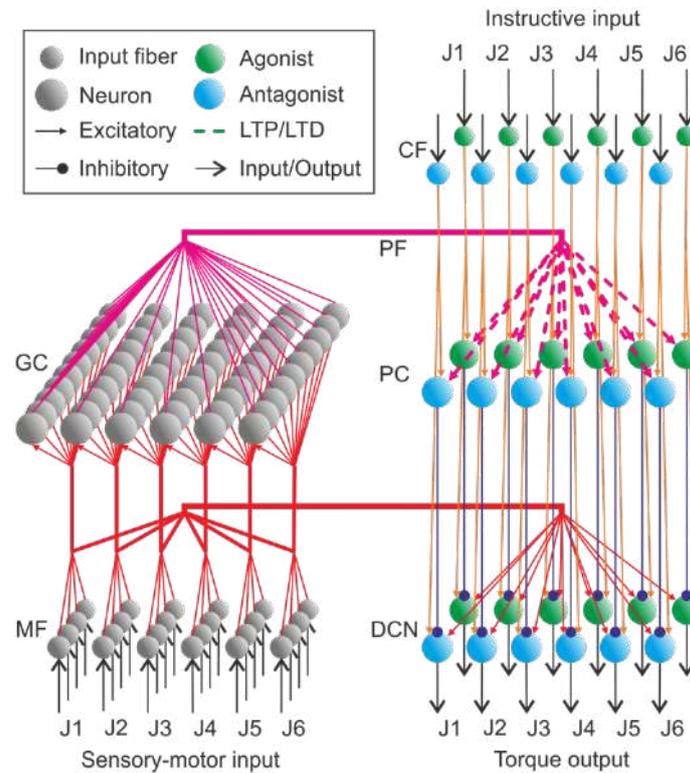
where  $k_{pos}$  and  $k_{vel}$  correspond to the position and velocity error gains, with the following values from joint 1 to 6:  $k_{pos} = (1.5, 2.0, 3.0, 2.0, 3.0, 3.0)$ ,  $k_{vel} = (1.5, 1.0, 3.0, 1.0, 3.0, 0.5)$ . Once the cerebellar network computed a torque command ( $\tau$ ) it was sent to the robot, inducing movement to the arm. Consequently, the cerebellar network input sensory information was modified, thus, closing the loop. Cerebellar input and output signals were updated every 2 ms (500 Hz), guaranteeing low latency and continuous communication, a mandatory requirement for RT robot control.

The cerebellar model ran in the SNN simulator EDLUT [66], [67], [68], specially designed to perform embodiment experimentation thanks to its ability to slow down or speed up neural computation to cope with RT requirements.

## 2. Cerebellar adaptive motor control

### 2.2.3 Cerebellar controller – The neural network

The cerebellar network consisted of five neural layers: i) mossy fibers (MFs), ii) granule cells (GCs), iii) climbing fibers (CFs), iv) Purkinje cells (PCs), and v) deep cerebellar nuclei (DCN), see Fig. 2.4. The cerebellar network was divided into six microcomplexes [85], each one focusing on controlling one of the six DOF.



**Figure 2.4. Cerebellar network scheme.** Schematic representation of the main neural layers, cells, connections, and plasticity site considered in the cerebellar model. The mossy fibers (MFs) convey the sensory signals while the climbing fibers (CFs) convey the teaching/error signals, providing inputs to the cerebellar network. MFs project sensorimotor information onto granule cells (GCs) and deep cerebellar nuclei cells (DCN). GCs, in turn, project onto Purkinje cells (PCs) through parallel fibers (PFs). PCs also receive excitatory inputs from the CFs. Finally, DCN cells receive excitatory inputs from MFs and CFs, and inhibitory inputs from PCs. DCN cells drive the cerebellar output torque commands.

The MFs constituted the input layer through which the input sensorimotor information was conveyed toward the inner cerebellar network layers: actual and desired robot state (position and velocity per joint) translated into spiking patterns (see 2.2.6 for analog to spike translation). The MFs activity was projected through excitatory afferents to GCs, which recoded this sensorimotor information into somatosensory neural activity that was later propagated to the PCs through the parallel fibers (PFs), i.e., excitatory axons of the GCs. The PC layer also received, via CFs (i.e., excitatory axons of the inferior olive, IO), the teaching/error signal: mismatch between the desired and actual robot state translated into spikes. Last, the DCN layer received inhibitory synapses from PCs and excitatory synapses from MFs and CFs. The DCN neural activity was translated into analog motor commands sent to the robot (see 2.2.7 for spike to analog translation),

thus closing the loop. 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 (conveyed through MFs and recoded at GCs) and the teaching/error signal (conveyed through CFs). Synaptic plasticity allowed for error reduction through iterative trial-and-error executions of each motor task. The regulation of PCs activity shaped the DCNs activity through the PC-DCN inhibitory synapses, thus shaping the cerebellar output activity to minimize the error in the execution of the motor task.

The cerebellar neural network distribution was the following (topology summarized in Table 2.1):

- MFs were modeled as 240 input fibers able to propagate the sensorimotor information toward GCs and DCN at each time step (2 ms). These MFs conveyed in spike form the input analog signals (desired and actual joint position and velocity,  $Q_d, \dot{Q}_d, Q_a, \dot{Q}_a$ ). The 240 MFs were divided into six groups of 40 fibers each, i.e., one group per DOF. Each MF group was in turn subdivided into four equal subgroups (10 MFs), each one directed to mapping one of the four input signals ( $Q_d, \dot{Q}_d, Q_a, \dot{Q}_a$ ). At each time step, only four non-overlapped MFs per group were active, representing the input neural state.
- GCs were modeled as 60000 LIF neurons emulating a state generator [82], [87], [88]. These 60000 GCs were organized in six groups of 10000 neurons each, i.e., one group per DOF. Each GC received four input synapses [89] coming from the MF group dedicated to the very same DOF. The connectivity pattern between MFs and GCs was designed in a way that non-overlapped GC neural activation could univocally represent all possible MF neural input combinations. The granular and molecular layers operating as a state generator transform into non-overlapped spatiotemporal patterns of neural activations the sensorimotor neural information received by each GC from both external sources (MFs) and other interneurons (Golgi cells, Lugaro cells, unipolar brush cells, etc.). The granular layer, then, acts as a reservoir of interacting spiking neurons within a recurrent topology, whereas the subsequent PC neuron acts as a readout layer. This state generator assumption allowed us to merge granular and molecular layers into one “granular layer” (see [99] for further details). The MF-GC connectivity pattern facilitated the generation of these spatiotemporal neural states and their readout layer.
- CFs were modeled as 600 input fibers able to propagate the teaching/error signal (mismatch between desired and actual robot state) toward PCs and DCN. These 600 fibers were organized in six microcomplexes of 100 CFs each, i.e., one microcomplex per DOF. Each microcomplex was also divided into two symmetric subgroups, each one dedicated to controlling the clock or anticlockwise movement of the robot joint actuator (emulating the agonist-antagonist interplay of human muscles). A probabilistic Poisson process transformed the analog error (obtained when comparing the actual and desired

## 2. Cerebellar adaptive motor control

trajectories of each joint) into CF spiking neural activations. Each CF spike encoded well-timed information regarding the instantaneous error. The probabilistic spike sampling of the error ensured a proper representation of the entire error region over trials, while maintaining the CF activity between 1 and 10 Hz per fiber (similar to electrophysiological data [100]). The error evolution could be sampled accurately even at such a low frequency [79], [101].

- PCs were modeled as 600 LIF neurons. These 600 PCs were organized in six microcomplexes of 100 neurons each, i.e., one microcomplex per DOF. Each microcomplex was also divided into two symmetric subgroups, each one dedicated to controlling the clock or anticlockwise movement of the robot joint actuator. Each PC was connected to all PFs (excitatory axons of GCs), thus receiving the sensorimotor information concerning all joints at once. CFs and PCs were one-to-one connected, maintaining the six microcomplex architecture. Thus, each PC microcomplex received the same sensorimotor information via PFs, but a different teaching/error signal through its corresponding CFs microcomplex. Correlating these two different sources of neural information allows each PC microcomplex to adapt the cerebellar input-output response of each DOF via a plasticity mechanism that modified the overall PF synaptic weight distribution (see 2.2.5 for synaptic plasticity).
- DCN cells were modeled as 600 LIF neurons. These 600 neurons were organized in six microcomplexes of 100 neurons each, i.e., one per DOF. Each microcomplex was also divided into two symmetric subgroups, each one dedicated to controlling the clock or anticlockwise movement of the robot joint actuator. Each DCN neuron was innervated by an inhibitory afferent from a PC, and an excitatory afferent from the CF which simultaneously innervated the same PC, preserving the six microcomplex architecture. Each DCN neuron also received excitatory connections from all MFs, which provided the baseline DCN activity.

**Table 2.1. Cerebellar neural network topology.** Dash entries indicate not applicable.

Neurons		Synapses			
Pre-synaptic cells	Post-synaptic cells	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	1.6	[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	-

Neuron and synapse numbers were limited by the computing resources available, following a trade-off between RT capability and input-output control signals resolution. The connectivity ratios (divergence and convergence ratios) between cerebellar layers were preserved according to MF-GC, GC-PC [102], MF-DCN, PC-DCN, CF-PC, and CF-DCN [103], [104], [105]. The PC-DCN smaller convergence ratio was compensated by using higher synaptic weight values, thus maintaining the input drive.

#### 2.2.4 Spiking neuron models

The cerebellar neural network consisted of LIF neurons [62] due to their minimal computational cost in spike generation and processing, a key factor in RT computation. Our LIF neurons elicited a spike once the corresponding membrane potential reached a certain threshold and, immediately after, the membrane potential was reset. The LIF neural dynamics was defined by the membrane potential and the excitatory (AMPA and NMDA) and inhibitory (GABA) chemical conductance as follows:

$$C_m \cdot \frac{dV}{dt} = I_{internal} + I_{external} \quad (2.2)$$

$$I_{internal} = -g_l \cdot (V + E_L) \quad (2.3)$$

$$I_{external} = -\left(g_{AMPA}(t) + g_{NMDA}(t) \cdot g_{NMDA\_INF}\right) \cdot (V - E_{AMPA}) - g_{GABA}(t) \cdot (V - E_{GABA}) \quad (2.4)$$

$$g_{AMPA}(t) = g_{AMPA}(t_0) \cdot e^{\frac{t-t_0}{\tau_{AMPA}}} + \sum_{i=1}^N \delta_{AMPA_i}(t) \cdot w_i \quad (2.5)$$

$$g_{NMDA}(t) = g_{NMDA}(t_0) \cdot e^{\frac{t-t_0}{\tau_{NMDA}}} + \sum_{i=1}^N \delta_{NMDA_i}(t) \cdot w_i \quad (2.6)$$

$$g_{GABA}(t) = g_{GABA}(t_0) \cdot e^{\frac{t-t_0}{\tau_{GABA}}} + \sum_{i=1}^N \delta_{GABA_i}(t) \cdot w_i \quad (2.7)$$

$$g_{NMDA\_INF} = \frac{I}{1 + \exp(62 \cdot V) \cdot \frac{1.2}{3.57}} \quad (2.8)$$

where  $C_m$  denotes the membrane capacitance;  $V$  is the membrane potential;  $I_{internal}$  is the internal current; and  $I_{external}$  is the external current.  $E_L$  is the resting potential and  $g_l$  the conductance responsible for the passive decay term toward the resting potential. Conductances  $g_{AMPA}$ ,  $g_{NMDA}$ , and  $g_{GABA}$  integrate all the contributions received by each receptor type (AMPA, NMDA, and GABA) through individual synapses, being  $g_{NMDA\_INF}$  the NDMA activation channel. These conductance terms were defined as decaying exponential functions [66], [62], with their values 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

## 2. Cerebellar adaptive motor control

reset to  $E_L$  during the refractory period ( $T_{ref}$ ). The configuration parameters for the neuron models are shown in Table 2.2.

**Table 2.2. Neuron models parameters.** Dash entries indicate not applicable.

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
$\tau_{AMPA}$ (ms)	1.0	1.2	0.5
$\tau_{NMDA}$ (ms)	-	-	14.0
$\tau_{GABA}$ (ms)	-	-	10.0
$V_{thr}$ (mV)	-50.0	-52.0	-40.0
$T_{ref}$ (ms)	1.0	2.0	1.0

### 2.2.5 Synaptic plasticity

The adaptive motor process of the cerebellar network was implemented through an STDP mechanism located at PF-PC synapses. This STDP mechanism balanced long-term potentiation (LTP) and long-term depression (LTD) as follows:

$$LTP \Delta w_{PF_j-PC_i}(t) = \alpha \cdot \delta_{PF_{spike}}(t) \cdot dt \quad (2.9)$$

$$LTD \Delta w_{PF_j-PC_i}(t) = \beta \cdot \int_{-\infty}^{t_{CF_{spike}}} k(t - t_{CF_{spike}}) \cdot \delta_{PF_{spike}}(t) \cdot dt \quad (2.10)$$

where  $\Delta w_{PF_j-PC_i}(t)$  denotes the synaptic weight change between the  $j^{th}$  PF and the target  $i^{th}$  PC;  $\alpha = 0.002$  nS is the synaptic efficacy increment;  $\delta_{PF}$  is the Dirac delta function corresponding to an afferent spike from a PF;  $\beta = -0.001$  nS is the synaptic efficacy decrement; and the kernel function  $k(x)$  is defined as:

$$k(x) = \begin{cases} \frac{-(x + d_k)}{\tau_{LTD} - d_k} \cdot e^{\frac{x+d_k}{\tau_{LTD}-d_k}-1} & \text{if } x < -d_k \\ 0 & \text{if } x \geq -d_k \end{cases} \quad (2.11)$$

where  $\tau_{LTD} = 100$  ms is the time constant that is aligned with the biological sensorimotor pathway delay ( $\sim 100$  ms), i.e., the time period elapsed from the sensory information reception to information transmission along nerve fibers, neural processing time responses, and the final motor output response [106].  $d_k = 0.07$  s allows for the

adjustment of the kernel width. The kernel maximum value,  $k(x) = 1$ , is obtained when  $x = -\tau_{LTD}$ , and zero or close to zero when  $x > -dk$  or  $x < -\tau_{LTD} - 10 \cdot (\tau_{LTD} - dk)$ .

The STDP rule defined by Eq. 2.9 – 2.11, caused a fixed synaptic efficacy increment (LTP) each time a spike arrived through the PFs to the target PC and a variable synaptic efficacy decrement (LTD) each time a spike arrived through a CF to the target PC. The amount of synaptic decrement depended on the activity that arrived through the PFs prior to the CF spike. Both activities were convolved using the integrative kernel defined in Eq. 2.11 and were multiplied by the synaptic decrement  $\beta$ . The effect on the presynaptic spikes arriving through PFs was maximal during the 100 ms time window ( $\tau_{LTD} = 100$  ms) before the postsynaptic CF spike arrival, thus accounting for the sensorimotor pathway delay [52], [79], [107].

The STDP mechanism correlated the neural activity patterns coming through the PFs toward PCs with the instructive signals coming from CFs toward PCs. This correlation process at PC level identified certain PF activity patterns codifying certain sensorimotor information and, consequently, diminished the PC output activity by a PF-PC synaptic weight reduction. A reduction on the PC activation caused a subsequent reduction on the PC inhibitory action over the target DCN. Conversely, in the absence of any correlation, the STDP mechanism increased the PC output activity by a PF-PC synaptic weight potentiation. Since the DCN were driven by a near constant baseline MF activation, a lack of PC inhibitory action would cause an increasing DCN activity whereas an incremental PC inhibitory action would do otherwise. Well-timed sequences of increasing/decreasing levels of DCN activation during the learning acquisition process ultimately shaped the cerebellar output activity and diminished the overall error.

### 2.2.6 Analog to spike translation

The SNN sensorial input information, originated as analog signals at Baxter's sensors ( $Q_a, \dot{Q}_a$ ) and the trajectory generator module ( $Q_d, \dot{Q}_d$ ), had to be translated into spiking neural activity (MFs activity) that the cerebellar network could process. Each DOF sensorial state was mapped by 40 MFs (i.e., a total of 240 MFs for all six DOF), which were again divided into four subgroups of 10 MFs each coding  $Q_d, \dot{Q}_d, Q_a$  and  $\dot{Q}_a$ , respectively. Each of the 10 neurons of the subgroup acted as a sensory receptor for a specific interval within the analog range of the joint; i.e., a neuron fired a spike,  $\delta_{MFspike}$ , when the analog value ( $Q$ ) was within its receptor interval ( $R_n$ ), described as follows:

$$\delta_{MFspike}(t) \leftrightarrow Q(t) \in R_n \quad (2.12)$$

$$R_n = c_n \pm w_n \quad (2.13)$$

$$c_n = r_{\min} + \left( \frac{r_{\max} - r_{\min}}{S - 1} \right) \cdot n \quad (2.14)$$

$$w_n = \frac{1}{2} \cdot \left( \frac{r_{\max} - r_{\min}}{S - 1} \right) \quad (2.15)$$

## 2. Cerebellar adaptive motor control

where  $\delta_{MFspike}$  defines the Dirac delta function of an afferent spike from one 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 for each input analog signal, and  $S = 10$  stands for the total number of neurons within the subgroup. Because 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.

The teaching/error signal,  $\varepsilon(t)$ , obtained by comparing desired ( $Q_d, \dot{Q}_d$ ) and actual robot state ( $Q_a, \dot{Q}_a$ ), was also translated from analog to spike domain (CFs activity). Each DOF had 100 CFs to code the error signal, symmetrically divided into two subgroups of 50 CFs devoted to the clock/anticlockwise sensed error, i.e., positive/negative joint error. The CFs chaotic and low firing rate, between 1 and 10 Hz per neuron [100], was replicated using a Poisson process: given the error signal  $\varepsilon(t)$  and a random number  $\eta(t) \in [0, 1]$ , the given CF fired a spike  $\delta_{CFspike} \leftrightarrow \varepsilon(t) > \eta(t)$ , remaining silent otherwise [79], [108].

### 2.2.7 Spike to analog translation

The DCN neural activity had to be translated into analog torque commands ( $\tau$ ) that could be applied by Baxter actuators. The activity of each of the six DCN microcomplexes, one per DOF, was transformed at every time step (2 ms) following the spike to analog translation described by:

$$DCN_{j,i}(t) = \int_{t-T_{step}}^t \delta_{DCN_{j,i}}(t) \cdot dt \quad (2.16)$$

$$DCN_j(t) = \sum_{i=1}^{N=50} DCN_{j,i}(t) - \sum_{i=51}^{N=100} DCN_{j,i}(t) \quad (2.17)$$

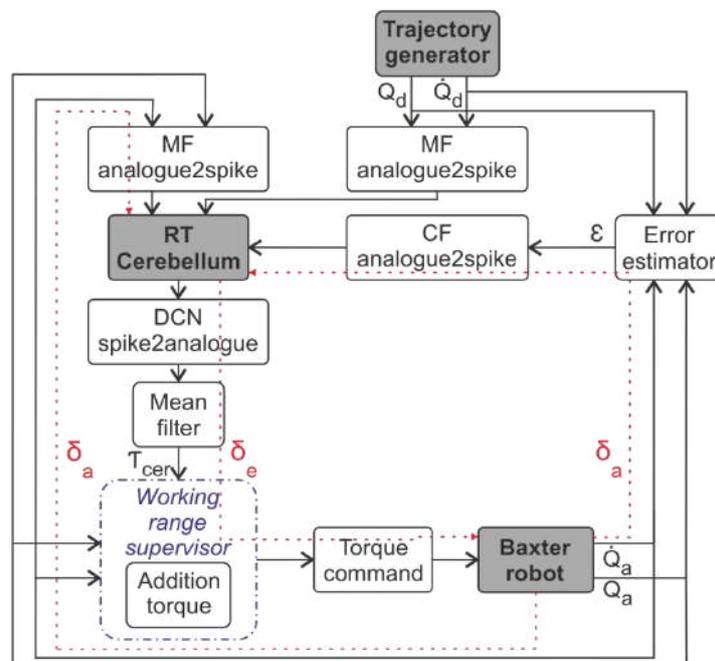
$$\tau_j(t) = \frac{\alpha_j}{15} \cdot \sum_{x=1}^{15} DCN_{output}(t - (x-1) \cdot T_{step}) \quad (2.18)$$

where  $j \in [1, 6]$  stands for the joint index;  $i \in [1, 100]$  defines the DCN neuron index within the microcomplex related to joint  $j$  (the first 50 DCN cells encoding the agonist movement, the last 50 DCN cells encoding the antagonist movement); and  $\delta$  stands for the Dirac delta function representing a spike event. The spike to analog conversion in Eq. 2.16 and 2.17 was then convolved with a 15-taps mean filter, Eq. 2.18, emulating the low-pass behavior of muscles. The final torque command per joint,  $\tau_j$ , was modulated by a factor  $\alpha$  to weight the DCN output according to the relative position, orientation, and mass of each joint,  $\alpha_j = (0.75, 1.0, 0.375, 0.5, 0.05, 0.05)$  Nm/spike.

## 2.2.8 ROS control loop implementation

The control loop consisted of three main elements: i) trajectory generator, ii) cerebellar SNN torque controller, iii) Baxter robot. The implementation and communication among these three elements were achieved using ROS, allowing modularity. Fig. 2.5 depicts the control loop diagram. Each block defines a ROS node and each black arrow represents a ROS topic used to communicate different ROS nodes using either analog signals or spike trains.

The control loop was designed accounting for the sensorimotor pathway delay ( $\sim 100$  ms) [109]. The 100 ms delay comprised the efferent and afferent delays ( $\delta_e = \delta_a = 50$  ms), depicted as dashed red arrows in Fig. 2.5. A motor command originated at time  $t$  was applied by the robot actuators at time  $t + \delta_e$  and its effect was sensed back at the cerebellar network at time  $t + \delta_e + \delta_a$ . The cerebellar plasticity mechanism described in 2.2.5 compensated for this sensorimotor delay.



**Figure 2.5. Detailed cerebellar closed control loop scheme.** The boxes represent ROS nodes, the gray boxes depicting the three main elements of the control loop. Each black arrow represents a ROS topic communicating different nodes. Red dashed arrows represent the sensorimotor pathway delay.

The control loop assisted the cerebellar controller in the generation of the torque commands to minimize the mismatch between the reference signal and the robot state (desired vs. actual joint position and velocity). Different ROS nodes were implemented to that end:

- The trajectory generator node generated the desired trajectory signals, while Baxter generated the actual robot state signals and executed the torque commands.

## 2. Cerebellar adaptive motor control

- The RT cerebellum node accommodated the SNN cerebellar controller implemented in EDLUT (imported to ROS as a C++ library). This node received, computed, extracted, and propagated the neural activity.
- Desired, actual, and instructive signals needed to be transformed into spike trains that the SNN could process. The MF and CF *analog2spike* nodes carried out this transformation.
- The error estimator node provided the cerebellar controller with the teaching/error signal needed for neural adaptation. The error estimator node required comparing desired and actual trajectories.
- The cerebellar output spiking signals needed to be transformed into analog commands that Baxter could process and apply. The DCN *spike2analog* node transformed the spike trains into torque commands, lately smoothed by a mean filter node.
- The torque command node closed the loop sending to Baxter the torque commands obtained from the mean filter node.
- The supervisor node was implemented as a safe mechanism mimicking mechanical brakes (watchdog). This supervisor maintained Baxter within a safe working range during the first stages of neural adaptation. Only at the event of any of the joints getting outside its working range, the supervisor node, implemented as a PD controller, added a corrective torque value to the cerebellar torque command to prevent damages.

All nodes were synchronized thanks to a reference time signal extracted from Baxter's internal clock running under the Network Time Protocol (NTP), providing RT time steps [110]. Each event, i.e., analogue signal or spike train, generated on a ROS node carried a time stamp indicating the event processing time to the subsequent node. Each target node incorporated an input buffer in which events were stored for later synchronous processing according to their time stamps. The RT cerebellum node, however, allowed asynchronous processing of the events stored at its input/output activity buffers thanks to the RT mechanism incorporated in EDLUT [29]. In the event of empty input buffers, the neural simulation was halted. In the event of an almost empty output buffer, the neural simulation was speeded-up (see [29] for an in-depth review on EDLUT RT neural simulation). Hence, the RT cerebellar node could deal with neural activity volleys encountered during the cerebellar simulation that could not be processed synchronously.

### 2.2.9 Benchmarking the cerebellar controller – Behavioral tasks

We drew inspiration from the cerebellar role in motor control and movement coordination to implement a novel control strategy for hardware compliant robots. It is thus appropriate to evaluate the performance of our cerebellar-like model in the field of

robot dynamics control under a set of different conditions. To this aim, we proposed the experimental evaluation through two trajectory families:

- A set of fast movements in smooth trajectories consisting of sinusoidal-like profiles for both position and velocity per joint. The end-effector described both circular and eight-like Cartesian trajectories in the horizontal plane [84], [111]. These trajectories are well suited for revealing the complex dynamics of a 6 DOF robot arm [112], including interaction forces to be compensated by the cerebellar controller [113].
- Reaching movements; that is, fast, ballistic arm movements with bell-shaped velocity profiles (s-curve), toward a target point [114]. Arm reaching movements are primarily used for characterizing cerebellar pathologies in human motor control by measuring the time to target and precision. Arm dynamics control is critical due to the constraint at stake when moving masses. A single-joint limb movement in fast de/acceleration causes motion in all other limb joints, thus arising interaction forces to be compensated by the cerebellum [113].

The circular trajectory in Cartesian space meets Eq. 2.19, while Eq. 2.20 describes the eight-like trajectory:

$$\left. \begin{aligned} x &= R \cdot \cos(2 \cdot \pi \cdot t \cdot 0.5) \\ y &= R \cdot \sin(2 \cdot \pi \cdot t \cdot 0.5) \\ z &= \alpha \end{aligned} \right\} t \in [0, 2]; \alpha = const \quad (2.19)$$

$$\left. \begin{aligned} x &= 0.5 \cdot R \cdot \sin(2 \cdot 2 \cdot \pi \cdot t \cdot 0.5) \\ y &= R \cdot \cos(2 \cdot \pi \cdot t \cdot 0.5) \\ z &= \alpha \end{aligned} \right\} t \in [0, 2]; \alpha = const \quad (2.20)$$

where  $R$  denotes a 12 cm radius which is halved for the  $x$  coordinate in Eq. 2.20 to keep the eight-like trajectory within the working space limits of the robot. The  $z = \alpha$  coordinate and the end-effector vertical orientation were kept constant to maintain the horizontal plane through the trajectories. Each trajectory lasted 2 seconds. Once the translation from Cartesian  $(x, y, z)$  to joint space positions  $(Q_I - \delta)$  was completed (see 2.2.2), the joint velocity profiles  $(\dot{Q}_I - \delta)$  were obtained as the position derivative over time.

Regarding the target reaching tasks, the center of the circle trajectory was the starting position. Eight different points along the circular trajectory perimeter constituted the reaching targets following an even distribution every  $\pi/4$  radians. As aforementioned, this task tested the controller through point-to-point multijoint movements with s-curve velocity profiles that provided fast acceleration/deceleration changes, i.e., ballistic movements. The subsequent high jerk values entailed high inertial forces to be compensated by the cerebellar controller. Each target-reaching movement lasted 2 seconds back and forth between the target and the central position, i.e., 1 second to reach the target and 1 second to go back to the central position.

## 2. Cerebellar adaptive motor control

These three different behavioral tasks provided a varying context to test the cerebellar network. For every task, the cerebellar network acquired the motor commands needed to achieve the desired goal behavior through learning. The learning process was accomplished through the repetition over time of a specified trajectory.

The performance evaluation was carried out comparing the goal and the actual behavior; i.e., desired and actual joint positions. The average difference constitutes the position mean absolute error (MAE), which is our performance evaluation metric following:

$$MAE_{joint} = \sum_{i=0}^K (Q_{i,desired} - Q_{i,actual}) \quad (2.21)$$

$$MAE = \frac{\sum_{j=1}^N MAE_j}{N} \quad (2.22)$$

where  $K = 1000$  denotes the number of samples of the 2 s trajectories; and  $N = 6$  stands for the number of joints. The MAE provided a numerical performance indicator for the quality of the cerebellar controller, allowing us to compare it against the factory-default position controller.

For reproducibility and comparative purposes, the experimental setup (benchmarking included), source code, and experimental results are available at: [https://github.com/EduardoRosLab/EDLUT\\_BAXTER](https://github.com/EduardoRosLab/EDLUT_BAXTER).

## 2.3 Results

We tested our cerebellar-like controller in different behavioral tasks, considering the factory-default position controller as a performance baseline to validate the results using the aforementioned set of trajectories as cerebellar benchmarking. We completed the validation with a set of interactions in an unstructured environment to test compliance.

### 2.3.1 Circular trajectory

The first behavioral task consisted of following a 12 cm radius circular path in the horizontal plane ( $xy$ ) repeated over time to facilitate learning and adaptation, each trial with a duration of 2 s. The STDP mechanism governing the learning process modulated the cerebellar output (see 2.2) driving the robot behavior toward the goal. The behavioral evolution through time is illustrated in Fig. 2.6. Three snapshots were taken at different stages of the cerebellar learning process: initial, intermediate and final stage.

- Initial learning stage. The cerebellar model started learning from scratch. At an initial stage (Fig. 2.6, left column) the synaptic adaptation mechanism at PF-PC synapses that correlated the somatosensory information with the instructive signal was not effectively deployed yet. Thus, the inhibitory action from PCs onto DCN was of marginal utility; making the DCN output activity saturated as

it solely responded to the excitation coming from MF and CF afferents (Fig. 2.6 a, first row). Consequently, the corresponding initial torque commands (Fig. 2.6 a, second row) were far from leading the robot toward the desired goal (Fig. 2.6 a, third row; and d). As depicted in Fig. 2.6 d, the density function generated from 10 trials before  $t_1$  snapshot (Fig. 2.6, left column) reveals that the robot was still exploring the working area, performing low consistent, dispersed movements.

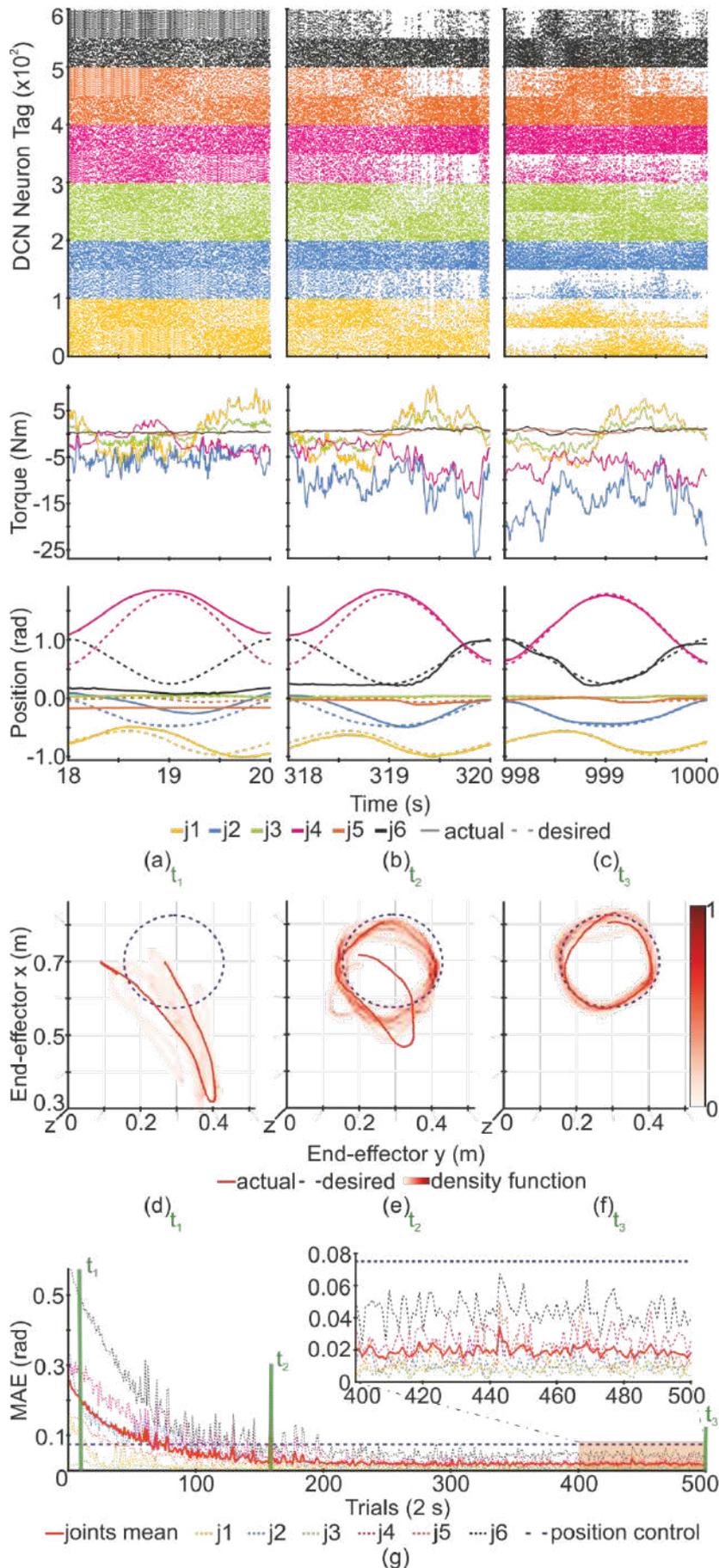
- Intermediate learning stage. At an intermediate stage (Fig. 2.6 central column) the synaptic adaptation allowed the recognition of some somatosensory patterns at the PCs, reflected in an emerging differentiated DCN activity between agonist and antagonist subgroups at each microcomplex (Fig. 2.6 b, first row). Consequently, the robot behavior began getting closer to the desired goal (Fig. 2.6 b, third row; and e).
- Final learning stage. Once the learning process reached advanced stages (Fig. 2.6 right column) the robot executed the desired trajectory with minimal error. The agonist/antagonist DCN activity was clearly differentiated at each microcomplex (Fig. 2.6 c, first row), and translated into the required torque commands via a spike to analogue translation (see 2.2.7). The synaptic adaptation process was reflected in a clear evolution of the torque values compared to previous stages, directly affecting the robot output behavior. All joints closely followed the desired trajectory at this stage (Fig. 2.6 c, third row) and, consequently, the end-effector barely missed at describing the desired circular path (Fig. 2.6 f), having a consistent behavior around the goal trajectory over trials.

The overall performance through the learning process is depicted in Fig. 2.6 g; illustrating how the cerebellar controller performance was improved as adaptation and learning were fulfilled. MAE evolution indicates that the cerebellar controller needed about 300 trials (i.e., 600 seconds) to converge, outperforming the accuracy of the factory-default position controller baseline ( $0.019 \pm 0.003$  vs.  $0.077 \pm 0.0004$  rad, Table 2.3).

**Table 2.3. Circular and eight-like trajectories: position MAE (mean and std) at different learning stages.**

		Cerebellar torque controller at [trials]			Factory-default position controller at [trials]
		[0-100]	[100-200]	[400-500]	[0-500]
MAE (rad)	○	$0.115 \pm 0.055$	$0.036 \pm 0.013$	$0.019 \pm 0.003$	$0.077 \pm 0.0004$
	∞	$0.111 \pm 0.034$	$0.046 \pm 0.013$	$0.017 \pm 0.003$	$0.063 \pm 0.0003$

## 2. Cerebellar adaptive motor control



**Figure 2.6. Behavioral evolution through circle trajectory trials (2 s).** (a) Initial learning stage ( $t_1=18-20$  s). (b) Intermediate learning stage ( $t_2=318-320$  s). (c) Final learning stage ( $t_3=998-1000$  s). The first row depicts the cerebellar output activity (DCN layer), whereas the second row shows its analog conversion into torque commands. The third row illustrates the desired vs. actual trajectory per joint. (d), (e), and (f) reveal the desired vs. actual trajectory of the end-effector in Cartesian space at  $t_1$ ,  $t_2$ , and  $t_3$  respectively, along with the density functions corresponding to the performed trajectories of the prior 10 trials. (g) Represents the position Mean Absolute Error (MAE) per trial through the learning process. Comparison of the MAE of each joint and the mean of all joints with the factory-default position controller baseline performance.

### 2.3.2 Eight-like trajectory

The eight-like trajectory was concentric to the previously discussed circle-shaped; it had a “radius” of 12 cm and each trial lasted 2 seconds. In terms of robot dynamics, the eight-like trajectory was more demanding than the circular trajectory, as faster and steeper changes in velocity module and direction were required for trajectory completion [112]. Benchmarking control capacities were further increased by allowing the linear and angular velocity of Baxter’s end-effector to vary within the trajectory [111], [115], [116], [117]. Nonetheless, the obtained results were equally satisfying (see Table 2.3).

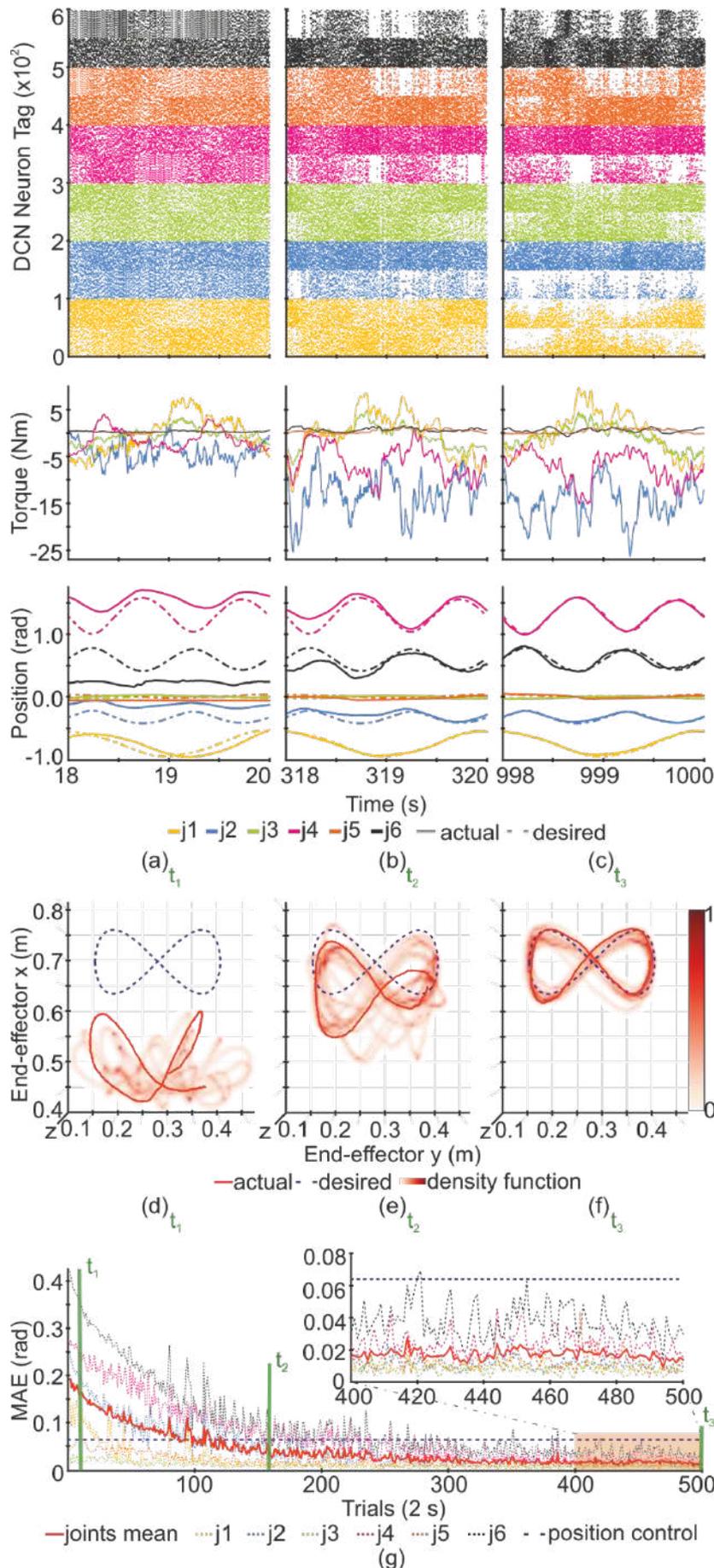
- Initial learning stage. At an early learning stage (Fig. 2.7, left column) the robot behavior was clearly far from the desired goal. DCN activity at this stage responded exclusively to the excitatory drive from MF-DCN and CF-DCN afferents, thus, it was saturated (Fig. 2.7 a, first row). The MAE value was high (0.165 rad) and the performed trajectory was far from the goal (Fig. 2.7 a, third row; d, and g).
- As learning progressed, the PF-PC synaptic adaptation mechanism began shaping the DCN activity causing an incipient neural activity differentiation between agonist and antagonist microcomplexes (Fig. 2.7 b, first row). In consequence, the corresponding torque values significantly differed from those of early stages (Fig. 2.7 b, second row), and the robot behavior began getting closer to the desired one (Fig. 2.7 b, third row; and e).
- Finally, once learning was fully deployed the robot behaved as desired (Fig. 2.7 c, third row; and f). The DCN activity was clearly sculpted to produce the needed torque commands to perform the desired trajectory (Fig. 2.7 c), maintaining a stable behavior over trials ( $0.017 \pm 0.003$  rad).

More demanding dynamics were introduced by the eight-like trajectory due to the velocity profile. Humans performing curved-profile hand motions show high linear velocities during segments of low curvature and low velocities during segments of high curvature [118]. The eight-like trajectory consisted of a combination of low/high curvature segments able to reveal the human performance in curved motions. We found that Baxter’s velocity profiles were biologically consistent, that is, the end-effector moved at high velocity when the trajectory curvature remained low and at low velocity when the trajectory curvature was high (see Fig. 2.8).

The eight-like velocity profile is given, ideally, by deriving the Cartesian trajectory (Eq. 2.20) as in:

$$\left. \begin{aligned} x' &= \pi \cdot R \cdot \cos(2 \cdot 2 \cdot \pi \cdot t \cdot 0.5) \\ y' &= -\pi \cdot R \cdot \sin(2 \cdot \pi \cdot t \cdot 0.5) \\ z' &= 0 \end{aligned} \right\} t \in [0, 2] \quad (2.23)$$

## 2. Cerebellar adaptive motor control

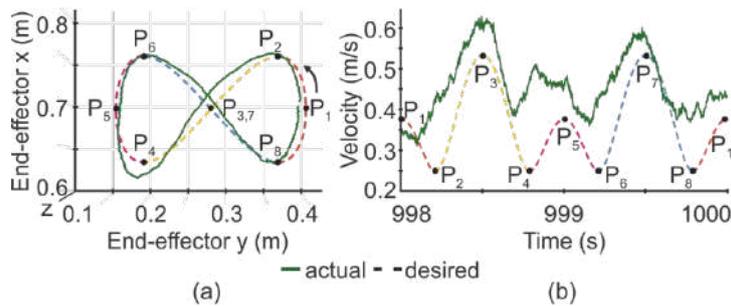


**Figure 2.7. Behavioral evolution through eight-like trajectory trials (2 s).** (a) Initial learning stage ( $t_1=18 - 20$  s). (b) Intermediate learning stage ( $t_2=318 - 320$  s). (c) Final learning stage ( $t_3=998 - 1000$  s). The first row depicts the cerebellar output activity (DCN layer), whereas the second row shows its analog conversion into torque commands. The third row illustrates the desired vs. actual trajectory per joint. (d), (e), and (f) reveal the desired vs. actual trajectory of the end-effector in Cartesian space at  $t_1$ ,  $t_2$ , and  $t_3$  respectively. Also the density functions corresponding to the prior 10 trials are depicted. (g) Represents the position Mean Absolute Error (MAE) per trial through the learning process. The MAE of each joint is illustrated as well as the average MAE of all joints, completed with the factory-default position controller baseline performance.

The linear velocity of Baxter’s end-effector is finally given by:

$$v = \sqrt{(x')^2 + (y')^2 + (z')^2} \quad (2.24)$$

The greater difficulty of the eight-like trajectory was noted in a lower convergence speed for the cerebellar controller to reach a stable behavior (Table 2.3 shows a slower MAE convergence speed compared to the circular trajectory). However, the final performance accuracy obtained also outperformed the factory-default position controller baseline ( $0.017 \pm 0.003$  vs.  $0.063 \pm 0.0003$  rad).



**Figure 2.8. Baxter’s end-effector position and linear velocity when performing the eight-like trajectory.** (a, b) Desired vs. actual position and linear velocity of the end-effector. Trajectory points, P<sub>1</sub> to P<sub>8</sub>, illustrate the relation high trajectory curvature – low velocity (P<sub>2</sub>, P<sub>4</sub>, P<sub>6</sub>, and P<sub>8</sub>) and low trajectory curvature – high velocity (P<sub>1</sub>, P<sub>3</sub>, P<sub>5</sub>, and P<sub>7</sub>) found in human hand curved motions [118]. Both graphs correspond to the end of the learning process (998 – 1000 s). Our adaptation process granted greater influence to position error when generating the instructive signal causing the velocity error to remain larger.

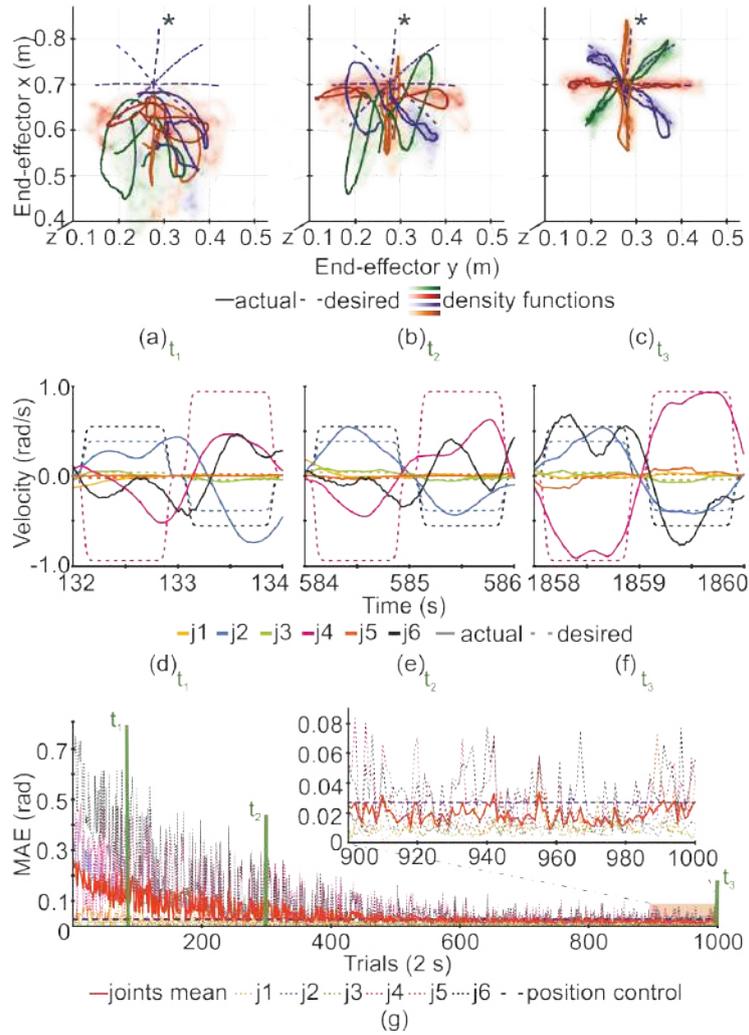
### 2.3.3 Target reaching

This task consisted of eight different reaching movements, sharing the same starting point. The challenge lies in the high speed of the movements and the randomness in the order of trials (transitions between the eight reaching movements were stochastic). The growth in complexity for the cerebellar controller was illustrated by a lower MAE convergence speed entailing higher standard deviation values inter trials and the need of more trials to reach stability compared to the two previous behavioral tasks (Table 2.4). Nevertheless, the cerebellar controller was able to perform these ballistic movements, improving its performance through learning and reaching again better accuracy than the factory-default position controller (Fig. 2.9) ( $0.019 \pm 0.006$  vs.  $0.026 \pm 0.006$  rad). Therefore, not only the cerebellar controller was able to perform accurate smooth trajectories but also fast-ballistic movements.

**Table 2.4. Target reaching: position MAE (mean and std) at different learning stages.**

	Cerebellar torque controller at [trials]			Factory-default position controller at [trials]
	[0-100]	[300-400]	[900-1000]	[0-1000]
MAE (rad)	$0.155 \pm 0.050$	$0.043 \pm 0.024$	$0.019 \pm 0.006$	$0.026 \pm 0.006$

## 2. Cerebellar adaptive motor control



**Figure 2.9. Behavioral evolution through target reaching trials (2 s).** Each trial consisted of one of the eight possible tasks. **(a)** Initial learning stage ( $t_1=158 - 160$  s). **(b)** Intermediate learning stage ( $t_2=598 - 600$  s). **(c)** Final learning stage ( $t_3=1998 - 2000$  s). **(a)**, **(b)**, and **(c)** depict the last performed trajectory for each of the eight possibilities in Cartesian space prior to  $t_1$ ,  $t_2$ , and  $t_3$  respectively. The density functions reveal the end-effector behavior over the last 80 trials, grouping the eight possible tasks by trajectory direction. **(d)**, **(e)**, and **(f)** show the velocity profiles related to the target reaching trajectory marked with \* in the Cartesian space. The illustrated trials correspond to the last \* iteration prior to  $t_1$ ,  $t_2$ , and  $t_3$  respectively. Note that the rising and lowering times achieved are consistent with human data ( $\sim 250$ - $500$  vs.  $\sim 400$ - $500$  ms) in 1 s target reaching movements [72]. **(g)** Represents the position Mean Absolute Error (MAE) per trial through the learning process. The MAE of each joint is illustrated as well as the mean MAE of all joints. High standard deviation values reflect how some reaching movements were more demanding than others. The position control baseline is the average MAE of the factory-default position controller under the same stochastic distribution over trials.

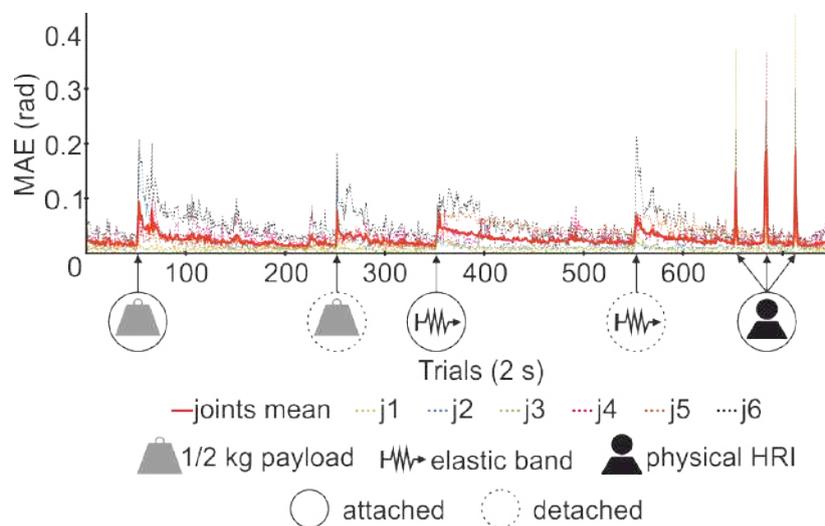
### 2.3.1 Unstructured interactions

Aiming at testing the compliance of the cerebellar controller, we tested its response in an unstructured environment. While performing the circular trajectory, some interactions were undertaken (Fig. 2.10). First, the dynamics of the robotic arm was modified in two different ways: i) adding a 0.5 kg payload to the end-effector attached to a rod, mimicking a pseudo “conical pendulum.” The tension force of the rod acting

on the robot varied with the alignment between the payload and the end-effector. ii) Attaching an elastic band to apply an elastic force that tried to return the band to its natural length. In both cases, the cerebellar-like controller successfully adapted to the new context after a learning period.

Subsequently, human interactions were performed: i) a human was able to move the robotic arm by applying an extremely low force (i.e., one-finger push); ii) a human grabbed the robotic arm and moved it through the working space with no opposition from the robot; iii) a human got in the way of the robotic arm trajectory with no risk of injury.

These results allow us to confirm that the cerebellar-like controller was able to accurately perform the desired trajectories, no matter the dynamics modifications; and guaranteed a safe human-robot interaction as no damages were suffered, at either human or robot side, when interrupting the robot task.



**Figure 2.10. Performance in an unstructured environment.** While performing the already learnt circular trajectory a set of unstructured interactions were undertaken: i) A 0.5 kg payload was attached to the end-effector and later on detached. ii) An elastic band was attached to the end-effector and later on detached. iii) A series of physical Human-Robot interactions. The figure depicts the position MAE through trials as interactions are undertaken, illustrating the cerebellar adaptation to unknown scenarios.

Within the supplementary material of the paper corresponding to this chapter, four movies are included to fully illustrate the cerebellar learning and adaptation process. The circular, eight-like, and target reaching trajectory movies show from up to down and left to right the following clips, all of them playing synchronized RT information: i) a frontal shot of the robot performing the trajectory; ii) the evolution of the position MAE per trial; iii) a nadir shot of the robot performing the trajectory; iv) the trajectory being performed by the end-effector in Cartesian space; v) the cerebellar output activity (DCN layer spikes); vi) the corresponding torque commands obtained from the spike to analog conversion of the DCN activity. Different cuts corresponding to an initial, intermediate, and final learning stage verify the behavioral evolution. Finally, the

## 2. Cerebellar adaptive motor control

unstructured environment movie shows the cerebellar adaptation and, therefore, robot adaptation, to unknown, unstructured scenarios; thus, proving compliance. These videos can be found at: <https://ieeexplore.ieee.org/document/8880621/media#media> .

### 2.4 Discussion

The role of the cerebellum in controlling human motor coordination has led and crystallized into different cerebellar modeling approaches in the robotics control field. According to their understanding of the cerebellar operation, these models can be categorized as follows [35]:

- Functional models. A functional understanding of specific cerebellar operations suffices to build these non-biological models. By setting aside any attempt to compel biological accuracy, these models offer the most computational friendly approach to provide motor learning and control. The MPFIM model [77] follows this approximation. Functional models can be found in [119].
- State-encoder driven models. These models operate as an abstraction of the granular and molecular layer of the cerebellum. They split up the state space assuming granule cells as on/off entities performing a mapping of inputs onto binary outputs. The CMAC [75], cerebellar adaptive filter [120], LWPR [121], or APG [76] models fall into this non-biologically plausible category, used in [122], [123], [124], [125], [126], [127], [128].
- Cellular-level models. A set of differential equations models each cerebellar neuron, simulating its biological behavior. These biological plausible models offer an insight into the cerebellar function on cellular level. The Schweighofer-Arbib model [78] is an example of this category. The computational cost of these models usually limits them to small-sized networks [29], non-RT simulated scenarios [99], [129], or low resolution output control signals [130].

Our SNN cerebellar torque controller fits in the cellular-level context. It stands out for its capacity to perform compliant robot control in RT, a feature that, to the best of our knowledge, was not accomplished before by means of a cerebellar SNN. The ~62 K neurons and 36 M synapses endowed with plasticity enable a detailed mapping of the sensorimotor space and the generation of precise output motor commands to achieve the goal behavior. Baxter's factory-default position controller was set as the baseline reference with the sole function of validating our controller performance.

### 2.5 Conclusion

Physical HRI implies controlling nonlinearities at the robot end, thus demanding adaptive control. In this chapter, taking inspiration from biology, we expand the family of RT adaptive controllers beyond machine learning [131], fuzzy logic [132], [133],

and ANN solutions [72], [134]. The intrinsic characteristics of SNNs, i.e., timing codification of evolving sensorimotor states, make them an appealing approach for motor control architectures [63], [64]. Here, we present a novel biologically plausible motor control architecture with a cerebellar-like SNN controller at its core that is able to drive a 6 DOF robot via torque commands in RT.

The implementation of a controller equipped with the main cerebellar plasticity mechanism (STDP) makes the availability of a detailed dynamic model of the robot dispensable. The cerebellar-like SNN is able to self-adapt and learn from scratch to control a given robot, making unnecessary any prior dynamics knowledge. Thus, the complexity of modeling nonlinear systems is tackled, and this SNN controller constitutes a plausible solution to control not only our Baxter robot, but any torque controlled robot. Previously achieved SNN position control [130], [135] does not provide compliance as physical perturbations or interactions are not supported; hence the importance of reliable torque control toward achieving safe physical HRI.

The variety of demanding behavioral tasks in terms of control requirements here accomplished proves our SNN cerebellar-like controller a valid solution. Our SNN controller succeeded in terms of position accuracy, high-speed movements, and compliance since the baseline performance (i.e., factory-default position controller) was utterly improved in all the experimental behavioral tasks.

## 2. Cerebellar adaptive motor control

# 3. MOTOR CONTROL UNDER NONDETERMINISTIC TIME DELAYS

The work presented through this chapter can be found in:

*Abadía, I., Naveros, F., Ros, E., Carrillo, R. R., & Luque, N. R. (2021). A cerebellar-based solution to the nondeterministic time delay problem in robotic control. Science Robotics, 6(58), eabf2756.*

*DOI: 10.1126/scirobotics.abf2756*

## 3.1 Introduction

Physical HRI must be safe for both human and robot, thus requiring compliant adaptive controllers for cobot operation. As already discussed, HRI can be compromised by contextual variables such as unstructured scenarios, unknown dynamics [71], or sensorimotor delays [136]. In previous chapters we have presented cerebellar SNNs as an effective approach for robot control providing both accuracy and compliance, key elements in safe HRI. Yet, the nondeterministic time delay control challenge was sidestepped, constituting the focus of this chapter.

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 [137]. 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 [22]; cloud-robotics, a growing field, relies robot control on remote cloud computing resources that lead to computation and transmission latencies within the control loop [21]; wireless communications carry additional time delays when compared to wired connections [20]. This 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

### 3. Motor control under nondeterministic time delays

decrease of the robotic system and a higher sensitivity as its static gain increases [136]. To stabilize time delayed systems, both adapted classic controllers and specifically designed controllers have been proposed [138]. 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 [139]; ii) parameter space method to tune the PID coefficients for an LTI system with time delays [140]; iii) using the Nyquist criterion to compute a set of PID controllers to stabilize a given n-order LTI system with time delay [141]. Unfortunately, these families of methods cannot be easily applied to HRI cobots whose dynamics are strongly nonlinear due to soft or elastic components [71]. Regarding the second category, it includes the dead-time compensators (DTC) [142], a family of controllers specifically designed for systems with time delays: i) Smith predictor based controllers [143], [144], only applicable when delays are constant [138]; ii) the finite spectrum assignment approach [145], [146]. However, DTC solutions strong dependence on the accuracy of the system model [138] makes them non-reliable for HRI control as the growing use of flexible-joints and elastic materials [15], [18] makes intractable the mathematical modeling of cobots nonlinear dynamics [17].

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 [106], [147]. 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  $\delta_e$ ) until its effect is sensed back at the cerebellum (afferent delay  $\delta_a$ ). These sensorimotor delays range from 100 to 150 ms approximately, with inter and intra individual variations [109]. To compensate them, the cerebellum acquires internal representations of the sensorimotor transformations needed to generate the motor commands to achieve a desired movement [148], and generates predictive motor commands by means of an STDP mechanism that correlates present and past sensorimotor signals, thus allowing motor learning even in the presence of sensorimotor delays [79]. Our cerebellar SNN controller, thanks to its biological plausibility, benefits from these CNS inherent features: it adopts the biological delays and mimics the cerebellar STDP mechanism.

In next sections, we present the evaluation of the performance of our SNN controller under time delays of different nature: steady and nondeterministic 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 affecting the transmission of sensorial information and motor commands, thus, applying an inherent feature of the CNS to a robotic control challenge.

## 3.2 Methods

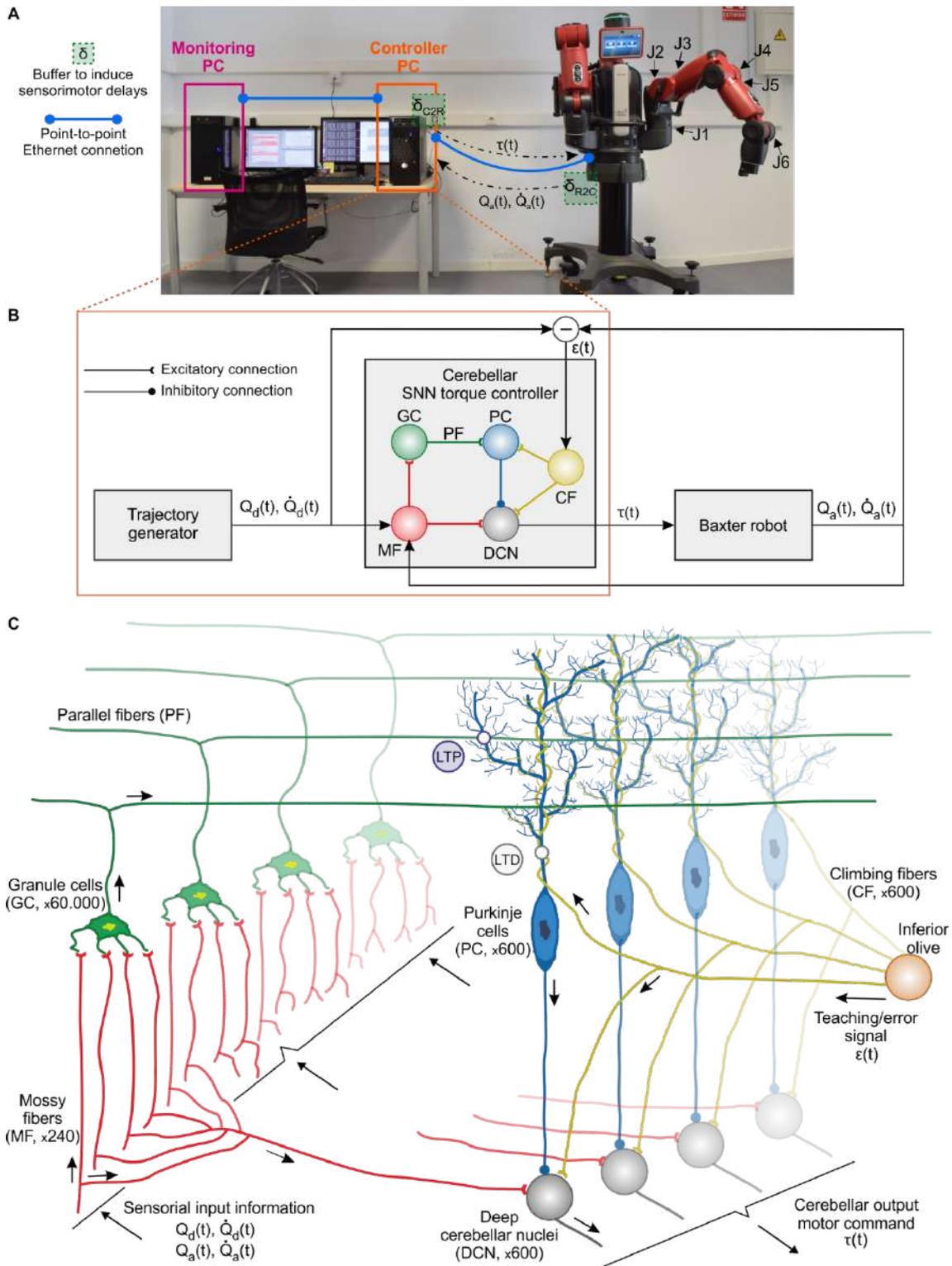
### 3.2.1 The cerebellar control loop with induced transmission delays

To validate the robustness against time delays of our cerebellar SNN torque controller, we placed it at the core of a robotic feedback control loop (Fig. 3.1), using Baxter robot [70] as the front-end body to be controlled. The SNN served as the torque controller to operate six DOF 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 sensory information and generated the subsequent output motor commands at 500 Hz; see *chapter 2* of this dissertation and [149] for an in-depth review of the learning process. The input sensory 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 ( $\varepsilon$ ) 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,  $\tau$ , per joint) and sent to the robot.

After SNN learning stabilization and thereby achievement of the desired trajectory, we induced different transmission delays ( $\delta\tau$ ) in the sensorimotor pathway to test whether our SNN inherits the cerebellar natural ability to deal with nondeterministic time delays [109], [49]. 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. The following scheme was used to artificially induce both R2C and C2R transmission delays: 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 (Fig. 3.1 A).

Besides the artificially induced transmission delays, the cerebellar SNN torque controller was also tested under real use-case scenarios with inherent transmission delays: i) wireless communications, i.e., Wi-Fi connection between robot and controller; ii) cloud-robotics, i.e., remote, long-distance ( $\sim 400$  km) connection over the Internet between robot and controller. See section 3.3.4 for further details.

### 3. Motor control under nondeterministic time delays



**Figure 3.1. 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, while 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. 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 an STDP mechanism at PF-PC connections.

### 3.2.2 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 neural population layers (see Fig. 3.1, B and C), each 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 [150]; 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 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 3.1, and the overall depiction of the cerebellar neural network is shown in Fig. 3.1 C.

LIF neurons [62] were used to build the cerebellar neural network due to their minimal computational cost, thus enabling our RT computation requirement. The neuron models used were the same as those described in 2.2.4 and Table 2.2. See *chapter 2* and [149] for an in-depth review on the cerebellar neural layers, their connectivity, and neuron models.

### 3. Motor control under nondeterministic time delays

**Table 3.1. Cerebellar neural network topology.** Dash entries indicate not applicable.

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	-

#### 3.2.3 The STDP mechanism - The “eligibility trace” and how it enters the learning rule equation

The STDP mechanism conjugating LTD and LTP at PF-PC synapses was the same as the one described in 2.2.5. However, the value of some parameters was modified to maximize the cerebellar tolerance to time delays and to adapt to Baxter’s dynamics changes induced by prolonged use of the robot. Hence, the LTD process was still described by:

$$LTD \Delta w_{PF_j-PC_i}(t) = \beta \cdot \int_{-\infty}^{t_{CFspike}} k(t - t_{CFspike}) \cdot \delta_{PFspike}(t) \cdot dt \quad (3.1)$$

$$k(x) = \begin{cases} \frac{-(x + d_k)}{\tau_{LTD} - d_k} \cdot e^{\frac{x + d_k}{\tau_{LTD} - d_k} + 1} & \text{if } x < -d_k \\ 0 & \text{if } x \geq -d_k \end{cases} \quad (3.2)$$

where  $\Delta w_{PF_i-PC_j}$  is the synaptic weight change between the  $j^{th}$  PF and the  $i^{th}$  PC,  $\beta = 0.0008$  nS is the synaptic weight decrement,  $\delta_{PF}$  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  $\tau_{LTD}$  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  $\tau_{LTD}$  ms before the CF spike arrival. For our SNN torque controller we established  $\tau_{LTD} = 150$  ms.

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

$$LTP \Delta w_{PF_j-PC_i}(t) = \alpha \cdot \delta_{PFspike}(t) \cdot dt \quad (3.3)$$

where  $\Delta w_{PF_i-PC_j}$  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  $\delta_{PF}$  is the Dirac delta function of an afferent spike from a PF.

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 STDP mechanism operating at this cerebellar layer combines a supervised LTD mechanism driven by the motor-error related signal and an unsupervised LTP mechanism that occurs even in the absence of such error signal [151].

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. 3.2, which only considers those PF spikes within the time-window before the CF spike discharge. The past activity of the afferent PF is evaluated similarly to a time-logged “eligibility trace,” [152], [153], [154]. 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 [99], [152], [155].

The amount of LTD produced is not constant (see the LTD kernel vs. time representation in Fig. 3.6), with a maximum depression 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 muscles 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 [156]. 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.

### 3. Motor control under nondeterministic time delays

#### 3.2.4 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. Slight changes were conducted on the spike to analog translation described in 2.2.7 to accommodate dynamics changes suffered by Baxter after prolonged operation and to account for the motor delays directly affecting transmission of torque commands.

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:

$$DCN_{j,i}(t) = \int_{t-t_{step}}^t \delta_{DCN_{j,i}}(t) \cdot dt \quad (3.4)$$

$$DCN_{output,j}(t) = \alpha_j \cdot \sum_{i=1}^{N=50} DCN_{j,i}(t) - \sum_{i=51}^{N=100} DCN_{j,i}(t) \quad (3.5)$$

where  $j \in [1, 6]$  for each of the six DOF;  $i \in [1, 100]$  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;  $\alpha_j = (0.75, 1.1, 0.375, 0.63, 0.078, 0.078)$  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:

$$\tau_j(t) = \frac{I}{2x+1} \cdot \left( \sum_{i=0}^x DCN_{output,j}(t+i \cdot t_{step}) + \sum_{i=1}^x DCN_{output,j}(t-i \cdot t_{step}) \right) \quad (3.6)$$

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 42 ms temporal window. This was in agreement with the upper motor neuron maximal discharge rates during slow isometric ramp contractions [157]. Predicted, past and current torque samples were placed within the mean filter based on their application time.

For analog to spike translation the same scheme as the one described in 2.2.6 was used, see [149] and [158] for further details.

### 3.2.5 Desired trajectories definition

We designed three motor tasks to be performed by the SNN torque controller under different 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 [111], [113], [112]. 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) [84], [111].

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

$$\left. \begin{aligned} x &= R \cdot \cos\left(2\pi \cdot \frac{t}{T} + \pi\right) \\ y &= R \cdot \sin\left(2\pi \cdot \frac{t}{T} + \pi\right) \\ z &= \alpha \end{aligned} \right\} t \in [0, 2]; \alpha = const \quad (3.7)$$

The inclined circle trajectory is described by:

$$\left. \begin{aligned} x &= R \cdot \cos\left(2\pi \cdot \frac{t}{T} + \pi\right) \cdot \cos\left(\frac{\pi}{6}\right) \\ y &= R \cdot \sin\left(2\pi \cdot \frac{t}{T} + \pi\right) \\ z &= R \cdot \cos\left(2\pi \cdot \frac{t}{T} + \pi\right) \cdot \sin\left(\frac{\pi}{6}\right) \end{aligned} \right\} t \in [0, 2] \quad (3.8)$$

The parametric equations of the Lissajous trajectory are:

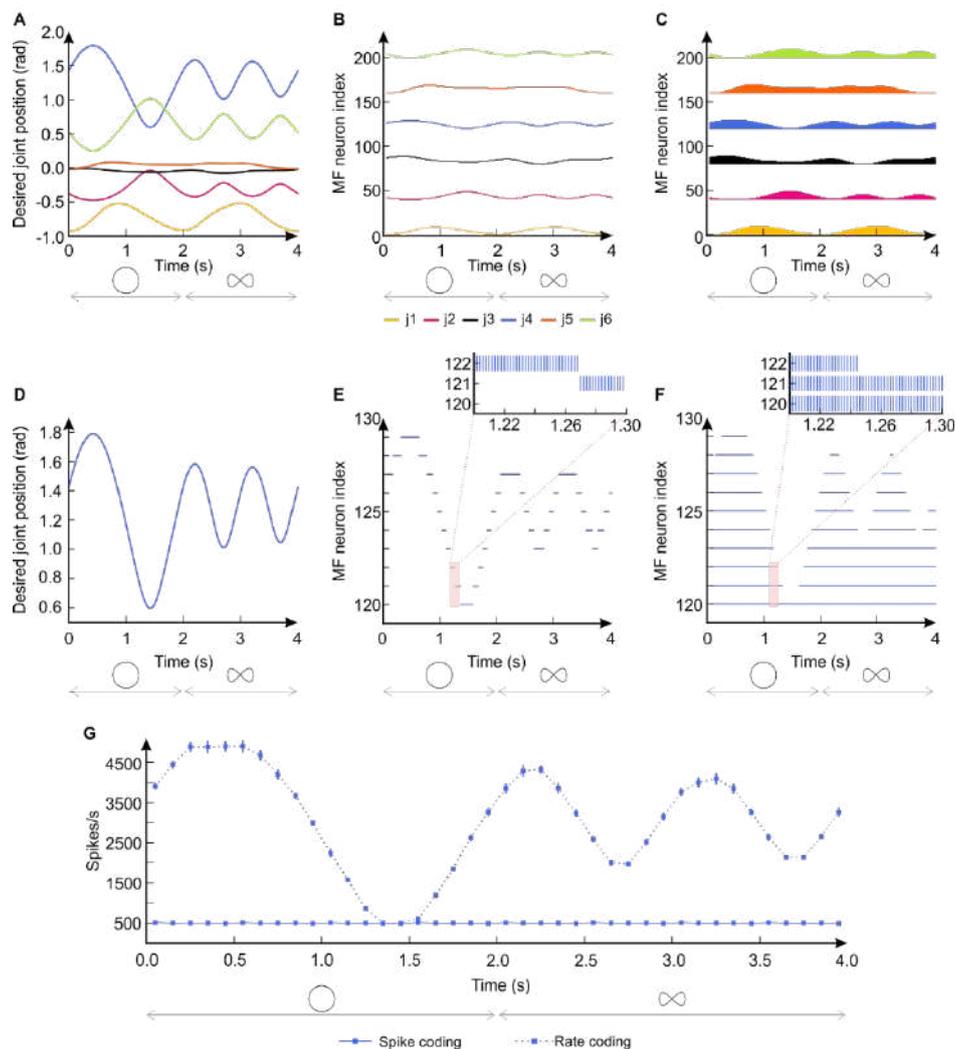
$$\left. \begin{aligned} x &= B \cdot \sin\left(b \cdot 2 \cdot \pi \cdot \frac{t}{T}\right) \\ y &= A \cdot \sin\left(a \cdot 2 \cdot \pi \cdot \frac{t}{T} + \delta\right) \\ z &= \alpha \end{aligned} \right\} \left. \begin{aligned} x &= \frac{R}{2} \cdot \sin\left(4 \cdot \pi \cdot \frac{t}{T}\right) \\ y &= R \cdot \cos\left(2 \cdot \pi \cdot \frac{t}{T}\right) \\ z &= \alpha \end{aligned} \right\} t \in [0, 2]; \alpha = const \quad (3.9)$$

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 [98], 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 ( $Q_d, \dot{Q}_d$ ).

### 3. Motor control under nondeterministic time delays

#### 3.2.6 The ANN cerebellar model

In conducting a more in-depth assessment of our SNN, we also developed a conceptually closer analog artificial neural network (ANN) controller. We used the analog cerebellar solution from [83], [84] 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, while PC long-term potentiation was related to PF activity. Thus, the cerebellar ANN helped to better contextualize the temporal nature of the SNN as key in dealing with time delays (see Fig. 3.2 for spike coding used by the cerebellar SNN, rather than rate coding).



**Figure 3.2. 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 (4 s: 2 s for circle + 2 s 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.

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 [159]. We implemented four main layers:

- Granular layer: implemented as a state-generator able to provide for different time stamps along the executed trajectory [88], [160] 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 PCs is defined by:

$$PC_i(t) = f_i(PF(t)), i \in \{1, 2, \dots, \text{number of motors}\} \quad (3.10)$$

where  $PC_i(t)$  represents the average firing rate of the PCs associated with the  $i^{\text{th}}$  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 MFs transmitting a baseline neural activity during the trajectory execution according to studies of eyeblink conditioning experiments [87], [161], [162].
- 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, 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. 3.11 describes the DCN layer behavior:

$$DCN_i(t) = MF_i(t) \cdot w_{MF-DCN,i} - PC_i(t) \cdot w_{PC-DCN,i} + CF_i(t) \cdot w_{CF-DCN,i}, \quad (3.11)$$

$$i \in \{1, 2, \dots, \text{Number of motors}\}$$

$DCN_i(t)$  represents the average firing rate of the DCN cells associated with the  $i^{\text{th}}$  motor,  $MF_i(t)$  stands for the baseline activity of the MFs associated to the  $i^{\text{th}}$  motor, and  $w_{MF-DCN,i}$  the synaptic strength of the MF-DCN connection to the  $i^{\text{th}}$  motor.  $w_{PC-DCN,i}$  represents the synaptic strength of the PC-DCN connection of the  $i^{\text{th}}$  motor. Finally,  $CF_i(t)$  represents the average firing rate of the CFs associated with the  $i^{\text{th}}$  motor, being  $w_{CF-DCN,i}$  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.

### 3. Motor control under nondeterministic time delays

$$\begin{aligned}\varepsilon(t) &= (q_{desired}(t) - q(t)) + (\dot{q}_{desired}(t) - \dot{q}(t)) \\ CF_i(t) &= \frac{\varepsilon_i(t)}{\varepsilon_{\max_i}} \in [0,1] \quad , i \in \{1, 2, \dots, \text{Number of motors}\}\end{aligned}\quad (3.12)$$

#### 3.2.7 Performance accuracy and learning convergence measurement

To evaluate the performance accuracy 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. See 2.2.9 for further details.

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 nondeterministic output, first we obtained the 100 iterations average torque per joint as follows:

$$\tau_j = \frac{1}{100} \sum_{i=1}^{100} \tau_i(t) \quad (3.13)$$

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:

$$\Delta\tau_j = \frac{t_{step}}{T} \sum_{t=t_{step}}^T \frac{\tau_j(t) - \tau_j(t - t_{step})}{t_{step}} \quad (3.14)$$

$$\Delta\tau = \frac{1}{N} \sum_{j=1}^N \Delta\tau_j \quad (3.15)$$

We also evaluated the performance of a proportional-derivative (PD) controller tuned to provide a performance accuracy similar to that of the factory-default position controller (see 3.3.1). 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.

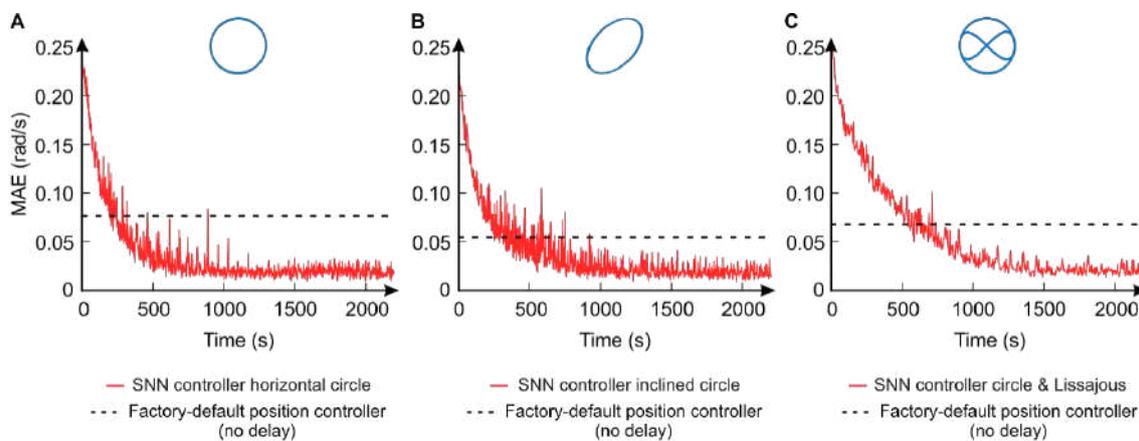
#### 3.2.8 Modules implementation

A 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](https://github.com/EduardoRosLab/EDLUT_BAXTER_DELAYS).

### 3.3 Results

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

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 3.2.5 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. Learning convergence is shown in Fig. 3.3.



**Figure 3.3. 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).

Once the learning process was stabilized, transmission delays were artificially induced between the two ends of the robotic feedback control loop in R2C and C2R directions (Fig. 3.1 A and section 3.2.1). A total transmission delay of  $\delta T = \delta_{R2C} + \delta_{C2R}$  was induced ( $\delta_{R2C} = \delta_{C2R} = \delta T/2$ ), which was progressively increased. 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 MAE illustrated the learning convergence of the SNN torque controller across a wide range of induced delays  $\delta T$  (Fig. 3.4 A). 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.

Since the factory-default position controller could not be tested in a time delay framework, we tuned a PD controller for each of the motor tasks using the Ziegler-Nichols method [163]. The resultant PD torque controller performance was similar to the factory-default position controller under no-delay circumstances (PD MAE = 0.076 rad vs. factory-default MAE = 0.077 rad for the horizontal circle trajectory, PD MAE =

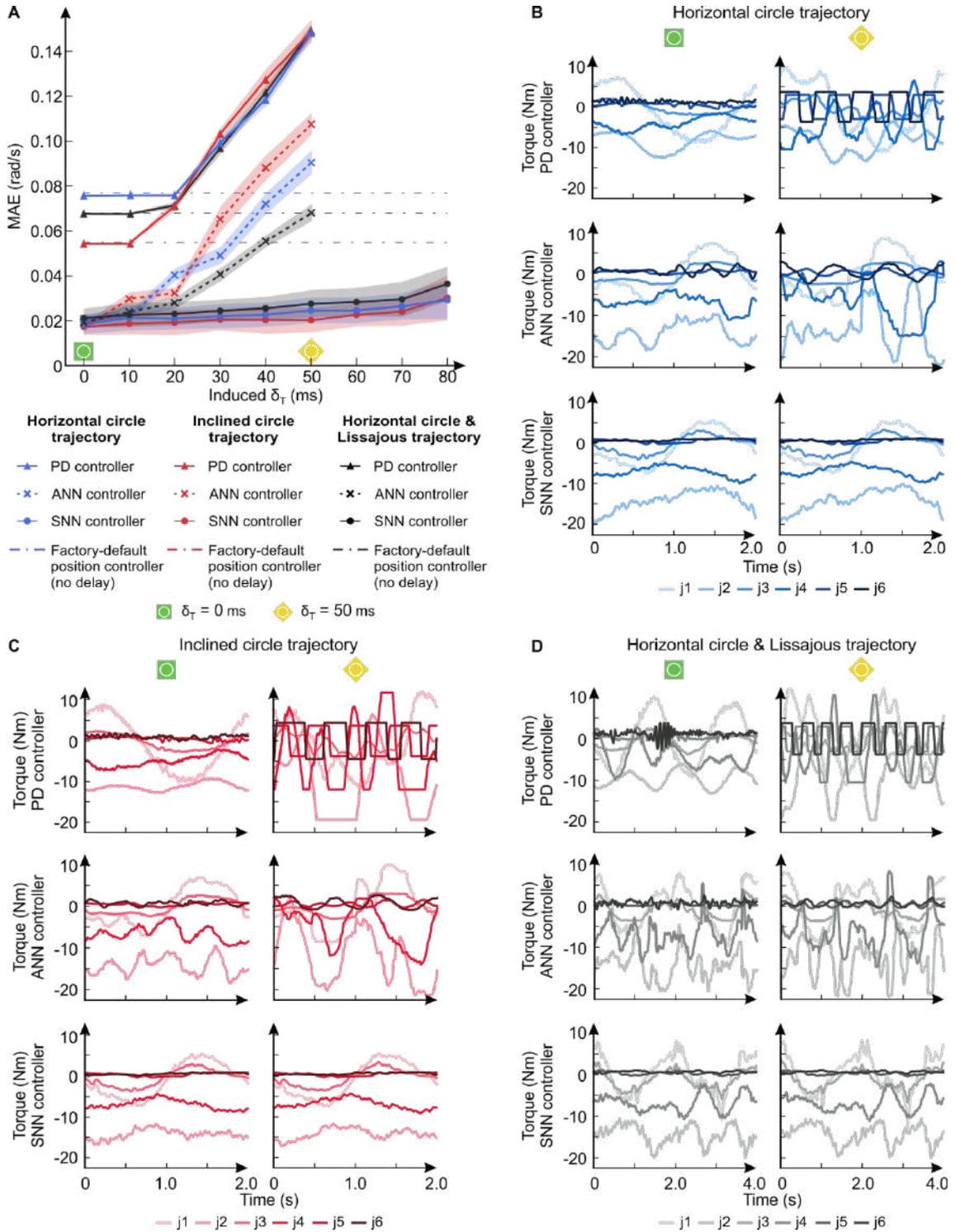
### 3. Motor control under nondeterministic time delays

0.054 rad vs. factory-default MAE = 0.055 rad for the inclined circle trajectory, and MAE = 0.068 rad for both the PD and factory-default controller for the circle-Lissajous sequence), thus, serving as a performance reference (Fig. 3.4 A). The cerebellar ANN performance was similar to the SNN cerebellar solution and better than the default-factory position controller under no-delay circumstances: ANN MAE =  $0.021 \pm 0.002$  rad vs. SNN MAE =  $0.018 \pm 0.004$  rad for the horizontal circle trajectory, ANN MAE =  $0.017 \pm 0.002$  rad vs. SNN MAE =  $0.017 \pm 0.004$  rad for the inclined circle trajectory and ANN MAE =  $0.019 \pm 0.001$  rad vs. SNN MAE =  $0.021 \pm 0.004$  rad for the circle-Lissajous sequence (Fig. 3.4A).

As the induced delay  $\delta T$  increased from 0 to 50 ms, the PD and ANN controllers performance degraded significantly (Fig. 3.4 A) 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. 3.4 B, C, and D). PD control instability 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 [136], 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 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 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  $\delta T$  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 =  $0.024 \pm 0.011$ ,  $0.022 \pm 0.008$  and  $0.027 \pm 0.007$  rad respectively (Fig. 3.4 A). For the PD and ANN controllers, 3 to 4 times larger MAE deviations were obtained: average MAE =  $0.099 \pm 0.027$  (PD controller) and  $0.053 \pm 0.026$  rad (ANN controller) for the horizontal circle trajectory,  $0.092 \pm 0.036$  (PD) and  $0.061 \pm 0.030$  rad (ANN) for the inclined circle trajectory,  $0.097 \pm 0.032$  (PD) and  $0.047 \pm 0.021$  rad (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. 3.4 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  $\delta_T$  was limited to 80 ms according to the predictive time margin of the deployed learning mechanism (see 3.2.3 and 3.3.2).

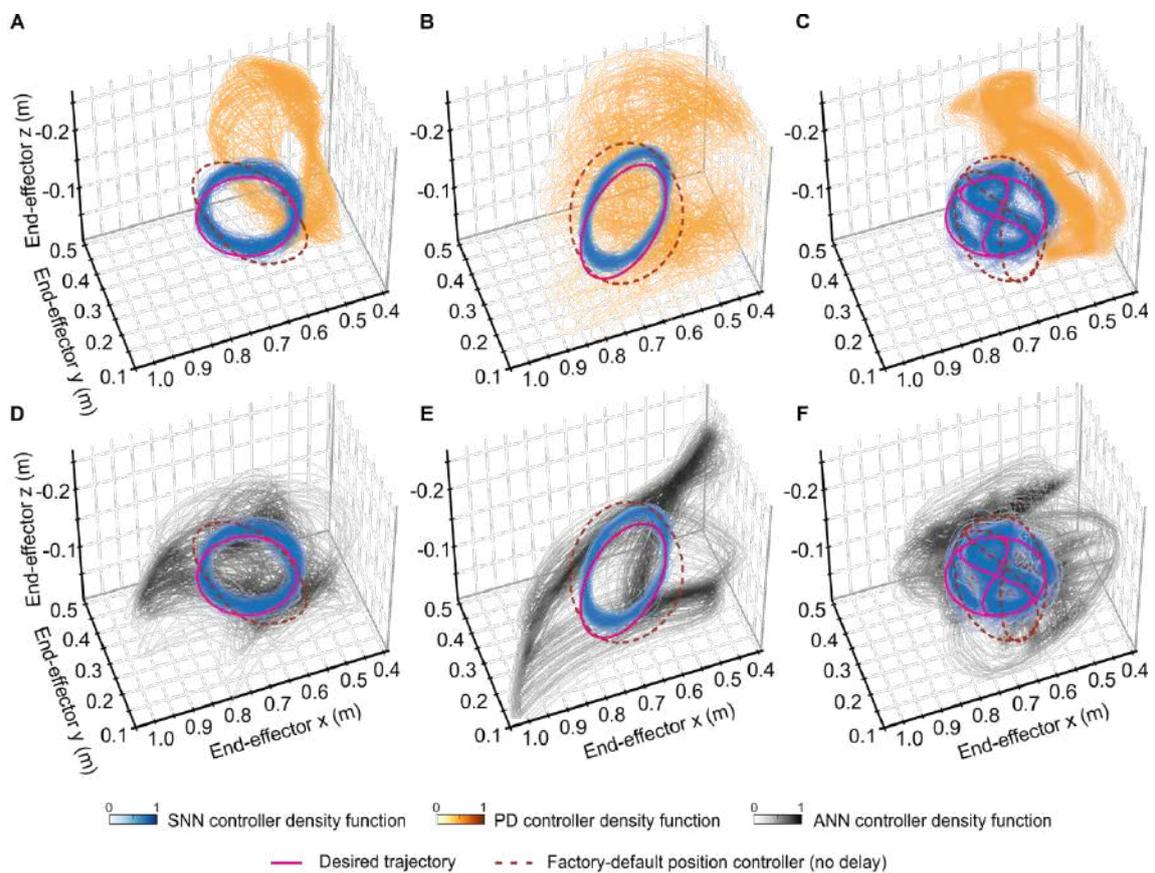


**Figure 3.4. PD and cerebellar ANN vs cerebellar SNN control response to steady time delays.** The induced transmission delays ( $\delta_T$ ) comprised symmetrical R2C and C2R steady time delays ( $\delta_T = \delta_{R2C} + \delta_{C2R}$ ;  $\delta_{R2C} = \delta_{C2R}$ ). (A) As  $\delta_T$  increased from 0 to 80 ms, mean MAE and standard deviation of 100 trials per  $\delta_T$  value performed by the Ziegler-Nichols tuned PD, the ANN and the SNN torque controller solutions. After tuning its parameters, the PD performed

### 3. Motor control under nondeterministic time delays

similar 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 [111], [112]. SNN controller MAE plateaued for values under  $\delta_T = 80$  ms, whereas both PD and ANN should not operate above  $\delta_T = 20$  ms (for safety reasons,  $\delta_T$  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  $\delta_T$  values from 0 to 50 ms (left and right column respectively).

Outstanding levels of accuracy were achieved by the SNN torque controller in the execution of the trajectories (Fig. 3.5). Comparative ANN vs SNN results indicated the time-related capability of the SNN form of synaptic plasticity accountable for coping with the delay.



**Figure 3.5. 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  $\delta_T = 50$  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.

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

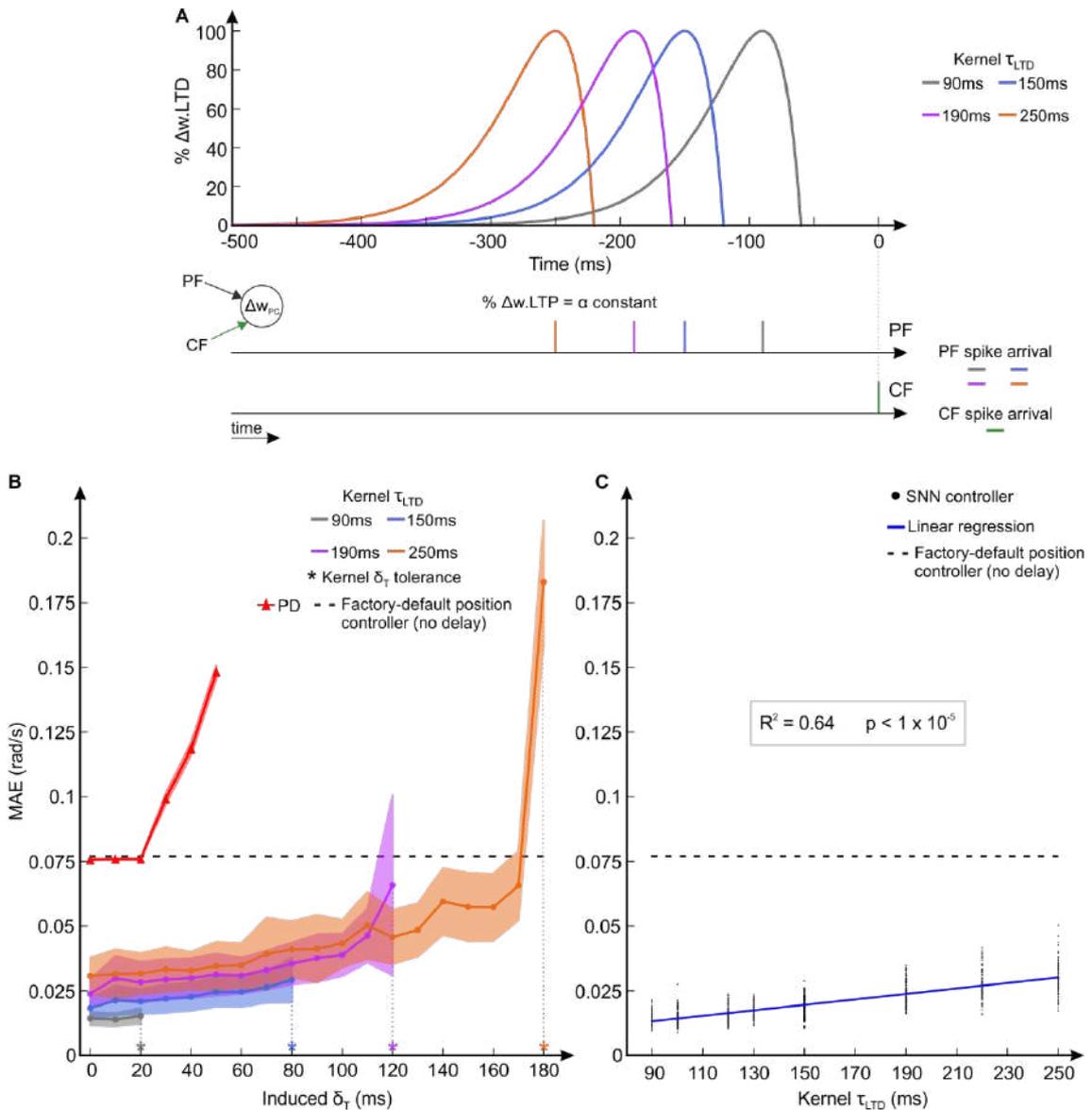
The presence of the biological sensorimotor delay causes a given sensorimotor state at time  $t$  to be received at the CNS at time  $t + \delta_a$  (afferent delay), and the subsequent motor command to be applied at time  $t + \delta_a + \delta_e$  (efferent delay). The tolerance of the biological learning 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 while 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 [99], [82]. 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” [99], [152], similar to a convolved coincidence detection able to compensate for transmission delays [164]. 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. 3.6 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. 3.6 B), but the performance error also increased (Fig. 3.6 C).

Electrophysiological recordings [108], [152] show an LTD contribution more acute for those PF spikes which 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,

### 3. Motor control under nondeterministic time delays

neural activity computation, analog to spike and spike to analog conversion, and torque commands application by the robot actuators. Please, see 3.2.3 for a more in depth description of the temporal kernel operation.

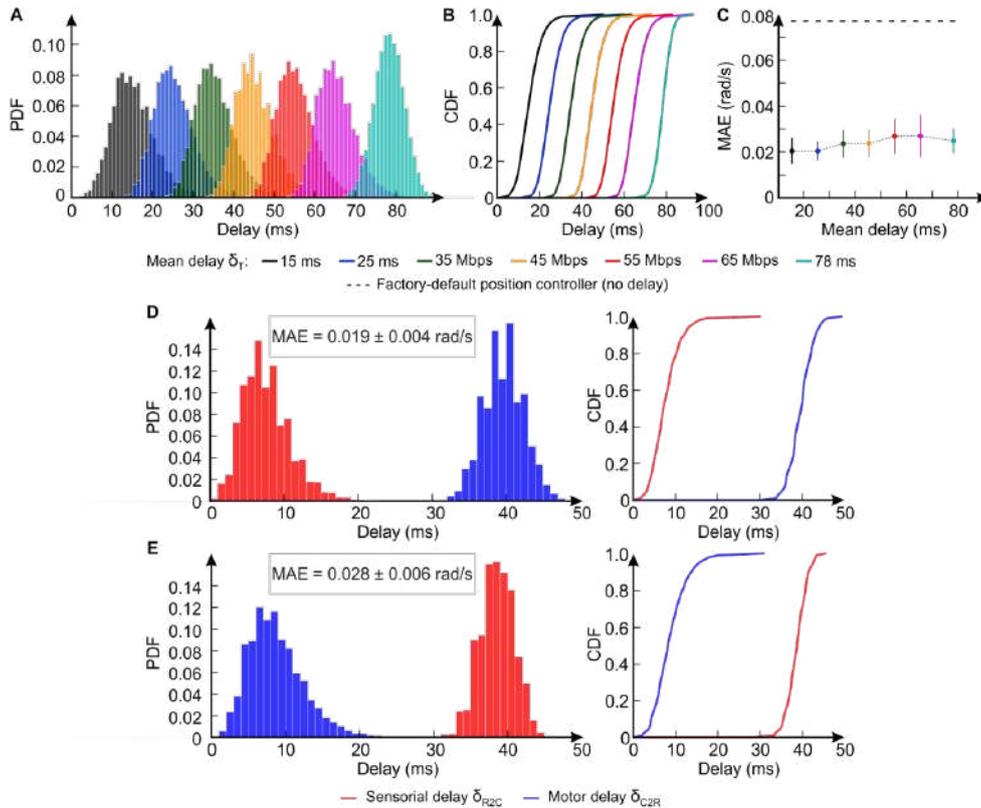


**Figure 3.6. Modifying the cerebellar predictive time margin by varying the STDP kernel.** (A) Set of CF-PF convolution kernels with different “eligibility trace” peaks ( $\tau_{LTD}$ ) [99] and how the CF spike arrival is correlated to previous PF spike for each convolution kernel. (B) Performance accuracy (MAE) obtained by the SNN controller for each of the convolution kernels ( $\tau_{LTD}$  peak varying from 90 to 250 ms), and PD controller reference. 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  $\tau_{LTD} = [50 - 150 \text{ ms}]$  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: a) “eligibility trace” peaks shorter than transmission delays, b) oversized “eligibility trace” peaks, i.e., beyond 300 ms.

### 3.3.3 Benchmarking the nondeterministic time delays

The learning convergence of our SNN output against steady time delays was tested so far; performance under nondeterministic time delays was still to be analyzed. Here, we characterized the response of our SNN to nondeterministic delays in a lab-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 ( $\delta_T = \delta_{R2C} + \delta_{C2R}$ ;  $\delta_{R2C} = \delta_{C2R} = \delta_T/2$ ), providing the following mean delays:  $15 \pm 5$  ms,  $25 \pm 5$  ms,  $35 \pm 5$  ms,  $45 \pm 5$  ms,  $55 \pm 5$  ms,  $65 \pm 5$  ms, and  $78 \pm 4$  ms (see Fig. 3.7 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. 3.1 A. 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. 3.7 C). Note that gamma distributions are proven to adequately model network delays [165], [166].

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)  $\delta_{R2C} = 8 \pm 3$  ms and  $\delta_{C2R} = 40 \pm 3$  ms (Fig. 3.7 D), ii)  $\delta_{R2C} = 39 \pm 2$  ms and  $\delta_{C2R} = 9 \pm 4$  ms (Fig. 3.7 E). We found that the SNN was able to cope with both symmetric and asymmetric nondeterministic delays.



**Figure 3.7. Symmetric and asymmetrical 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, (B) corresponding  $\delta_T$  CDF, and (C) SNN MAE performance. (D) and (E) asymmetrical nondeterministic delays scenarios. The depicted data accounts for 100 trials of the horizontal circle trajectory per delay distribution.

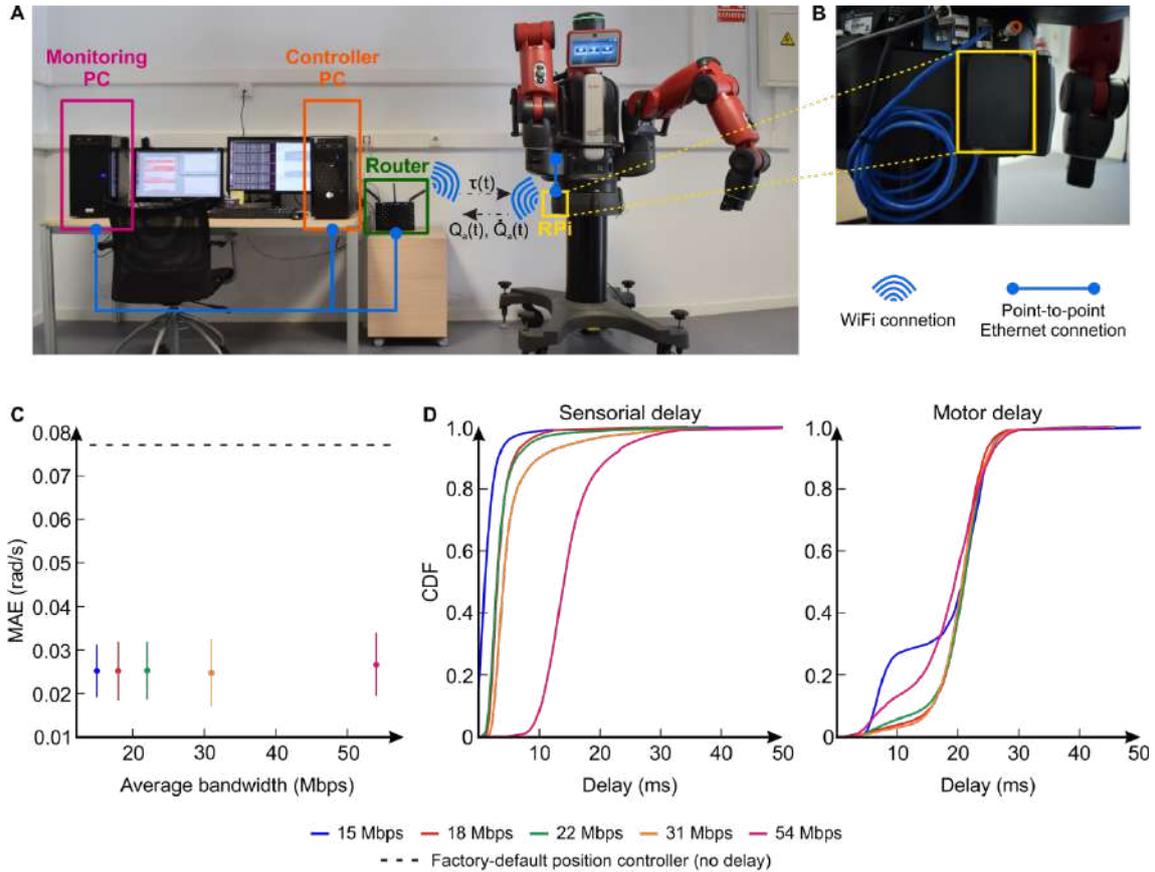
### 3. Motor control under nondeterministic time delays

#### 3.3.4 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. 3.8 A and B) to circumvent Baxter's lack of wireless support. We attached an RPI to the robot using an Ethernet connection with negligible delay ( $\delta_{Eth} \sim 0$  ms). The RPI, in turn, connected with the controller via a Wi-Fi connection which carried inherent nondeterministic time delays ( $\delta_{Wi-Fi}$ ). Thus, the RPI operated as a robot-controller gateway establishing an end-to-end Wi-Fi communication with nondeterministic time delays ( $\delta = \delta_{Eth} + \delta_{Wi-Fi} \sim \delta_{Wi-Fi}$ ) (Fig. 3.8 A and B). The Wi-Fi connection was established using a Tenda® AC15 AC1900 Smart Dual-band Gigabit Wi-Fi Router.

The nondeterministic delays inherent to a Wi-Fi connection [167] 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. 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 induced additional UDP traffic to the control loop end-to-end communication using the tool *Iperf* [168]. 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 nondeterministic time delays. The cost of the RPI acting as a bottleneck could be prevented if access to Baxter's onboard PC were granted or other more powerful nodes were used instead of the RPI.

We found that the SNN torque controller performance accuracy was kept at the same level regardless of the asymmetrical and nondeterministic time delays (Fig. 3.8 C); i.e., from bandwidth consumption of 15 to 54 Mbps, we obtained an average MAE of  $0.025 \pm 0.007$  rad, comparable to the  $0.024 \pm 0.011$  rad obtained at the artificial delays scenario with  $\delta_T$  from 0 to 80 ms. 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. 3.8 D), which added to the associated sensorial delay would set the PD and ANN controller in the instability zone (Fig. 3.4 A), risking robot and personnel safety.

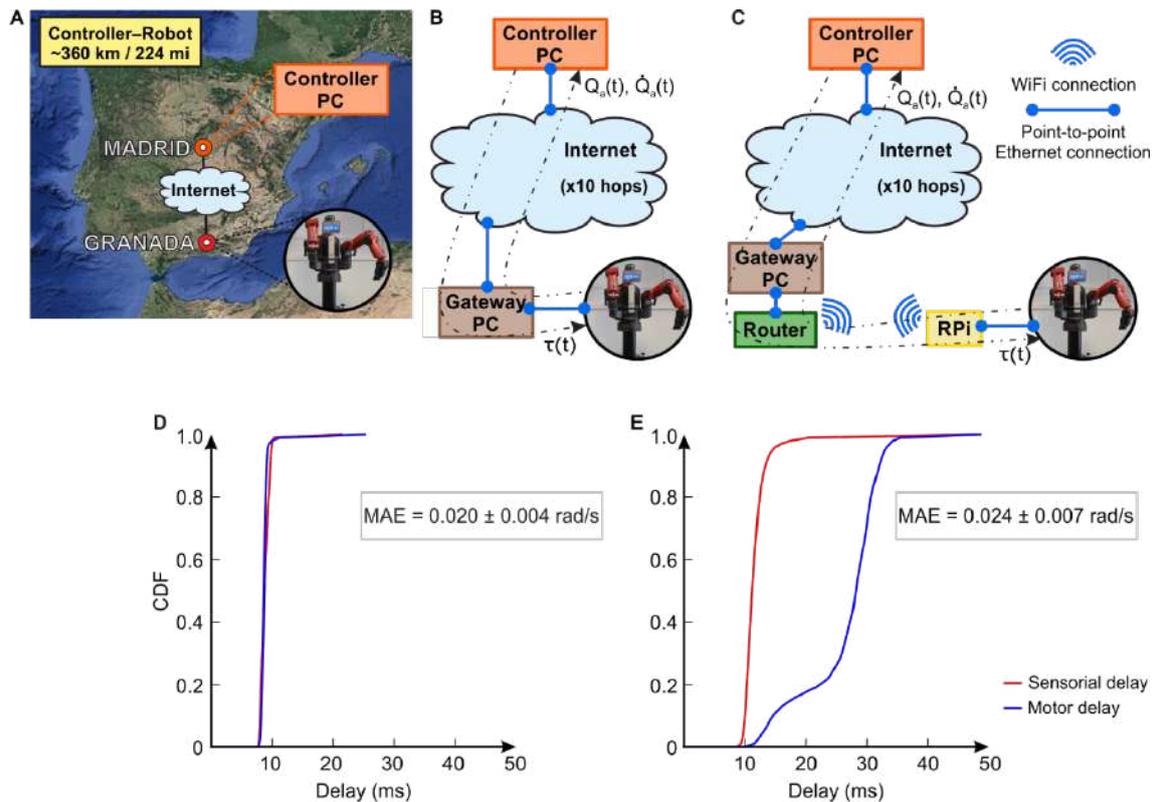


**Figure 3.8. Cerebellar response to nondeterministic Wi-Fi delays.** (A) Experimental setup in which the computer allocating the cerebellar controller and the robot communicated through Wi-Fi. The Controller and Monitoring PCs were connected to a router, which established a Wi-Fi connection with the (B) RPi attached to the robot. (C) Performance accuracy, 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 nondeterministic time delays, the SNN torque controller provided for compliance and accuracy.

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. 3.9 A). Two scenarios were tested: i) the robot connected to the Internet through an Ethernet connection via a gateway computer (Fig. 3.9 B); ii) the robot connected to the Internet via Wi-Fi (Fig. 3.9 C). In the first scenario, the sensorimotor time delay accounted for cloud-robotics inherent latency [169], [170]. The CDF of the sensorimotor time delays (Fig. 3.9 D) confirmed the 50<sup>th</sup>, 90<sup>th</sup>, and 99<sup>th</sup> 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

### 3. Motor control under nondeterministic time delays

predictive time margin (Fig. 3.6 B). The round-trip time (RTT) of the remote connection barely varied throughout the day (i.e., average RTT of  $20.0 \pm 1.3$  ms, from 8:00 to 24:00). In the second scenario, the connection was additionally hampered by the Wi-Fi nondeterministic time delays. The CDF confirmed the 50<sup>th</sup>, 90<sup>th</sup>, and 99<sup>th</sup> percentiles below 12, 14, and 20 ms for the sensorial messages; and below 29, 32, and 36 ms for the motor messages (Fig. 3.9 E); values below the 80 ms limit (Fig. 3.6 B). The accuracy obtained in both cases (1st and 2nd scenario MAE =  $0.020 \pm 0.004$  and  $0.024 \pm 0.007$  rad) 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 3.9. 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.

### 3.4 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 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 delays, nerve conduction delays, synaptic delays, neuromuscular junction delays, electromechanical delays, and force generation delays [47]. 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 [171], [172], [173]. 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 [49], [50], [74], [174] due to its ability to acquire internal models of the human body and external tools through motor learning [51], [107], [175], [176].

Consequently, cerebellum-inspired solutions have been proposed to different control problems: gaze stabilization [29], [127], [177], adaptive control of linear [130], [178] and nonlinear [119], [149], [179] systems, acquisition of forward/inverse [79], [180] dynamic models, or computation of inverse kinematics [181]. 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 [180]. 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 [182]. 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+50 ms), and tuned to exceed that delay by 20 ms (WH+70 ms) [164]. 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 [183]. 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.

### 3. Motor control under nondeterministic time delays

Cellular-level cerebellar controllers offer an insight into cerebellar function at neuron level. Yet, the significant computational cost of these models [184] 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, see *chapter 2* and [149]. 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 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 [55], 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 [185]. 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 [186]. 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 [187]. 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 [169]. Nonetheless, cloud-robotics faces the technical challenge of dealing with communication latencies [169], [170] between the cloud and edge nodes. Motor control can be highly sensitive to time delays as they drive the system towards instability and unmanageability [136], 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 [188], modifying the communications paradigm [189] and protocols [190], or implementing new communication technologies [191]. 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 [149] even in the presence of sensorimotor delays.

In this chapter, 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) [192] with potential use in cloud-robotics and remote control with long latencies.

### 3.4.1 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 [193], 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 [194]. 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).

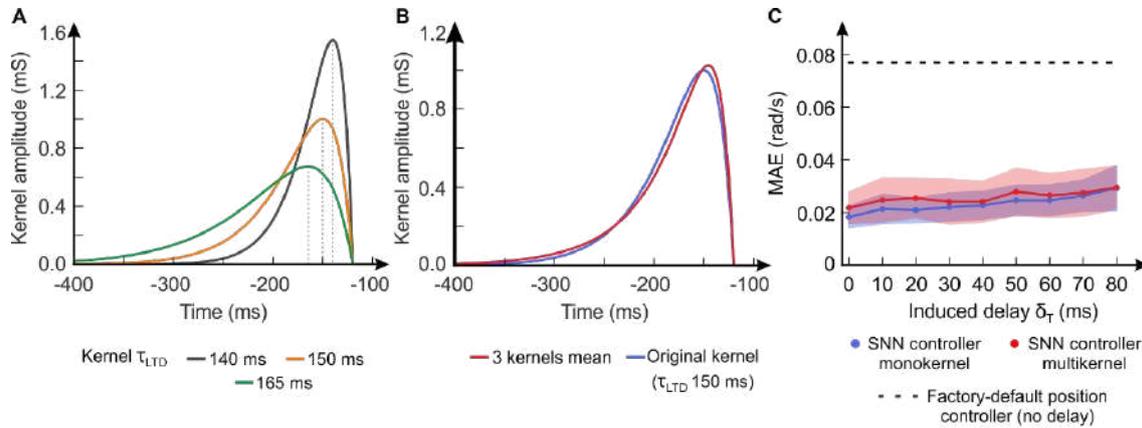
LTD is induced in the flocculus when PFs activate 120 ms before the CFs, assuming a PF-PC LTD monokernel presumably tuned to a unique pathway delay [194]. This plasticity at PF-to-PC synapses differs from plasticity found at PF-to-PC synapses in the vermis, in which LTD is induced by a range of PF-CF pairing interval (50 to 150 ms), assuming PF-PC LTD multikernels presumably tuned to a set of pathway delays [194]. 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 [195] to cognitive signals with, a priori, longer latencies [196], [197], [198]. 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 [194].

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

### 3. Motor control under nondeterministic time delays

with RT capacity being granted despite the multikernel approach higher computational cost (see Fig. 3.10), a multikernel solution adapted to a multipaired cerebellar architecture [77] could be a good approach to avoid the trade-off between accuracy and delay tolerance encountered in the monokernel solution.



**Figure 3.10. 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. 3.1 A, with induced delays  $\delta_T$  from 0 to 80 ms, using a unique robotic sensorimotor pathway delay (150 ms). 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 [29]. 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 is found at the cerebellar vermis [194]. The larger number of the multikernel PC outputs coalesced into the same number of DCNs for the monokernel network [199]. 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.

# 4. THE CENTRAL NERVOUS SYSTEM: A DISTRIBUTED NEURAL CONTROLLER – BEYOND CEREBELLAR CONTROL

## 4.1 Introduction

Motor behavior in mammals is governed by the hierarchical interaction of different neural areas of the CNS; a complex process in which different areas participate in several functions, making it hard to isolate them from one another [200]. Nonetheless, the implication of some areas of the CNS in this complex interaction can be highlighted: the motor cortex plays a key role in the volitional control of motor output, i.e., initiation and suppression of movements [201]; the basal ganglia are involved in selecting the most appropriate motor behavior [202]; the brainstem regulates autonomic functions such as cardiac and respiratory processes [203], [204]; the spinal cord receives descending control signals from higher brain areas, such as the cerebellum, and excites and regulates the activity of motor neurons, which ultimately drive muscle activation inducing movement [34]. There is a large body of research trying to understand the neural operation behind fine motor behavior, from movement planning to execution. As more light is shed on the different CNS areas involved in motor processes, the impact on health care and rehabilitation therapies will be a natural effect [205]. But, as already discussed, also robotics can benefit from our understanding of musculoskeletal motor control.

From an engineering perspective, the diversity of areas of the CNS involved in motor control can be understood as a distributed controller. Rather than just one controller receiving all the sensory inputs and generating all motor outputs, there is a variety of control centers using different kinds of input stimuli, exchanging signals among them and complementing their functions. The previous chapters of this dissertation have presented the benefits and possible application domains of a cerebellar-like robot controller: compliant cobot control suitable for physical HRI and robust against sensorimotor delays. The cerebellar SNN was here presented as a stand-alone controller, but the cerebellum cannot be fully understood without its interaction with the rest of the CNS.

#### 4. The central nervous system: a distributed neural controller – Beyond cerebellar control

Input and output cerebellar information is carried by the cerebellar peduncles, three paired bundles of white matter concentrating fibers that carry afferent and efferent connections to and from the cerebellum [206], [207]. The inferior peduncle carries mainly afferent fibers from the inferior olivary nucleus, and the spinocerebellar and vestibular systems. The middle peduncle, the largest of the three, contains axons from the pontine nuclei, thus connecting the cerebellum with the pons, which receives inputs from the cerebral cortex. The superior peduncle consists of efferent fibers connecting the cerebellar nuclei to the red nucleus, which projects to the spinal cord, and the thalamus, which connects back to the cerebral cortex. Hence, the cerebellum, although it plays a key role in motor control, is just one of the interconnected pieces involved in a well-orchestrated process.

In the present chapter, we go beyond cerebellar control and apply to robotics other key elements of the CNS and musculoskeletal system; we add spinal cord (SC) circuits and muscle dynamics to the control loop. Some preliminary results are here presented.

### 4.2 Spinal cord and cerebellum integration

The SC regulates motor behavior by transmitting control signals from higher brain areas to the muscles, and also by sending sensory signals generated at muscle receptors to those higher brain areas, which eventually modulate their motor output depending on the feedback. Besides, the SC also regulates fast reflex responses, allowed by direct muscle feedback; it can generate rhythmic movements such as locomotion; and it is also involved in motor learning [34], [208].

The role of the SC in regulating muscle activity is allowed by the sensory feedback coming directly from muscles. The primary afferent fibers (Ia) are the largest and most rapidly conducting peripheral nerves, and they originate at muscle spindles (stretch receptors sensitive to changes in muscle length and velocity) [209]. Ia afferents coming from a given muscle spindle connect to motor neurons of the same and synergistic muscles [34], i.e., muscles involved in a given movement (e.g., elbow flexion). The monosynaptic pathway (i.e., a direct connection involving just one synapse between the sensory and the motor neuron) received from Ia afferent fibers contributes to the stretch reflex: muscle stretching implies an increase in muscle length, encoded by the muscle spindle and transmitted through Ia afferent fibers, which activate the corresponding motor neuron, thus contracting the muscle and opposing the stretching. Another mechanism supported by Ia afferents is the reciprocal inhibition between antagonist motor neurons, i.e., those motor neurons involved in opposite movements (e.g., elbow flexion vs. extension). Reciprocal inhibition is carried through Ia inhibitory interneurons, which receive an excitatory input from the Ia afferent of a given motor neuron, and inhibit the corresponding antagonist motor neuron [210]. The connection between motor neurons and inhibitory Ia interneurons from the antagonistic muscle, serves to regulate reciprocal inhibition, which balances excitation-inhibition of agonist-antagonist muscles in flexion-extension movements [211]. Both the stretch reflex and reciprocal inhibition can be bypassed and regulated by connections from higher brain

areas, and they are just a glimpse of the processes intervened by SC circuits (e.g., recurrent inhibition supported by Renshaw cells inhibitory connections to homonymous and synergistic motor neurons; Ib afferents originated at Golgi tendon organs, sensitive to muscle contraction; group II afferents originated at muscle spindles and involved in posture and gait control) [34].

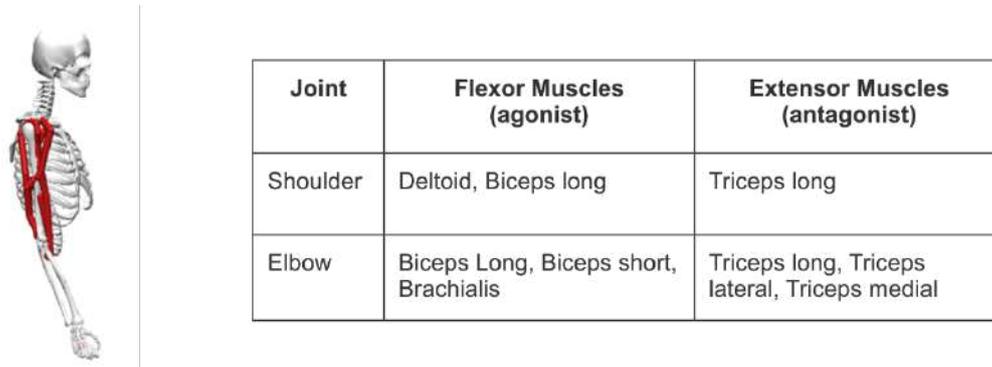
Here, we complement the cerebellar model presented in previous chapters with SC circuits. Cerebellar motor adaptation allowed by the STDP mechanism at PF-PC synapses, is now assisted by SC circuits including stretch reflex and reciprocal inhibition between antagonist muscles. We study whether cerebellar motor adaptation can be improved by the presence of SC circuits, how these two neural structures complement their roles in performing upper limb motor control, and the possible role of SC circuits in handling perturbations, i.e., external forces opposing the desired motion.

## 4.3 Methods

### 4.3.1 Front-end body: musculoskeletal upper limb

Muscles are at the core of SC operation, being the final actuators of motor behavior. Their activity is regulated by SC circuits which receive direct muscle feedback, thus allowing fast reflexes. Hence, prior to integrating the cerebellum-SC in a robot control loop, we used a simulated musculoskeletal upper limb as the front-end body to be controlled. The use of a musculoskeletal body allows direct muscle interaction and feedback, and also provides a more biologically relevant scenario to study the cerebellum and SC interaction.

To simulate the musculoskeletal upper limb we used the OpenSim simulator [212], a widely used tool among the biomechanics community. We simplified an existing arm model based on Hill-type muscles [213]. Our simulated arm model consisted of two DOFs (shoulder and elbow) actuated by seven Hill-type muscles (deltoid, triceps long, triceps lateral, triceps medial, biceps long, biceps short, and brachialis) (Fig 4.1).



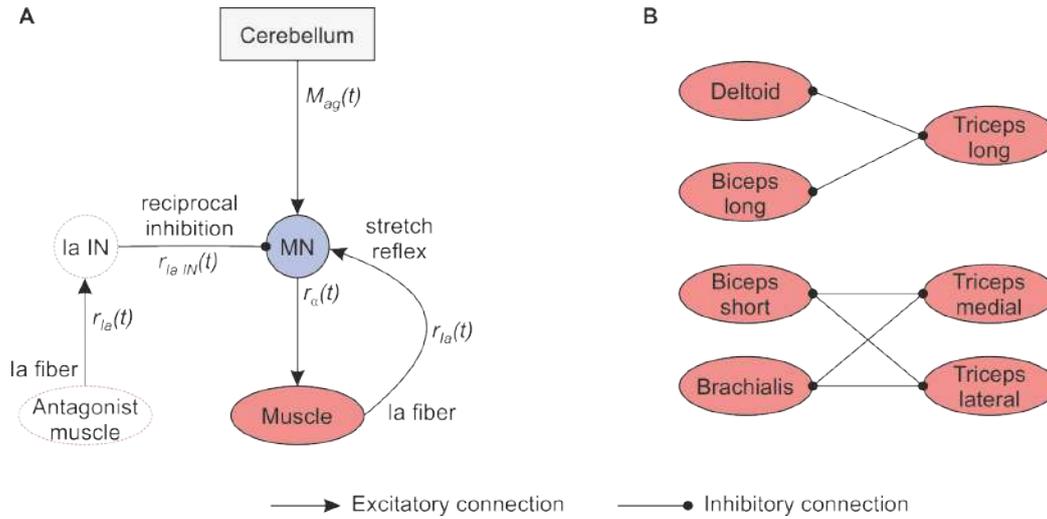
**Figure 4.1. Musculoskeletal upper limb model.** The model consists of two DOF (shoulder and elbow), actuated by seven muscles (deltoid, triceps long, triceps lateral, triceps medial, biceps long, biceps short, and brachialis). The table represents the muscles actuating each DOF, differentiating between flexor (agonist) and extensor muscles (antagonist).

#### 4. The central nervous system: a distributed neural controller – Beyond cerebellar control

The shoulder was actuated by two flexor muscles (deltoid and biceps long), and one extensor (triceps long). The elbow had three flexors (biceps long, biceps short, and brachialis), and three extensors (triceps long, triceps lateral, and triceps medial). Biceps long and triceps long were bi-articular muscles, i.e., they actuated both shoulder and elbow. The flexion-extension movement of the joints was regulated by the agonist-antagonist muscle interplay.

##### 4.3.2 Spinal cord circuits

We implemented a simple SC model that includes stretch reflex and reciprocal inhibition between antagonistic muscles. For that, each muscle was activated by one motor neuron (MN), which received the input control signal from the cerebellum ( $M_{ag}(t)$  or  $M_{ai}(t) \in [0, 1]$ , for agonist or antagonist muscles), an excitatory connection coming from the Ia afferent fiber and an inhibitory connection through an Ia interneuron (Ia IN). The Ia afferent fiber was originated at the same muscle activated by the MN, whereas the Ia IN was activated by the Ia afferent fiber originated at the corresponding antagonist muscle. See Fig. 4.2 for a schematic representation of the SC circuits and the agonist-antagonist relation between the seven muscles included in the musculoskeletal model.



**Figure 4.2. SC circuits and muscle pairs.** (A) Schematic representation of SC circuits. Each motor neuron (MN) received an excitatory input from the cerebellum (control signal), an excitatory connection from the Ia afferent fiber of the corresponding muscle (stretch reflex), and an inhibitory connection from the Ia interneuron (Ia IN) of the corresponding antagonist muscle (reciprocal inhibition). Muscle excitation given by MN firing activity,  $r_a(t)$ , conducted to muscle activation which induced movement. (B) Agonist-antagonist relation between muscles.

The dynamics of MN was described as a simple leaky neuron, with a firing rate ( $r_a$ ) given by:

$$\tau_m \frac{r_a(t)}{dt} = -r_a(t) + \sigma \left( \sum_i^N w_i \cdot r_i(t) \right) \quad (4.1)$$

$$\sigma(x) = \frac{1}{1 + e^{(-6(x-0.5))}} \quad (4.2)$$

where  $\tau_m$  stands for the membrane time constant, set to 10 ms; sigmoid  $\sigma$  emulates the on-off behavior of neurons;  $w_i$  is used to factor the weight of the input synapse, set to 1 for excitatory and 0.5 for inhibitory synapses; and  $r_i(t) \in [0, 1]$  stands for the  $N$  input signals, i.e.,  $M_{ag}(t)$  or  $M_{at}(t)$  control signals from the cerebellum;  $r_{Ia}(t)$  corresponding to stretch reflex, and  $r_{Ia\ IN}(t)$  corresponding to reciprocal inhibition through Ia IN. The dynamics of Ia IN neurons was also defined by Eq. 4.1 and 4.2, the only difference being the input signals (i.e., Ia IN neurons only input was from an excitatory Ia afferent fiber). The Ia afferent fibers firing rate is defined based on [214], as follows:

$$r_{Ia}(t) = 4.3v(t - \tau_s)^{0.6} + 2l(t - \tau_s) + 10 \quad (4.3)$$

where  $v(t)$  and  $l(t)$  are the muscle velocity and length, respectively;  $\tau_s$  is the sensory feedback delay set to 30 ms. The output is normalized to  $[0, 1]$ . Finally, muscle activation  $a(t)$  is given by the first-order activation dynamics related to muscle excitation (i.e., firing of MN,  $r_\alpha(t)$ ):

$$\frac{a(t)}{dt} = \frac{r_\alpha(t) - a(t)}{\tau(a(t), r_\alpha(t))} \quad (4.4)$$

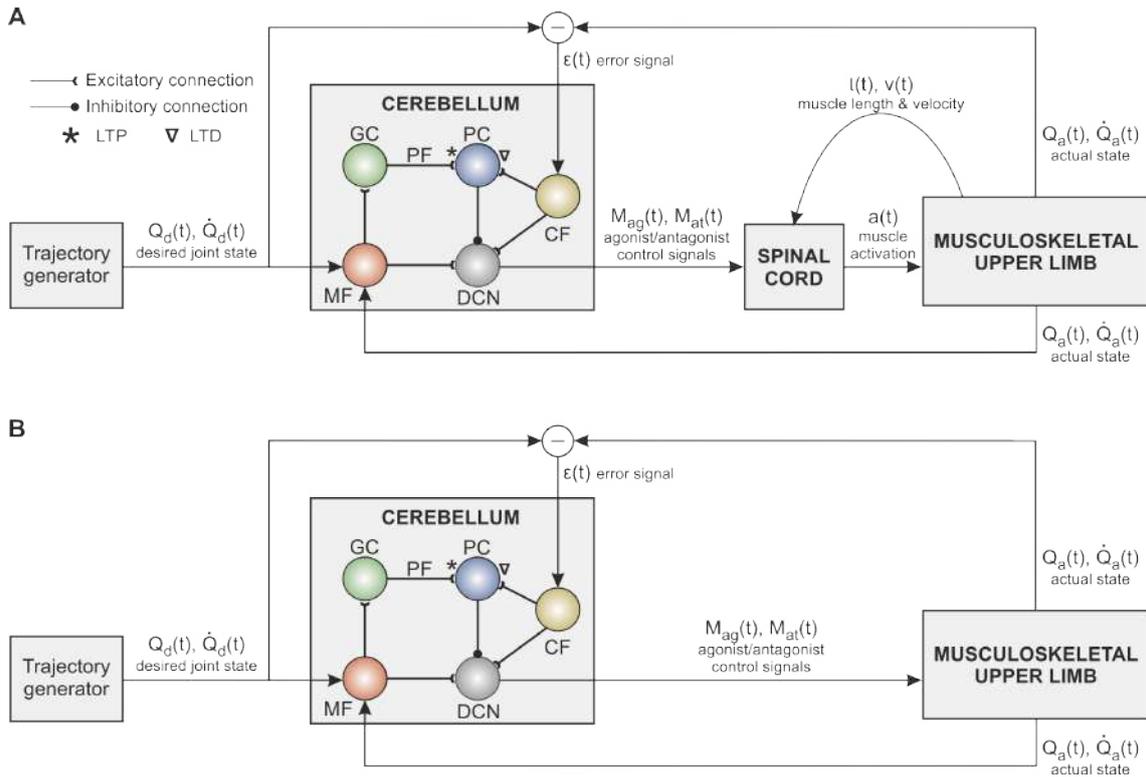
$$\tau(a(t), r_\alpha(t)) = \begin{cases} t_{act} (0.5 + 1.5a(t)) & ; r_\alpha(t) > a(t) \\ \frac{t_{deact}}{(0.5 + 1.5a(t))} & ; r_\alpha(t) \leq a(t) \end{cases} \quad (4.5)$$

where  $\tau(a(t), r_\alpha(t))$  expresses the excitation-activation variable time constant, with  $t_{act}$  and  $t_{deact}$  being the activation and deactivation time constants, assumed to be 10 and 40 ms, respectively [215], [216]. Muscle activation continuously varies between 0 and 1 (no activation – full activation), producing movement of the upper limb model.

### 4.3.3 Connecting the cerebellum and the spinal cord

The ROS framework presented in previous chapters was expanded with the new pieces. The SC circuits were placed between the cerebellum and the front-end body; the OpenSim upper limb model that substituted the robot. Both SC circuits and OpenSim model were loaded in two additional ROS nodes. Fig. 4.3 A depicts the cerebellum-SC control loop. To better contextualize the effects of adding SC to the control loop and to gain a better understanding of how the cerebellum and SC complement their roles, the control loop was also implemented lacking SC circuits (Fig. 4.3 B), i.e., direct cerebellar control of the musculoskeletal body. Hence, we could compare the performance of the control loop with and without SC.

#### 4. The central nervous system: a distributed neural controller – Beyond cerebellar control

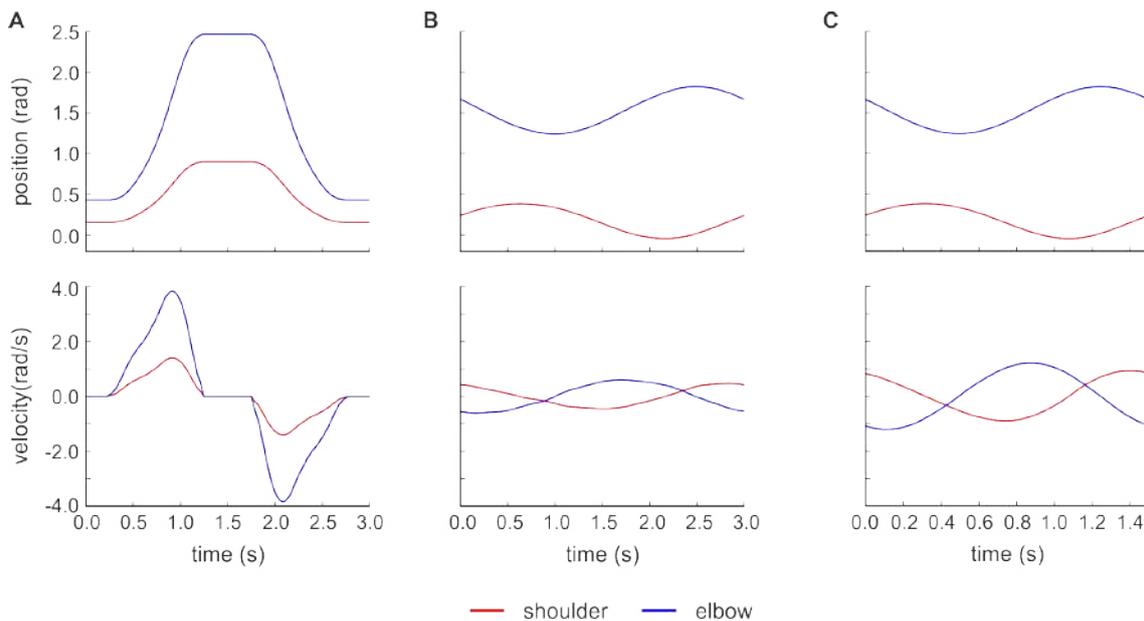


**Figure 4.3. Cerebellum-SC control loop. (A)** Schematic representation of the cerebellum-SC control loop of a musculoskeletal upper limb model. SC circuits, with the dynamics described in 4.3.2, generated the output activation signal  $a(t)$  for each muscle. Actual and desired joint position ( $Q_a, Q_d$ ) and velocity ( $\dot{Q}_a, \dot{Q}_d$ ) given in rad and rad/s, respectively, were used as cerebellar input; whereas SC circuits used muscle length ( $l$ ) and muscle velocity ( $v$ ) as input, given in mm and mm/s, respectively. **(B)** Cerebellar control loop of a musculoskeletal upper limb model.

In the cerebellum-SC scenario, the SC circuits received from the cerebellum two control signals per joint ( $M_{ag,j}(t)$  and  $M_{at,j}(t)$ ): one for the flexor (agonist) and one for the extensor muscles (antagonist). This configuration allowed maintaining the agonist-antagonist architecture of the cerebellar network discussed in previous chapters. In the case of bi-articular muscles (i.e., biceps long and triceps long), the control signal corresponded to the mean of the control signals received for both joints. The SC computed the muscle activation signals that were commanded to the upper limb model. In the scenario lacking the SC circuits, the agonist-antagonist control signals generated at the cerebellum were directly applied as muscle activation signals.

#### 4.3.4 Motor tasks

Two different motor tasks were used to test the cerebellum and SC integration: planar flexion-extension movement (3 s), and a circular trajectory executed in the vertical plane with different speeds (3 s, 1.5 s). These trajectories had the bell-shaped velocity profile found in experimental upper limb reaching movements. Fig. 4.4 depicts the position and velocity profiles of the benchmarking trajectories.



**Figure 4.4. Motor tasks.** (A) Flexion-extension movement, 3.0 s. (B) Slow circle trajectory, 3.0 s. (C) Fast circle trajectory, 1.5 s. The top row depicts the trajectory position for each joint; bottom row represents the trajectory velocity for each joint.

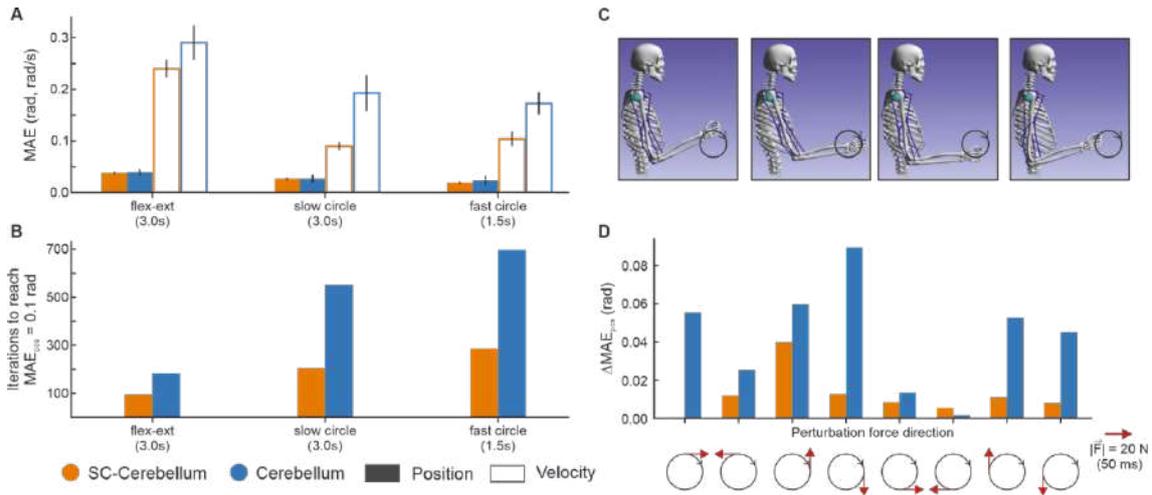
## 4.4 Results

We compared the performance with and without SC for the different trajectories, using the position and velocity MAE (i.e., mismatch between desired and actual trajectory, see 2.2.9 for further details) achieved after motor adaptation (Fig. 4.5 A). The results showed a similar final performance in terms of position accuracy, although the SC circuits provided a more stable performance (cerebellum-SC position MAE, mean  $\pm$  std =  $0.038 \pm 0.003$ ,  $0.027 \pm 0.002$ ,  $0.020 \pm 0.003$  rad; vs. cerebellum position MAE, mean  $\pm$  std =  $0.039 \pm 0.006$ ,  $0.027 \pm 0.007$ ,  $0.024 \pm 0.009$  rad; for the flexion-extension, slow and fast circle trajectories, respectively). In terms of velocity profiles the case with SC circuits provided a performance closer to the desired goal (cerebellum-SC velocity MAE, mean  $\pm$  std =  $0.24 \pm 0.017$ ,  $0.09 \pm 0.008$ ,  $0.104 \pm 0.014$  rad/s; vs. cerebellum velocity MAE, mean  $\pm$  std =  $0.29 \pm 0.034$ ,  $0.193 \pm 0.035$ ,  $0.173 \pm 0.022$  rad/s; for the flexion-extension, slow and fast circle trajectories, respectively).

Besides the performance provided after learning convergence, we also compared the learning convergence speed by measuring the number of iterations required to reach a position MAE of 0.1 rad (Fig. 4.5 B). Results showed that addition of the SC circuits helped to speed up cerebellar motor adaptation.

Finally, the SC role in handling motor perturbations was also tested. We induced motor perturbations in the form of external forces acting on the musculoskeletal model: 20 N during 50 ms applied in different directions while the model executed the circular trajectory (Fig. 4.5 C, D). Results showed that the control loop including SC circuits provided better response to motor perturbations, as the position MAE deviated less from the nominal behavior when the perturbation was applied.

#### 4. The central nervous system: a distributed neural controller – Beyond cerebellar control



**Figure 4.5. Performance with and without SC circuits (SC-Cerebellum vs Cerebellum controller).** (A) Performance accuracy given by the mean absolute error (MAE): difference between desired and actual joint position and velocity for each trajectory. (B) Motor adaptation convergence speed: number of iterations required to reach a position MAE of 0.1 rad. (C) Circular trajectory. (D) Position MAE deviation as force perturbations are applied to the hand in different directions ( $|F| = 20$  N during 50 ms) while performing the fast circle trajectory using both controllers, with and without SC circuits.

SC circuits were found to stabilize and fasten cerebellar motor learning and to handle perturbations through stretch reflexes. These features are of interest to robotics; hence, cerebellum-based cobot control could be further developed to include SC circuits, improving the benefits of cerebellar control discussed in previous chapters.

The results presented on cerebellum-SC integration were published at:

*Abadía, I., Bruel, A., Ijspeert, A., Ros, E., & Luque, N. R. (2022). Spino-cerebellar control of a musculoskeletal upper limb model, in 9<sup>th</sup> World Congress of Biomechanics, 2022.*

#### 4.5 Application to robotics – Adding muscles to cerebellar torque control

The benefits of adding SC circuits to cerebellar motor control have been presented in a musculoskeletal scenario. In order to apply those features to robotics we first need to accommodate in the robot control loop the sensory feedback required by SC circuits, i.e., muscle feedback. In the following sections we add muscle dynamics to Baxter robot. We discuss the benefits of adding muscle dynamics to the control loop used in previous chapters, beyond being a requirement for the future step of integrating SC circuits in robot control.

In robot control, one actuator controls the motion of one joint. In musculoskeletal control, joint actuation is rather more complex as several muscles control the joint

motion, with some muscles contributing to motion in one direction (flexor muscles) and others in the opposite direction (extensor muscles). Besides, muscles are not activated in an all-or-nothing fashion, i.e., both flexor and extensor muscles might be activated at the same time. Muscle synergies are pivotal for accurate motor control and motion stiffness, which is controlled by the modulation of the cocontraction of antagonistic muscles. The regulation of muscle cocontraction allows performing a given movement with different stiffness profiles, which can be modified depending on the motion requirements, e.g., in the presence of disturbances, cocontraction helps maintain position control [217].

In a physical HRI context, being able to control the robot motion stiffness appears as a promising approach. A low stiffness might be compared to an admittance controller, providing good performance in soft environments but contact instabilities and poor robustness in stiff environments [218]. Low stiffness motion would favor human intention during interactions between an operator and the robot. Conversely, a high stiffness motion might compare to an impedance controller, which provides good performance when the environment is stiff but poor accuracy when the environment is soft [218], thus the robot intention would prevail over possible disturbances. Switching between one scenario and the other (admittance vs. impedance) would be convenient in robotics depending on the requirements of the HRI use case put in place. Here we address this challenge by adding a muscle model to Baxter, and performing cerebellar control able to learn different stiffness profiles.

## 4.6 Methods

### 4.6.1 Muscle model

We added one simulated muscle to each joint of Baxter. The muscle model was derived from Ekeberg's antagonist muscle pair model [219], an approach applied to robotics before [28]. We added a cocontraction variable ( $c$ ) to the muscle model, being the torque generated by each muscle defined by Eq. 4.6:

$$\tau_j(t) = \alpha_j [M_{ag,j}(t) - M_{at,j}(t)] + \beta_j [(M_{ag,j}(t) + c) + (M_{at,j}(t) + c) + \gamma_j] [Q_{r,j} - Q_j(t)] + \delta [\dot{Q}_j(t)] \quad (4.6)$$

where  $\tau_j$  stands for the muscle output torque for joint  $j$ ;  $\alpha_j$  defines the muscle gain;  $M_{ag,j}(t)$  and  $M_{at,j}(t)$ ,  $\in [0, 1]$ , are the cerebellar agonist and antagonist control signals, respectively;  $\beta_j$  defines the muscle stiffness gain;  $\gamma_j$  is the muscle tonic stiffness;  $\delta_j$  is the muscle damping coefficient;  $Q_{r,j}$  defines the joint resting position; and  $Q_j(t)$  and  $\dot{Q}_j(t)$  are the joint position and velocity, respectively. We added the  $c$  component, common to all joints, which adds a cocontraction baseline to the cerebellar control signals. Changing  $c$  allows switching among different cocontraction profiles that modify the stiffness of motion. The values of the muscle parameters for each joint are defined in Table 4.1.

4. The central nervous system: a distributed neural controller – Beyond cerebellar control

**Table 4.1. Muscle model parameters.**

Joint	$\alpha$	$\beta$	$\gamma$	$\delta$
<i>J1</i>	30.0	1.0	3.0	-1.0
<i>J2</i>	40.0	1.0	3.0	-1.0
<i>J3</i>	30.0	1.0	3.0	-1.0
<i>J4</i>	30.0	1.0	3.0	-1.0
<i>J5</i>	10.0	1.0	3.0	-1.0
<i>J6</i>	10.0	1.0	3.0	-1.0

The Ekeberg muscle model uses a mathematical abstraction to represent antagonist muscles together, hence, for each Baxter joint we added one Ekeberg muscle that already accounts for both flexor and extensor muscles. The muscle model, which behaves as linear springs and dampers, includes an active torque component, the one directly controlled by  $M_{ag,j}(t)$  and  $M_{at,j}(t)$  control signals, and a passive torque element that depends on the joint position and velocity relative to the resting state. The resting position for joints J1 to J6 was  $Q_{r,j} = [0.0, 0.22, 0.0, 1.34, 0.0, 0.0]$  rad.

#### 4.6.2 Baxter cerebellar control loop with muscles

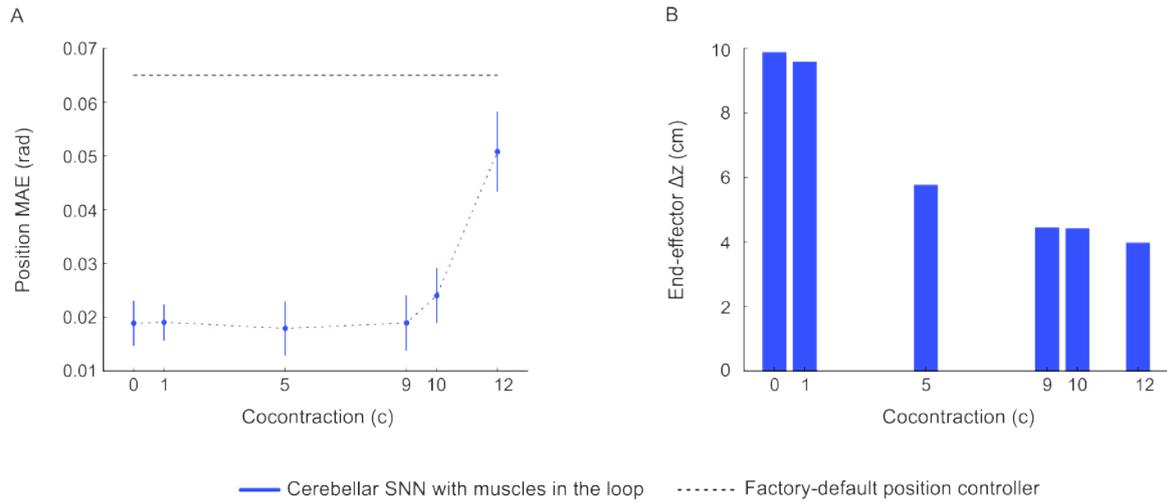
We maintained the ROS framework used in previous chapters with the addition of the muscle model between the cerebellum and the robot actuators. The agonist-antagonist architecture used in the cerebellar layers was maintained, allowing the generation of the  $M_{ag,j}(t)$  and  $M_{at,j}(t)$  control signals. For the spike-to-analog conversion, the equations used in 3.2.4 were used, but instead of subtracting the agonist and antagonist components (Eq. 3.5), they were kept as two separate output signals normalized to [0, 1], providing the final  $M_{ag,j}(t)$  and  $M_{at,j}(t)$  values commanded to muscles. The resulting torque from Eq. 4.6 was commanded to the robot.

## 4.7 Results

We tested the cerebellar control loop with simulated muscles and different cocontraction profiles performing the eight-like trajectory used in previous chapters (see 2.2.9). We established a baseline ( $c = x1$ ) cocontraction value of  $c_j = [1.5, 1.5, 1.5, 1.5, 0.5, 0.5]$  for joints J1 to J6. We then modified  $c_j$  and tested the following range of cocontraction profiles:  $c_j = x0$ ,  $c_j = x1$ ,  $c_j = x5$ ,  $c_j = x9$ ,  $c_j = x10$ ,  $c_j = x12$ .

We found that the cerebellar network performed accurate control for a wide range of cocontraction profiles (Fig. 4.6 A); for  $c_j = x10$  the MAE performance started to degrade, as the cerebellar output was close to saturation and the active torque component of muscles could not compensate for the passive component; and for  $c_j =$

$x12$  the performance was degraded significantly: position MAE =  $0.0189 \pm 0.0042$ ,  $0.0191 \pm 0.0034$ ,  $0.0179 \pm 0.0051$ ,  $0.0189 \pm 0.0052$ ,  $0.0240 \pm 0.0052$ ,  $0.0508 \pm 0.0075$  rad, for  $c = x0, x1, x5, x9, x10, x12$ , respectively. If higher cocontraction values were to be used, the muscle gain could be increased (note that for the different cocontraction profiles that were tested, the muscle parameters remained the same and only  $c$  was modified).



**Figure 4.6. Performance of the different cocontraction profiles. (A)** Performance accuracy, given by the position MAE, of the cerebellar SNN controller with muscles in the loop. The cocontraction  $c$  of the muscle model is modified. The cerebellar SNN learns to operate the robot for each  $c$  value, and then the position MAE is obtained for 100 trials of the eight-like trajectory. **(B)** Deviation of the end-effector in the vertical  $z$  plane as a 2.0 kg payload is attached. The distance from the resting position is measured for different  $c$  values, without any active component of the muscle being at play.

To test the response against perturbations we added a 2.0 kg payload to the end-effector in resting position, without any active component from the muscles, to see how the different cocontraction profiles handled the disturbance. We measured the deviation of the end-effector from its resting position in the vertical  $z$  axis. Results showed how the effect of the perturbation was reduced as higher cocontraction values were used:  $\Delta z = 9.9, 9.6, 5.8, 4.7, 4.4,$  and  $3.9$  cm, for  $c = x0, x1, x5, x9, x10, x12$ , respectively (Fig. 4.6 B).

## 4.8 Discussion and future work

The CNS can be an endless source of inspiration for the robotics community. In previous chapters we addressed the benefits of cerebellar-based control for cobots, which allows for adaptive, compliant control and robust to sensorimotor delays thanks to the deployed STDP mechanism. These benefits can be further developed by adding other CNS areas. Here we have discussed how SC circuits can complement the cerebellar network. SC circuits can help in handling perturbations thanks to fast-reflex responses using direct muscle feedback. Besides, direct muscle feedback allows the SC

#### 4. The central nervous system: a distributed neural controller – Beyond cerebellar control

to regulate muscle activation which has been shown to improve cerebellar motor adaptation.

The cerebellum-SC integration is currently under further development. We are conducting experimental studies with healthy subjects, who are asked to perform several upper limb movements involving 2 DOF (shoulder and elbow joints). We then extract kinematics and electromyography (EMG) data to be used in our simulated environment (i.e., from the lab to simulation). Our goal is to cross-validate our model using the EMG muscle profiles from the lab as reference against the performance of the musculoskeletal arm model in OpenSim being controlled by the integrated cerebellum and SC models. Validation of the model will help to further study the working hypothesis of the complementary roles of the cerebellum and SC circuits.

Integration of SC circuits in the Baxter control loop is currently also under further development. Here we have presented preliminary results on the addition of a muscle model to the robot. We are currently conducting a more in-depth analysis about the muscle model and cocontraction profiles, evaluating the possibility of the cerebellar network allowing switching between cocontraction profiles by making use of an input reward signal through the mossy fibers [220]. The different cocontraction profiles that were presented here were each controlled by a dedicated cerebellar network. If one single cerebellar network could learn several cocontraction profiles, then it could be used as one single controller able to modulate the degree of admittance/impedance control, a long-addressed challenge in the robotics community [218], [221]. Hardware efforts have succeeded in modifying the stiffness of motion [222], providing different compliance when interacting with the environment [223]; here we propose a biologically inspired, software approach.

Once the benefits of adding muscles to Baxter have been addressed, we plan to also include in the robot control loop the SC circuits studied using the OpenSim control loop. Hence, the cerebellar-SC integration could be applied to cobot control, further expanding the benefits of cerebellar control: adaptive, compliant cobot control robust to sensorimotor delays, now complemented by an increased robustness to perturbations and the ability to switch between different cocontraction profiles (admittance vs. impedance).

The work presented in this chapter and the current development have been made possible thanks to the participation in the European Project HBP (see 1.7), which has brought our research group together with the BioRobotics Laboratory, led by Prof. Auke Ijspeert, with long experience in SC circuits and musculoskeletal systems, and their application to robotics.

## 5. DISCUSSION

In this dissertation we have presented a cerebellar SNN robot controller. The cerebellar SNN, accounting with the main neural populations of the cerebellar cortex (MF, GC, PC, CF, and DCN) and the primary form of cerebellar synaptic plasticity (STDP), has been validated as an adaptive, compliant controller of a nonlinear robot, and also proven robust against sensorimotor time delays. Integrating into the cerebellar controller other parts of the CNS and musculoskeletal system (SC circuits and muscle dynamics) has also been addressed.

Physical HRI is driving robotics toward the use of elastic components, which provide passive compliance at the cost of increasing the dynamic complexity, i.e., elasticity brings a highly nonlinear component which hinders the mathematical modeling of the robot. Hence, new control approaches which do not rely on analytical solutions are required. This challenge can be addressed using different perspectives involving ANNs [17], [72]; here, we have proposed a cerebellar SNN controller; i.e., a biologically inspired approach as motor control of the musculoskeletal system inherently deals with nonlinearity induced by the viscoelastic dynamics of muscles, tendons, ligaments and tissues.

The cerebellum has long been proposed as an adaptive controller, with different modeling approaches being used (state-encoder-driven models, cellular-level models, functional models [35]). Cellular-level models can fully benefit from the features of cerebellar control and the findings provided by neuroscience studies. However, their high computational cost has usually relegated them to non-RT and simulated scenarios [99], [129], small-sized networks [29] and low resolution control signals [130]. In the work presented here, we addressed the challenge of applying a cellular-level cerebellar model to a real-world robot control loop. First, we overcame the challenge of operating in RT a large-enough cerebellar SNN (~62 k LIF neurons, ~36 M synapses). The size of the network had to be large-enough as to provide the required resolution to perform direct torque control of 6 DOF (Baxter robot), which brings the second challenge here addressed: accurate and adaptive torque control of 6 nonlinear DOF. The cerebellar network was proven able to provide fine motor control of Baxter, a robot with nonlinear dynamics due to the use of series elastic actuators and passive springs. Besides, the cerebellar SNN was tested dealing with motor perturbations and HRI, providing

## 5. Discussion

satisfactory results for both scenarios. Hence, the cerebellar SNN was validated as an adaptive, compliant controller (*chapter 2*).

Special mention must be made of EDLUT simulator [68], at the core of the work presented here. Some of the known features of biological cerebellar motor control (motor learning, adaptive and predictive control) could be successfully applied to a cobot control scenario thanks to EDLUT being specially designed to operate in RT allowing cellular-level models with enough resolution for input-output control signals. Efficient SNN simulators are a requirement to address the embodiment of biologically plausible neural systems, pivotal for neurorobotics development. EDLUT has been proven qualified for the task, without using any specific hardware (i.e., just using regular CPU and GPU computation).

Once the performance of the cerebellar SNN was validated in terms of accuracy and compliance, we tested its applicability in the presence of sensorimotor delays. The cerebellar predictive behavior allows dealing with the natural sensorimotor delays affecting the CNS. This robustness against communication latencies would be beneficial for robot control scenarios affected by such delays, a current challenge due to the increased use of wireless setups [20], cloud-robotics [21], remote connections or teleoperation architectures [22]. We introduced sensorimotor delays in the Baxter control loop using different approaches: constant and stochastic delays induced in a lab-controlled setup; and real-world use cases such as Wi-Fi communication and remote, long-distance connection over the Internet. The results proved the cerebellar SNN controller able to provide stable behavior despite the presence of different kinds of latency (*chapter 3*).

The symbiosis between neuroscience and robotics does not end with the cerebellum. Neurorobotics shall benefit from the understanding of the different CNS areas involved in motor control, from decision making, to motor planning and execution. In *chapter 4* we presented a first approach for the integration of the cerebellar controller and SC circuits including stretch reflex and reciprocal inhibition, which improved cerebellar learning and contributed to handling motor perturbations. The cerebellum-SC integration was studied in a simulated environment involving motor control of a musculoskeletal upper limb model; future work will focus on the application to robotics of the cerebellum-SC integrated model. Also in *chapter 4* we introduced muscle dynamics into Baxter, a line of research oriented to provide different stiffness profiles by modification of muscle cocontraction, while maintaining movement accuracy. Preliminary results presented here point toward a promising approach to allow for switching between admittance or impedance robot behavior. Selecting between admittance or impedance profiles will depend on the HRI requirements; i.e., admittance control favors performance in soft environments, while impedance control provides better results in stiff environments [218]. Future work will address the cerebellar capabilities for learning different stiffness profiles by using cerebellar reward signals [220]. Furthermore, to select which profile to use in a given case, a basal ganglia model could be added to the control loop [224], a CNS area involved in action selection processes. The basal ganglia model would provide a context-switching signal to

differentiate between admittance or impedance control; that signal would reach the cerebellar model through the mossy fibers, and the cerebellar network would learn how to adapt its behavior to provide for accurate motor commands for the different stiffness profiles. Cerebellar output motor commands would then be sent to SC circuits, which would induce muscle activation translated into output torque commands to be applied by the robot actuators. Therefore, future work will focus on further expanding the biologically inspired, *neurorobotics* approach here presented.

### 5.1.1 Summary of accomplished objectives

Within this dissertation, we have accomplished the following specific objectives:

- Development of the cerebellar SNN adapted for 6 DOF using EDLUT simulator (*chapter 2*).
- Integration of the cerebellar SNN in the control loop of a Baxter robot, in RT (*chapter 2*).
- Validation of the motor learning capability of the cerebellar SNN torque controller. The cerebellar SNN learnt torque commands to perform different trajectories involving 6 nonlinear DOF of one arm of the Baxter robot (*chapter 2*).
- Validation of the performance accuracy of the cerebellar SNN torque controller. The achieved motor learning provided fine accuracy in the execution of the motor tasks, outperforming the factory-default position controller (*chapters 2 and 3*).
- Validation of the adaptability of the cerebellar SNN torque controller to unstructured scenarios. The SNN was able to adapt to dynamic changes induced by context variations; i.e., additional 0.5 kg payload and external elastic-forces (*chapter 2*).
- Validation of the compliant behavior of the cerebellar SNN torque controller. The cerebellar SNN allowed safe HRI (*chapter 2*).
- Validation of the robustness of the cerebellar SNN torque controller against nondeterministic time delays. The SNN was able to control the cobot in scenarios affected by nondeterministic transmission delays between the robot and controller. The provided torque control was stable in the presence of both constant and stochastic time delays in lab-controlled and real-world use cases. The cerebellar tolerance against sensorimotor delays could be modified followed by variation of the performance accuracy; i.e., trade-off between robustness against time delays and performance accuracy (*chapter 3*).
- Integration of the cerebellar SNN and SC circuits (*chapter 4*).

## 5. Discussion

- Validation of the performance of the cerebellum-SC integration controlling a musculoskeletal upper limb model. The cerebellum-SC controller improved learning convergence and performance accuracy of the cerebellar controller (*chapter 4*).
- Validation of the contribution of SC circuits in handling motor perturbations. The cerebellum-SC controller was proven more robust against perturbations than the cerebellar controller (*chapter 4*).
- Integration of muscle dynamics in Baxter control loop (*chapter 4*).
- Validation of the performance of different muscle cocontraction profiles and their response against perturbations. The cerebellar controller successfully learnt different cocontraction profiles with similar performance accuracy. Higher cocontraction profiles were proven more robust against motor perturbations (*chapter 4*).

## 6. CONCLUSION

Robots are tools which have been around humans for decades; from their original application in factory automation, to more recent application domains such as health care or education. However, robot behavior is still far from the flexibility and adaptiveness provided by humans; the closer to human behavior, the more complex applications robotics will have. Thus, robotics development shall directly benefit from the study of human behavior itself, where many questions are still to be addressed.

In future years, neuroscience will expand the answers to decode and further understand human cognition, emotion, and motion. The application to robotics of such neuroscience findings will require bridging the gap between biology and technology, which can be done using computational neuroscience to artificially replicate the biological substrate. Artificial replicas of the CNS can be embedded in robots, which equipped with copies of the machinery behind human behavior would then be more suitable for application in complex contexts. But the duty of these robots does not end with physical jobs. The embodiment of biologically plausible neural networks allows these robots to repay the favor to neuroscience and be used as guinea pigs to study neurological function, disorders and possible treatments; closing the loop of neurorobotics. Robots can help us as mere hardware tools, but they can also play a key role in understanding our own nature and behavior.



# RESEARCH PUBLICATIONS

## Journal papers

- Abadía, I., Naveros, F., Garrido, J. A., Ros, E., & Luque, N. R. (2019). On robot compliance: a cerebellar control approach. *IEEE Transactions on Cybernetics*, 51(5), 2476-2489.

<b>IEEE Transactions on Cybernetics</b>			
Journal metrics 2019			
<b>Impact factor</b>	<b>Category</b>	<b>JIF Quartile</b>	<b>JIF Rank</b>
11.079	Automation & Control Systems	Q1	1/63
	Computer Science, Artificial Intelligence	Q1	5/137
	Computer Science, Cybernetics	Q1	1/22

*JIF = Journal Impact Factor*

- Abadía, I., Naveros, F., Ros, E., Carrillo, R. R., & Luque, N. R. (2021). A cerebellar-based solution to the nondeterministic time delay problem in robotic control. *Science Robotics*, 6(58), eabf2756.

<b>Science Robotics</b>			
Journal metrics 2021			
<b>Impact factor</b>	<b>Category</b>	<b>JIF Quartile</b>	<b>JIF Rank</b>
27.541	Robotics	Q1	1/30

*JIF = Journal Impact Factor*

## Conference presentations

- Abadía, I., Bruel, A., Ijspeert, A., Ros, E., & Luque, N. R. (2022). Spino-cerebellar control of a musculoskeletal upper limb model, in 9<sup>th</sup> World Congress of Biomechanics, 2022.



# RESUMEN

En esta tesis presentamos una solución biológicamente inspirada para un problema de control robótico: un modelo computacional que emula la estructura y funcionalidad del cerebelo, capaz de controlar por par de fuerzas, de manera adaptativa y segura, un brazo robótico de dinámica no lineal. El modelo de cerebelo se demuestra también robusto ante retardos de transmisión entre robot y controlador. Por último, ampliamos el enfoque bioinspirado añadiendo un modelo de médula espinal y dinámicas musculares al controlador basado en cerebelo.

## *Introducción*

En las últimas décadas, la robótica ha evolucionado hacia un nuevo paradigma caracterizado por la interacción física entre humanos y robots (HRI, *human-robot interaction*) [7]. Este nuevo escenario contrasta con los escenarios clásicos de robótica industrial, en los que se evita cualquier contacto entre el operario y el robot al no poder garantizar una interacción segura; la robótica ha ampliado sus aplicaciones hacia nuevos contextos que exigen que la interacción humano-robot sea segura (tareas de rescate, terapias de rehabilitación o asistentes educativos, entre otros). Para atender esta demanda, una nueva familia de robots ha entrado en juego: los robots colaborativos (cobots).

En el desarrollo de cobots, tanto el hardware como el software se han de tener en cuenta. En cuanto al hardware, los cobots cuentan con actuadores de baja potencia y componentes elásticos que ofrecen seguridad pasiva; esto es, en caso de impacto, el propio hardware del robot minimiza los posibles daños. Estos componentes elásticos y de baja potencia, que emulan en cierto modo las propiedades de músculos, tendones y ligamentos, contrastan con los robots rígidos y de alta potencia utilizados en entornos industriales. Ahora bien, el uso de este tipo de hardware introduce componentes no lineales en la dinámica del robot, dificultando así su modelado matemático e impidiendo, por tanto, la aplicación de esquemas de control clásico por par de fuerzas [17]. Así pues, en la parte del software, se necesitan nuevos controladores adaptativos que no dependan de un modelado previo de la dinámica del robot y capaces de tolerar las no linealidades. Además de los condicionantes impuestos por el hardware, los cobots tienen que desenvolverse en escenarios no estructurados debido a la naturaleza

imprevisible de la interacción humana, contrastando, una vez más, con los escenarios industriales clásicos de entornos estructurados y bien acotados.

Otra diferencia entre los escenarios robóticos clásicos y las nuevas aplicaciones es el enlace de comunicación entre el robot y el controlador. Tradicionalmente, un cable dedicado, punto a punto, establece la comunicación robot-controlador. Actualmente, nuevos tipos de conexión son cada vez más comunes: comunicación inalámbrica, enlaces en la nube (*cloud robotics*), teleoperación o control remoto. Estos nuevos esquemas de comunicación, que implican ventajas como el ahorro de cableado o una mayor flexibilidad de la infraestructura, conllevan retardos temporales que afectan a la transmisión de información en el ciclo de control [20], [21], [22]. La naturaleza no estructurada de los nuevos escenarios robóticos queda acentuada por esta latencia estocástica presente en el ciclo de control, a tener en cuenta en el diseño de controladores.

Así pues, la robótica colaborativa se caracteriza por el uso de actuadores de baja potencia y componentes elásticos, y por la necesidad de controladores adaptativos capaces de lidiar con las no linealidades de la dinámica del robot y escenarios no estructurados. Características, todas ellas, presentes de manera natural en el control motor que animales y humanos ejercemos sobre nuestro cuerpo. Controlamos elementos altamente no lineales como son los músculos y el resto de tejidos blandos, interactuamos con un entorno en constante cambio que exige nuestra adaptación continua, nuestras acciones son de naturaleza no estructurada, y los retardos temporales son fisiológicamente inevitables en la transmisión de información en el sistema nervioso. Por tanto, una línea de trabajo hacia el éxito de los robots colaborativos será hacer su comportamiento lo más parecido posible al nuestro. El estudio del control motor biológico puede servir como fuente de inspiración para el desarrollo de controladores adaptativos para cobots.

### *Neurorobótica*

La percepción de estímulos, procesamiento de información, y la planificación y ejecución de respuestas motoras constituyen un complejo proceso regulado por diferentes áreas del sistema nervioso (e.g., la corteza motora está implicada en el control de movimientos voluntarios [31], los ganglios basales juegan un papel en la toma de decisiones [32], la médula espinal regula la actividad muscular y genera respuestas reflejas [34], etc.). El conocimiento aportado por la neurociencia sobre las diferentes áreas nerviosas puede ser replicado utilizando modelos computacionales, inspirando así el desarrollo de soluciones tecnológicas que doten a los robots de capacidades que para nosotros son innatas. A su vez, la integración de estos modelos computacionales en ciclos de control robótico proporciona un entorno experimental con el que validar los modelos y poder estudiar el funcionamiento nervioso [29]. Esta simbiosis entre neurociencia y robótica es lo que llamamos *neurorobótica*.

De todas las áreas del sistema nervioso involucradas en el control motor, una de ellas es especialmente aplicable a la robótica: el cerebelo. Por un lado tiene una estructura neuronal muy regular que facilita su modelado computacional, y por otro lado su papel reconocido en el aprendizaje motor lo hace apropiado para control robótico adaptativo [35]. La teoría Marr-Albus-Ito sobre el cerebelo y su capacidad de adaptación motora [41], [42], [43], ampliamente aceptada en la comunidad científica, funda las bases sobre los mecanismos de aprendizaje supervisado en el cerebelo [33]. En el aprendizaje supervisado la respuesta del sistema a un estímulo se compara con el resultado deseado, generando una señal de aprendizaje que regula los parámetros internos del propio sistema. Este proceso iterativo permite que el sistema se ajuste hasta minimizar la diferencia entre la respuesta real y la deseada. En el cerebelo la plasticidad sináptica permite regular las conexiones neuronales en función de los estímulos sensorimotors de entrada y la evaluación de la correspondiente respuesta motora. El cerebelo nos permite modificar nuestras acciones para que sus consecuencias coincidan con nuestras expectativas [46].

Diferentes tipos de modelos computacionales del cerebelo han sido propuestos como controladores adaptativos con anterioridad [35]. De entre las diferentes propuestas, son los modelos de nivel celular los que ofrecen una visión más completa sobre el cerebelo, así como los que mejor aprovechan el conocimiento neurológico previo ya que modelan el comportamiento biológico a nivel de neurona. Las redes neuronales de impulsos (SNN, *spiking neural network*) son las redes neuronales artificiales más biológicamente plausibles ya que modelan la transmisión y procesado de información al igual que lo hacen las neuronas biológicas: mediante la temporización precisa de impulsos (potenciales de acción en la membrana celular) [61]. Las SNNs heredan las propiedades de las redes neuronales biológicas al replicar sus dinámicas a nivel celular, pero suponen un alto coste computacional en comparación con otros enfoques limitados a replicar la funcionalidad. En cuanto a la robótica, el coste computacional de las SNN ha limitado sus aplicaciones a escenarios simulados [99] o de baja resolución [130]. En esta tesis, solventamos esa limitación y presentamos un modelo de cerebelo biológicamente plausible, basado en SNN, aplicado a un problema de control robótico real: control adaptativo por par de fuerzas de seis grados de libertad, aplicado a un robot de dinámica no lineal, operando en tiempo real y robusto a retardos temporales estocásticos que afectan a la comunicación robot-controlador.

En un primer momento, integramos un modelo de cerebelo SNN (~62k neuronas LIF, *leaky integrate and fire*, y ~36M de sinapsis) en el ciclo de control de un robot Baxter, funcionando en tiempo real. El modelo de cerebelo SNN cuenta con las principales poblaciones neuronales de la corteza del cerebelo (*mossy fibers*, MF; *granule cells*, GC; *Purkinje cells*, PC, *climbing fibers*, CF; *deep cerebellar nuclei*, DCN) y su principal mecanismo de aprendizaje (*spike-timing-dependent plasticity*, STDP). La red SNN se

implementa utilizando el simulador de redes neuronales de impulsos EDLUT [68]. Una vez integrado en el ciclo de control, probamos la capacidad del modelo SNN de controlar por par de fuerzas los seis grados de libertad del brazo de Baxter, ejecutando con precisión diferentes tareas motoras. El brazo de Baxter es de dinámica no lineal debido al uso de actuadores elásticos en serie en cada articulación, y muelles pasivos en una de sus articulaciones; por tanto, el modelo de cerebelo SNN se demuestra capaz de controlar no linealidades, de interés para la robótica colaborativa. Después de comprobar su precisión, enfrentamos el controlador ante diferentes escenarios no estructurados: interacción directa humano-robot y cambios en la dinámica de la planta (un peso extra añadido al extremo del brazo robótico, una fuerza elástica que se opone al movimiento). En ambos casos, el controlador SNN es capaz de adaptar su funcionamiento y ofrecer una interacción segura. Estos resultados, presentados en el *capítulo 2*, demuestran la aplicabilidad del modelo de cerebelo SNN como controlador adaptativo por par de fuerzas, capaz de operar con precisión seis grados de libertad de dinámica no lineal, y de adaptar su comportamiento a cambios de contexto e interactuar de manera segura con humanos.

#### *Controlador SNN basado en cerebelo. Control robusto ante retardos temporales*

Los retardos temporales presentes en el sistema nervioso implican que el cerebelo recibe información sensorimotora pasada y genera órdenes motoras que serán ejecutadas en el futuro [106]. Este retardo de ida y vuelta (~50 – 150 ms) obliga el comportamiento predictivo del cerebelo: es capaz de correlacionar información pasada con comandos futuros y mantener un control motor preciso [79]. En el *capítulo 3* comprobamos la capacidad del modelo de cerebelo SNN de tolerar retardos temporales en la comunicación robot-controlador. Al mantener los mecanismos de aprendizaje y adaptación e incluir el retardo sensorimotor del cerebelo biológico, el controlador SNN es capaz de operar de manera precisa incluso en presencia de retardos temporales. Introducimos diferentes latencias en el ciclo de control (retardos constantes simétricos y asimétricos, retardos estocásticos debidos a conexión inalámbrica y enlace remoto) y comprobamos cómo el controlador SNN mantiene los niveles de precisión obtenidos en condiciones ideales sin latencia.

#### *Más allá del cerebelo: médula espinal y músculos*

Finalmente, en el *capítulo 4* ampliamos el enfoque bioinspirado basado en cerebelo añadiendo al controlador SNN otros componentes del control motor biológico. En primer lugar, integramos un modelo de médula espinal junto con el modelo de cerebelo. La médula espinal utiliza información sobre el estado de los músculos para regular la actividad de estos, y este ciclo de retroalimentación rápida permite también la generación de respuestas reflejas [34]. Para incluir la información muscular, sustituimos el robot como planta del ciclo de control por un modelo de brazo musculoesquelético, dotado de dos grados de libertad controlados por la actividad de siete músculos. Utilizar un modelo musculoesquelético también nos ofrece un entorno biomecánico más

representativo en el que estudiar la complementariedad del cerebelo y la médula. Comprobamos cómo la médula espinal mejora la adaptación motora del cerebelo y su capacidad de generar reflejos rápidos ofrece una mejor respuesta ante perturbaciones (fuerzas externas que se oponen al movimiento deseado). Así pues, la integración cerebelo-médula amplía los beneficios del controlador SNN basado en cerebelo presentados en capítulos anteriores.

Para poder integrar la médula en el ciclo de control robótico, presentamos también la inclusión de dinámicas musculares en cada articulación de Baxter. Estos músculos simulados permiten modificar el grado de rigidez del movimiento del brazo al contraer simultáneamente el músculo agonista (flexor) y antagonista (extensor). De este modo, variando el grado de cocontracción se consigue realizar un mismo movimiento con distinta rigidez [217]. Resultados preliminares muestran cómo el cerebelo SNN es capaz de aprender a controlar el brazo de Baxter utilizando diferentes perfiles de cocontracción, y cómo estos perfiles ofrecen diferentes respuestas ante interacciones externas. Proponemos las dinámicas musculares y el control adaptativo del modelo de cerebelo SNN como un controlador capaz de modificar el grado de rigidez del movimiento según convenga; una cocontracción baja puede equipararse a control por admitancia, mientras que una cocontracción alta es comparable a control por impedancia.

### *Discusión*

En esta tesis presentamos una red SNN basada cerebelo aplicada como controlador robótico. El modelo de cerebelo SNN se ha demostrado válido como controlador adaptativo por par de fuerzas de un robot de dinámica no lineal, ofreciendo además un rendimiento seguro para la interacción con humanos y robusto ante retardos sensorimotrices. Además, también presentamos resultados preliminares de añadir al controlador basado en cerebelo otras partes del sistema nervioso central y el sistema musculoesquelético: médula espinal y dinámicas musculares.

La interacción física humano-robot está impulsando el uso de componentes elásticos que añaden seguridad pasiva a costa de hacer más compleja la dinámica del robot; la elasticidad introduce no linealidades que dificultan el modelado matemático de la dinámica. Por tanto, se necesitan nuevos controladores que no dependan de soluciones analíticas. Esta exigencia puede ser atendida utilizando redes neuronales clásicas (ANNs) [17], [72]; en esta tesis presentamos un controlador SNN basado en cerebelo. Proponemos un enfoque biológicamente inspirado que se beneficia de cómo el control motor biológico consigue, de manera natural, controlar las no linealidades introducidas por las dinámicas viscoelásticas de músculos, tendones, ligamentos, y resto de tejidos blandos.

El cerebelo ha sido propuesto con anterioridad como controlador adaptativo utilizando diferentes enfoques (modelos basados en codificación de estados, modelos funcionales, modelos de nivel celular [35]). Los modelos de nivel celular ofrecen una visión más profunda del substrato biológico, y por tanto pueden beneficiarse en mayor medida de

las características del cerebelo y del conocimiento ofrecido por la neurociencia. Sin embargo, el alto coste computacional de estos modelos los ha relegado a escenarios simulados [99], [129], redes de tamaño reducido [29], y señales de control de baja resolución [130]. En el trabajo aquí presentado, abordamos el desafío de aplicar a un problema real de robótica un modelo de cerebelo de nivel celular. En primer lugar, superamos el reto de ejecutar en tiempo real una red SNN de tamaño suficientemente grande (~62k neuronas LIF y ~36M de sinapsis), permitiendo operar con señales de control de alta resolución adecuadas para el control por par de fuerzas de seis grados de libertad. Esto conlleva al segundo reto superado en esta tesis: control adaptativo y preciso por par de fuerzas de seis grados de libertad de un robot no lineal (Baxter). Además, se comprueba la capacidad del controlador SNN de lidiar con perturbaciones motoras e interacción física humano-robot, demostrando resultados satisfactorios en ambos casos. De este modo, el controlador SNN basado en cerebelo queda validado como controlador adaptativo y seguro (*capítulo 2*).

En este punto cabe realizar una mención especial al simulador de redes neuronales de impulsos EDLUT [68], pieza clave del trabajo aquí presentado. Propiedades del cerebelo biológico relacionadas con el control motor (aprendizaje, adaptación, control predictivo) han podido ser aplicadas a un escenario de control robótico gracias a la capacidad de EDLUT de operar en tiempo real. El diseño específico de EDLUT para cumplir con este requisito permite la ejecución de redes SNN operando con señales de control de entrada y salida de alta resolución. Se necesitan simuladores SNN eficientes para incorporar a ciclos de control redes neuronales biológicamente plausibles, fundamental para el avance de la neurobótica. EDLUT se ha demostrado capaz de resolver este problema sin necesidad de utilizar hardware específico, haciendo uso únicamente de computación convencional por CPU y GPU.

Una vez validado el rendimiento del controlador SNN en términos de precisión y seguridad, pasamos a comprobar su capacidad de operar bajo retardos temporales. El comportamiento predictivo del cerebelo permite tolerar los retardos sensorimotors presentes en el sistema nervioso de manera natural. Esta solidez ante latencias en la comunicación sería beneficiosa para escenarios de control robótico afectados por retardos, un desafío de la robótica actual debido al creciente uso de comunicaciones inalámbricas [20], en la nube (*cloud-robotics*) [21], conexiones remotas o teleoperación [22]. Introducimos retardos sensorimotors en el ciclo de control de Baxter y comprobamos la respuesta del controlador SNN. Los retardos introducidos obedecían a diferentes escenarios: latencia constante y variable bajo control en el laboratorio, y retardos no deterministas debidos a casos de uso reales como conexión Wi-Fi y conexión de larga distancia a través de Internet. Los resultados demuestran que el controlador SNN basado en cerebelo ofrece un comportamiento estable a pesar de la presencia de latencias de diferente naturaleza (*capítulo 3*).

La simbiosis entre neurociencia y robótica no se limita al cerebelo. La neurorobótica puede beneficiarse del conocimiento adquirido sobre las distintas partes del sistema nervioso involucradas en el control motor, desde la toma de decisiones, a la planificación y ejecución de movimientos. En el *capítulo 4* presentamos resultados

preliminares de la integración del modelo SNN de cerebelo junto con un modelo de médula espinal que implementa inhibición recíproca entre músculos antagonistas y reflejo miotático (i.e., respuesta refleja que se produce ante el estiramiento de un músculo). La integración del modelo de médula facilita el aprendizaje y adaptación del modelo de cerebelo y mejora la respuesta ante perturbaciones motoras. La integración de los dos modelos se ha llevado a cabo en un entorno simulado en el que el control motor se ejercía sobre un modelo musculoesquelético de brazo (dos grados de libertad y siete músculos). El trabajo futuro se centrará en incorporar el modelo de médula espinal al ciclo de control robótico. También en el *capítulo 4* añadimos dinámicas musculares a cada articulación de Baxter; una línea de trabajo orientada a modificar la rigidez del movimiento variando el grado de cocontracción muscular. Los resultados preliminares incluidos en esta tesis dibujan un camino prometedor hacia un controlador capaz de cubrir una gama de control por admitancia y por impedancia. Cambiar entre admitancia o impedancia dependerá de las exigencias del entorno de interacción física humano-robot: el control por admitancia implica un movimiento suave del robot, favoreciendo así la intención de movimiento del humano; el control por impedancia ofrece una respuesta rígida ante interacciones, favoreciendo la intención de movimiento del robot [218]. El trabajo futuro irá orientado a utilizar señales de recompensa en el cerebelo [220] para aprender diferentes perfiles de rigidez de movimiento, permitiendo así que un mismo controlador ofrezca diferentes comportamientos. Para elegir qué grado de rigidez utilizar en cada momento se puede incorporar al ciclo de control un modelo de ganglios basales [224]; un área del sistema nervioso involucrada en la toma de decisiones. El modelo de ganglios basales produciría una señal de contexto capaz de diferenciar entre distintos grados de control por admitancia o impedancia; esa señal serviría como señal de entrada al cerebelo por vía de las fibras musgosas, y el cerebelo aprendería a generar comandos motores precisos para los distintos grados de rigidez. Estos comandos motores se enviarían al modelo de médula espinal, que regularía la activación de la dinámica muscular, traducida finalmente en comandos de par de fuerzas enviados a los actuadores del robot. Así pues, el trabajo futuro está orientado a ampliar el enfoque bioinspirado utilizado en la tesis.

#### *Resumen de los objetivos alcanzados*

En esta tesis se han alcanzado los siguientes objetivos específicos:

- Desarrollo mediante SNN de un modelo de cerebelo adaptado a seis grados de libertad. Desarrollado utilizando el simulador EDLUT (*capítulo 2*).
- Integración en tiempo real del modelo SNN de cerebelo en el ciclo de control del robot Baxter (*capítulo 2*).
- Validación de la capacidad de aprendizaje del modelo SNN de cerebelo como controlador por par de fuerzas. El controlador SNN aprende los comandos de par de fuerzas necesarios para ejecutar diferentes movimientos que implican a los seis grados de libertad del brazo de Baxter (*capítulos 2 y 3*).

- Validación de la precisión del controlador SNN. El aprendizaje motor proporciona un alto grado de precisión en los movimientos (*capítulos 2 y 3*).
- Validación de la capacidad de adaptación del controlador SNN ante escenarios no estructurados. El controlador SNN adapta su comportamiento a cambios dinámicos debidos a modificaciones del contexto: una carga adicional de 0.5 kg, fuerzas elásticas externas que se oponen al movimiento (*capítulo 2*).
- Validación del comportamiento seguro del controlador SNN. El modelo de cerebelo SNN permite la interacción física humano-robot de manera segura (*capítulo 2*).
- Validación de la solidez del controlador SNN ante retardos temporales. El controlador SNN es capaz de controlar el robot a pesar de la presencia de retardos temporales que afectan a la comunicación robot-controlador. El control por par de fuerzas del modelo de cerebelo es estable en presencia de retardos temporales, tanto constantes como estocásticos, en entornos controlados de laboratorio y casos de usos reales (comunicación Wi-Fi, conexión remota vía Internet). La tolerancia del cerebelo a los retardos temporales puede modificarse a costa de cambios en la precisión del movimiento: una mayor tolerancia conlleva una reducción en la precisión (*capítulo 3*).
- Integración del modelo de cerebelo SNN y un modelo de médula espinal (*capítulo 4*).
- Validación de la integración cerebelo – médula espinal como controlador de un modelo de brazo musculoesquelético. El modelo de médula espinal facilita la adaptación y aprendizaje del cerebelo (*capítulo 4*).
- Validación de la mejora en la respuesta ante perturbaciones motoras del modelo integrado de cerebelo y médula espinal. La integración de la médula espinal ofrece una respuesta más sólida ante perturbaciones externas (*capítulo 4*).
- Integración de dinámicas musculares en el ciclo de control de Baxter (*capítulo 4*).
- Validación del control motor utilizando diferentes perfiles de cocontracción muscular. El modelo de cerebelo es capaz de aprender a controlar con precisión el brazo robótico utilizando diferentes grados de cocontracción. Los diferentes grados de cocontracción modifican la rigidez del brazo, permitiendo modificar la solidez de la respuesta del brazo ante perturbaciones externas (*capítulo 4*).

### *Conclusión*

Los robots son herramientas que han servido a la actividad humana desde hace décadas; de su aplicación original en entornos industriales a aplicaciones más recientes como

servicios de asistencia sanitaria, social o educativa. La complejidad de las aplicaciones de los robots crecerá conforme su comportamiento sea más cercano al comportamiento humano, de naturaleza adaptativa y flexible como aún no se ha visto en robótica. El estudio del comportamiento humano puede inspirar el desarrollo de robots más avanzados.

La neurociencia irá ampliando las respuestas detrás de la cognición, emoción y acción. Este conocimiento sobre el substrato biológico responsable de nuestro comportamiento puede ser replicado utilizando herramientas computacionales. De este modo, la biología puede inspirar soluciones tecnológicas: réplicas artificiales de partes del sistema nervioso integradas en ciclos de control robótico. La robótica puede beneficiarse de los millones de años de evolución que nos han permitido ser lo que somos. A su vez, el uso de redes neuronales artificiales biológicamente plausibles hace de estos robots plataformas experimentales con las que estudiar el funcionamiento del sistema nervioso, disfunción neurológica y tratamientos. Se cierra así el círculo de la neurobótica; los robots son herramientas que nos pueden asistir en tareas físicas, pero también pueden ayudarnos a entender nuestra propia naturaleza.



# BIBLIOGRAPHY

- [1] K. Capek, *RUR (Rossum's universal robots)*: Penguin, 2004.
- [2] K. Capek, N. Comrada, and B. Comrada, "About the Word Robot," 1993.
- [3] J. Delmerico, S. Mintchev, A. Giusti, B. Gromov, K. Melo, T. Horvat, *et al.*, "The current state and future outlook of rescue robotics," *Journal of Field Robotics*, vol. 36, pp. 1171-1191, 2019.
- [4] V. der Loos, H. Machiel, D. J. Reinkensmeyer, and E. Guglielmelli, "Rehabilitation and health care robotics," in *Springer handbook of robotics*, ed: Springer, 2016, pp. 1685-1728.
- [5] K. Wada and T. Shibata, "Living with seal robots—its sociopsychological and physiological influences on the elderly at a care house," *IEEE transactions on robotics*, vol. 23, pp. 972-980, 2007.
- [6] P. Baxter, E. Ashurst, R. Read, J. Kennedy, and T. Belpaeme, "Robot education peers in a situated primary school study: Personalisation promotes child learning," *PloS one*, vol. 12, p. e0178126, 2017.
- [7] S. Haddadin and E. Croft, "Physical human–robot interaction," in *Springer handbook of robotics*, ed: Springer, 2016, pp. 1835-1874.
- [8] A. Vysocky and P. Novak, "Human-Robot collaboration in industry," *MM Science Journal*, vol. 9, pp. 903-906, 2016.
- [9] K. Melo and L. Paez, "Modular snake robot gaits on horizontal pipes," in *2012 IEEE/RSJ international conference on intelligent robots and systems*, 2012, pp. 3099-3104.
- [10] A. Spröwitz, A. Tuleu, M. Vespignani, M. Ajallooeian, E. Badri, and A. J. Ijspeert, "Towards dynamic trot gait locomotion: Design, control, and experiments with Cheetah-cub, a compliant quadruped robot," *The International Journal of Robotics Research*, vol. 32, pp. 932-950, 2013.
- [11] Z. Ren, W. Hu, X. Dong, and M. Sitti, "Multi-functional soft-bodied jellyfish-like swimming," *Nature communications*, vol. 10, pp. 1-12, 2019.
- [12] A. J. Ijspeert, "Biorobotics: Using robots to emulate and investigate agile locomotion," *science*, vol. 346, pp. 196-203, 2014.
- [13] K. Karakasiliotis, R. Thandiackal, K. Melo, T. Horvat, N. K. Mahabadi, S. Tsitkov, *et al.*, "From cineradiography to biorobots: an approach for designing

- robots to emulate and study animal locomotion," *Journal of The Royal Society Interface*, vol. 13, p. 20151089, 2016.
- [14] J. A. Nyakatura, K. Melo, T. Horvat, K. Karakasiliotis, V. R. Allen, A. Andikfar, *et al.*, "Reverse-engineering the locomotion of a stem amniote," *Nature*, vol. 565, pp. 351-355, 2019.
- [15] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, pp. 467-475, 2015.
- [16] G. Gu, J. Zou, R. Zhao, X. Zhao, and X. Zhu, "Soft wall-climbing robots," *Science Robotics*, vol. 3, p. eaat2874, 2018.
- [17] H. Chaoui, P. Sicard, and W. Gueaieb, "ANN-based adaptive control of robotic manipulators with friction and joint elasticity," *IEEE Transactions on Industrial Electronics*, vol. 56, pp. 3174-3187, 2009.
- [18] A. Diamond, R. Knight, D. Devereux, and O. Holland, "Anthropomimetic robots: Concept, construction and modelling," *International Journal of Advanced Robotic Systems*, vol. 9, p. 209, 2012.
- [19] C. Bartolozzi, G. Indiveri, and E. Donati, "Embodied neuromorphic intelligence," *Nature Communications*, vol. 13, pp. 1-14, 2022.
- [20] N. J. Ploplys, P. A. Kawka, and A. G. Alleyne, "Closed-loop control over wireless networks," *IEEE Control Systems Magazine*, vol. 24, pp. 58-71, 2004.
- [21] S. Shen, A. Song, and T. Li, "Predictor-based motion tracking control for cloud robotic systems with delayed measurements," *Electronics*, vol. 8, p. 398, 2019.
- [22] G. Niemeyer, C. Preusche, S. Stramigioli, and D. Lee, "Telerobotics," in *Springer handbook of robotics*, ed: Springer, 2016, pp. 1085-1108.
- [23] F. Barranco, C. Fermuller, and Y. Aloimonos, "Bio-inspired motion estimation with event-driven sensors," in *International Work-Conference on Artificial Neural Networks*, 2015, pp. 309-321.
- [24] X. Zhang, Y. Zhuo, Q. Luo, Z. Wu, R. Midya, Z. Wang, *et al.*, "An artificial spiking afferent nerve based on Mott memristors for neurorobotics," *Nature communications*, vol. 11, pp. 1-9, 2020.
- [25] Z. Bing, A. E. Sewisy, G. Zhuang, F. Walter, F. O. Morin, K. Huang, *et al.*, "Toward Cognitive Navigation: Design and Implementation of a Biologically Inspired Head Direction Cell Network," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
- [26] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: a review," *Neural networks*, vol. 21, pp. 642-653, 2008.

- [27] A. J. Ijspeert, A. Crespi, D. Ryczko, and J.-M. Cabelguen, "From swimming to walking with a salamander robot driven by a spinal cord model," *science*, vol. 315, pp. 1416-1420, 2007.
- [28] R. Thandiackal, K. Melo, L. Paez, J. Herault, T. Kano, K. Akiyama, *et al.*, "Emergence of robust self-organized undulatory swimming based on local hydrodynamic force sensing," *Science Robotics*, vol. 6, p. eabf6354, 2021.
- [29] F. Naveros, N. R. Luque, E. Ros, and A. Arleo, "VOR adaptation on a humanoid iCub robot using a spiking cerebellar model," *IEEE transactions on cybernetics*, vol. 50, pp. 4744-4757, 2019.
- [30] R. I. Schubotz and D. von Cramon, "Functional-anatomical concepts of human premotor cortex: evidence from fMRI and PET studies," *Neuroimage*, vol. 20, p. S120, 2003.
- [31] J. N. Sanes and J. P. Donoghue, "Plasticity and primary motor cortex," *Annual review of neuroscience*, vol. 23, pp. 393-415, 2000.
- [32] K. Takakusaki, K. Saitoh, H. Harada, and M. Kashiwayanagi, "Role of basal ganglia–brainstem pathways in the control of motor behaviors," *Neuroscience research*, vol. 50, pp. 137-151, 2004.
- [33] J. L. Raymond and J. F. Medina, "Computational principles of supervised learning in the cerebellum," *Annual review of neuroscience*, vol. 41, pp. 233-253, 2018.
- [34] E. Pierrot-Deseilligny and D. Burke, *The circuitry of the human spinal cord: spinal and corticospinal mechanisms of movement*: Cambridge University Press, 2012.
- [35] M. Arbib, G. Metta, and P. v. d. Smagt, "Neurorobotics: From vision to action," in *Springer handbook of robotics*, ed: Springer Verlag, 2008.
- [36] F. A. Azevedo, L. R. Carvalho, L. T. Grinberg, J. M. Farfel, R. E. Ferretti, R. E. Leite, *et al.*, "Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain," *Journal of Comparative Neurology*, vol. 513, pp. 532-541, 2009.
- [37] E. Clarke and J. Stannard, "Aristotle on the anatomy of the brain," *Journal of the history of medicine and allied sciences*, pp. 130-148, 1963.
- [38] S. R. Cajal, "Histologie du syste me nerveux de l'Homme et des verte be s," *Maloine (Paris)*, vol. 2, pp. 891-942, 1911.
- [39] P. Cabaraux, J. Gandini, and M. Manto, "The Three Cornerstones of Cerebellar Ataxia: Closing the Loop of 200 Years of Cerebellar Research," in *Cerebellum as a CNS Hub*, ed: Springer, 2021, pp. 459-478.

- [40] J. Voogd and M. Glickstein, "The anatomy of the cerebellum," *Trends in cognitive sciences*, vol. 2, pp. 307-313, 1998.
- [41] D. Marr and W. T. Thach, "A theory of cerebellar cortex," in *From the Retina to the Neocortex*, ed: Springer, 1991, pp. 11-50.
- [42] M. Ito, "Neurophysiological aspects of the cerebellar motor control system," *Int. J. Neurol.*, vol. 7, pp. 126-179, 1970.
- [43] J. S. Albus, "A theory of cerebellar function," *Mathematical biosciences*, vol. 10, pp. 25-61, 1971.
- [44] T. D. Sanger, O. Yamashita, and M. Kawato, "Expansion coding and computation in the cerebellum: 50 years after the Marr–Albus codon theory," *The Journal of Physiology*, vol. 598, pp. 913-928, 2020.
- [45] M. Ito, M. Sakurai, and P. Tongroach, "Climbing fibre induced depression of both mossy fibre responsiveness and glutamate sensitivity of cerebellar Purkinje cells," *The Journal of physiology*, vol. 324, pp. 113-134, 1982.
- [46] J. F. Medina, "Teaching the cerebellum about reward," *Nature neuroscience*, vol. 22, pp. 846-848, 2019.
- [47] H. L. More and J. M. Donelan, "Scaling of sensorimotor delays in terrestrial mammals," *Proceedings of the Royal Society B*, vol. 285, p. 20180613, 2018.
- [48] D. M. Wolpert and Z. Ghahramani, "Computational principles of movement neuroscience," *Nature neuroscience*, vol. 3, pp. 1212-1217, 2000.
- [49] A. J. Bastian, "Learning to predict the future: the cerebellum adapts feedforward movement control," *Current opinion in neurobiology*, vol. 16, pp. 645-649, 2006.
- [50] R. Shadmehr, M. A. Smith, and J. W. Krakauer, "Error correction, sensory prediction, and adaptation in motor control," *Annual review of neuroscience*, vol. 33, pp. 89-108, 2010.
- [51] D. M. Wolpert, R. C. Miall, and M. Kawato, "Internal models in the cerebellum," *Trends in cognitive sciences*, vol. 2, pp. 338-347, 1998.
- [52] N. R. Luque, J. A. Garrido, R. R. Carrillo, S. Tolu, and E. Ros, "Adaptive cerebellar spiking model embedded in the control loop: Context switching and robustness against noise," *International journal of neural systems*, vol. 21, pp. 385-401, 2011.
- [53] T. Yamazaki, "Evolution of the Marr-Albus-Ito Model," in *Cerebellum as a CNS Hub*, ed: Springer, 2021, pp. 239-255.

- [54] Z. Gao, B. J. Van Beugen, and C. I. De Zeeuw, "Distributed synergistic plasticity and cerebellar learning," *Nature Reviews Neuroscience*, vol. 13, pp. 619-635, 2012.
- [55] T. Yamazaki and W. Lennon, "Revisiting a theory of cerebellar cortex," *Neuroscience research*, vol. 148, pp. 1-8, 2019.
- [56] J. D. Schmahmann, "The cerebellum and cognition," *Neuroscience letters*, vol. 688, pp. 62-75, 2019.
- [57] G. Dede and M. H. Sazlı, "Speech recognition with artificial neural networks," *Digital Signal Processing*, vol. 20, pp. 763-768, 2010.
- [58] F. Ahmad, N. A. Mat Isa, Z. Hussain, M. K. Osman, and S. N. Sulaiman, "A GA-based feature selection and parameter optimization of an ANN in diagnosing breast cancer," *Pattern Analysis and Applications*, vol. 18, pp. 861-870, 2015.
- [59] L. Fridman, D. E. Brown, M. Glazer, W. Angell, S. Dodd, B. Jenik, *et al.*, "MIT advanced vehicle technology study: Large-scale naturalistic driving study of driver behavior and interaction with automation," *IEEE Access*, vol. 7, pp. 102021-102038, 2019.
- [60] A. Krogh, "What are artificial neural networks?," *Nature biotechnology*, vol. 26, pp. 195-197, 2008.
- [61] S. Ghosh-Dastidar and H. Adeli, "Third generation neural networks: Spiking neural networks," in *Advances in Computational Intelligence*, ed: Springer, 2009, pp. 167-178.
- [62] W. Gerstner and W. M. Kistler, *Spiking neuron models: Single neurons, populations, plasticity*: Cambridge university press, 2002.
- [63] D. V. Buonomano and M. D. Mauk, "Neural network model of the cerebellum: temporal discrimination and the timing of motor responses," *Neural computation*, vol. 6, pp. 38-55, 1994.
- [64] W. Maass, "Networks of spiking neurons: the third generation of neural network models," *Neural networks*, vol. 10, pp. 1659-1671, 1997.
- [65] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, pp. 668-673, 2014.
- [66] E. Ros, R. Carrillo, E. M. Ortigosa, B. Barbour, and R. Agís, "Event-driven simulation scheme for spiking neural networks using lookup tables to characterize neuronal dynamics," *Neural computation*, vol. 18, pp. 2959-2993, 2006.

- [67] F. Naveros, N. R. Luque, J. A. Garrido, R. R. Carrillo, M. Anguita, and E. Ros, "A spiking neural simulator integrating event-driven and time-driven computation schemes using parallel CPU-GPU co-processing: a case study," *IEEE transactions on neural networks and learning systems*, vol. 26, pp. 1567-1574, 2014.
- [68] F. Naveros, J. A. Garrido, R. R. Carrillo, E. Ros, and N. R. Luque, "Event-and time-driven techniques using parallel CPU-GPU co-processing for spiking neural networks," *Frontiers in neuroinformatics*, vol. 11, p. 7, 2017.
- [69] J. J. Craig, "Force control of manipulators," in *Introduction to robotics: mechanics and control*. vol. 3, ed: Prentice Hall, 2005, pp. 317-338.
- [70] C. Fitzgerald, "Developing baxter," in *2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*, 2013, pp. 1-6.
- [71] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi, "An atlas of physical human-robot interaction," *Mechanism and Machine Theory*, vol. 43, pp. 253-270, 2008.
- [72] R. M. Robinson, C. S. Kothera, R. M. Sanner, and N. M. Wereley, "Nonlinear control of robotic manipulators driven by pneumatic artificial muscles," *IEEE/ASME Transactions on Mechatronics*, vol. 21, pp. 55-68, 2015.
- [73] M. Ito and M. Itō, *The cerebellum and neural control*: Raven press, 1984.
- [74] M. G. Paulin, "The role of the cerebellum in motor control and perception," *Brain, behavior and evolution*, vol. 41, pp. 39-50, 1993.
- [75] J. S. Albus, "A new approach to manipulator control: The cerebellar model articulation controller (CMAC)," 1975.
- [76] J. C. Houk, J. T. Buckingham, and A. G. Barto, "Models of the cerebellum and motor learning," *Behavioral and brain sciences*, vol. 19, pp. 368-383, 1996.
- [77] D. M. Wolpert and M. Kawato, "Multiple paired forward and inverse models for motor control," *Neural networks*, vol. 11, pp. 1317-1329, 1998.
- [78] N. Schweighofer, "Computational models of the cerebellum in the adaptive control of movements," University of Southern California, 1995.
- [79] N. R. Luque, J. A. Garrido, R. R. Carrillo, J.-M. C. Olivier, and E. Ros, "Cerebellarlike corrective model inference engine for manipulation tasks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, pp. 1299-1312, 2011.
- [80] J. A. Garrido, E. Ros, and E. D'Angelo, "Spike timing regulation on the millisecond scale by distributed synaptic plasticity at the cerebellum input stage: a simulation study," *Frontiers in computational neuroscience*, vol. 7, p. 64, 2013.

- [81] J. A. Garrido, N. R. Luque, S. Tolu, and E. D'Angelo, "Oscillation-driven spike-timing dependent plasticity allows multiple overlapping pattern recognition in inhibitory interneuron networks," *International journal of neural systems*, vol. 26, p. 1650020, 2016.
- [82] N. R. Luque, F. Naveros, R. R. Carrillo, E. Ros, and A. Arleo, "Spike burst-pause dynamics of Purkinje cells regulate sensorimotor adaptation," *PLoS computational biology*, vol. 15, p. e1006298, 2019.
- [83] J. A. Garrido Alcazar, N. R. Luque, E. D'Angelo, and E. Ros, "Distributed cerebellar plasticity implements adaptable gain control in a manipulation task: a closed-loop robotic simulation," *Frontiers in Neural Circuits*, vol. 7, p. 159, 2013.
- [84] N. R. Luque, J. A. Garrido, R. R. Carrillo, E. D'Angelo, and E. Ros, "Fast convergence of learning requires plasticity between inferior olive and deep cerebellar nuclei in a manipulation task: a closed-loop robotic simulation," *Frontiers in computational neuroscience*, vol. 8, p. 97, 2014.
- [85] M. Ito, "Cerebellar microcomplexes," *International review of neurobiology*, vol. 41, pp. 475-487, 1997.
- [86] R. R. Carrillo, F. Naveros, E. Ros, and N. R. Luque, "A metric for evaluating neural input representation in supervised learning networks," *Frontiers in Neuroscience*, p. 913, 2018.
- [87] T. Yamazaki and S. Tanaka, "Computational models of timing mechanisms in the cerebellar granular layer," *The cerebellum*, vol. 8, pp. 423-432, 2009.
- [88] T. Honda, T. Yamazaki, S. Tanaka, S. Nagao, and T. Nishino, "Stimulus-dependent state transition between synchronized oscillation and randomly repetitive burst in a model cerebellar granular layer," *PLoS computational biology*, vol. 7, p. e1002087, 2011.
- [89] T. Ishikawa, M. Shimuta, and M. Häusser, "Multimodal sensory integration in single cerebellar granule cells in vivo," *Elife*, vol. 4, p. e12916, 2015.
- [90] N. R. Luque, R. R. Carrillo, F. Naveros, J. A. Garrido, and M. Sáez-Lara, "Integrated neural and robotic simulations. Simulation of cerebellar neurobiological substrate for an object-oriented dynamic model abstraction process," *Robotics and Autonomous Systems*, vol. 62, pp. 1702-1716, 2014.
- [91] N. R. Luque, J. A. Garrido, J. Ralli, J. J. Laredo, and E. Ros, "From sensors to spikes: Evolving receptive fields to enhance sensorimotor information in a robot-arm," *International journal of neural systems*, vol. 22, p. 1250013, 2012.
- [92] G. A. Pratt and M. M. Williamson, "Series elastic actuators," in *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, 1995, pp. 399-406.

- [93] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, 2004, pp. 2149-2154.
- [94] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, *et al.*, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, 2009, p. 5.
- [95] E. Salinas and R. Romo, "Conversion of sensory signals into motor commands in primary motor cortex," *Journal of Neuroscience*, vol. 18, pp. 499-511, 1998.
- [96] W. Thach, "Correlation of neural discharge with pattern and force of muscular activity, joint position, and direction of intended next movement in motor cortex and cerebellum," *Journal of neurophysiology*, vol. 41, pp. 654-676, 1978.
- [97] S. H. Scott and J. F. Kalaska, "Reaching movements with similar hand paths but different arm orientations. I. Activity of individual cells in motor cortex," *Journal of neurophysiology*, vol. 77, pp. 826-852, 1997.
- [98] S. Chitta, I. Sucan, and S. Cousins, "Moveit![ros topics]," *IEEE Robotics & Automation Magazine*, vol. 19, pp. 18-19, 2012.
- [99] N. R. Luque, J. A. Garrido, F. Naveros, R. R. Carrillo, E. D'Angelo, and E. Ros, "Distributed cerebellar motor learning: a spike-timing-dependent plasticity model," *Frontiers in computational neuroscience*, vol. 10, p. 17, 2016.
- [100] S. Kuroda, K. Yamamoto, H. Miyamoto, K. Doya, and M. Kawato, "Statistical characteristics of climbing fiber spikes necessary for efficient cerebellar learning," *Biological cybernetics*, vol. 84, pp. 183-192, 2001.
- [101] R. R. Carrillo, E. Ros, C. Boucheny, and J.-M. C. Olivier, "A real-time spiking cerebellum model for learning robot control," *Biosystems*, vol. 94, pp. 18-27, 2008.
- [102] W.-K. Li, M. J. Hausknecht, P. Stone, and M. D. Mauk, "Using a million cell simulation of the cerebellum: Network scaling and task generality," *Neural networks*, vol. 47, pp. 95-102, 2013.
- [103] M. Uusisaari and E. De Schutter, "The mysterious microcircuitry of the cerebellar nuclei," *The Journal of physiology*, vol. 589, pp. 3441-3457, 2011.
- [104] J. F. Medina and M. D. Mauk, "Simulations of cerebellar motor learning: computational analysis of plasticity at the mossy fiber to deep nucleus synapse," *Journal of Neuroscience*, vol. 19, pp. 7140-7151, 1999.
- [105] J. F. Medina and M. D. Mauk, "Computer simulation of cerebellar information processing," *nature neuroscience*, vol. 3, pp. 1205-1211, 2000.

- [106] A. Sargolzaei, M. Abdelghani, K. K. Yen, and S. Sargolzaei, "Sensorimotor control: computing the immediate future from the delayed present," *BMC bioinformatics*, vol. 17, pp. 501-509, 2016.
- [107] M. Kawato and H. Gomi, "A computational model of four regions of the cerebellum based on feedback-error learning," *Biological cybernetics*, vol. 68, pp. 95-103, 1992.
- [108] C. Boucheny, R. Carrillo, E. Ros, and O. J. Coenen, "Real-time spiking neural network: An adaptive cerebellar model," in *International Work-Conference on Artificial Neural Networks*, 2005, pp. 136-144.
- [109] M. Gerwig, K. Hajjar, A. Dimitrova, M. Maschke, F. P. Kolb, M. Frings, *et al.*, "Timing of conditioned eyeblink responses is impaired in cerebellar patients," *Journal of Neuroscience*, vol. 25, pp. 3919-3931, 2005.
- [110] D. L. Mills, "Internet time synchronization: the network time protocol," *IEEE Transactions on communications*, vol. 39, pp. 1482-1493, 1991.
- [111] P. van der Smagt, "Benchmarking cerebellar control," *Robotics and Autonomous Systems*, vol. 32, pp. 237-251, 2000.
- [112] H. Hoffmann, G. Petkos, S. Bitzer, and S. Vijayakumar, "Sensor-assisted adaptive motor control under continuously varying context," in *ICINCO-ICSO*, 2007, pp. 262-269.
- [113] N. Schweighofer, M. A. Arbib, and M. Kawato, "Role of the cerebellum in reaching movements in humans. I. Distributed inverse dynamics control," *European Journal of Neuroscience*, vol. 10, pp. 86-94, 1998.
- [114] A. Karniel and G. F. Inbar, "A model for learning human reaching movements," *Biological cybernetics*, vol. 77, pp. 173-183, 1997.
- [115] S. Vijayakumar, A. D'souza, T. Shibata, J. Conradt, and S. Schaal, "Statistical learning for humanoid robots," *Autonomous Robots*, vol. 12, pp. 55-69, 2002.
- [116] C. Williams, S. Klanke, S. Vijayakumar, and K. Chai, "Multi-task gaussian process learning of robot inverse dynamics," *Advances in neural information processing systems*, vol. 21, 2008.
- [117] S. Schaal, C. G. Atkeson, and S. Vijayakumar, "Scalable techniques from nonparametric statistics for real time robot learning," *Applied Intelligence*, vol. 17, pp. 49-60, 2002.
- [118] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *Journal of neuroscience*, vol. 5, pp. 1688-1703, 1985.
- [119] E. D. Wilson, T. Assaf, M. J. Pearson, J. M. Rossiter, S. R. Anderson, J. Porrill, *et al.*, "Cerebellar-inspired algorithm for adaptive control of nonlinear dielectric

- elastomer-based artificial muscle," *Journal of The Royal Society Interface*, vol. 13, p. 20160547, 2016.
- [120] P. Dean, J. Porrill, C.-F. Ekerot, and H. Jörntell, "The cerebellar microcircuit as an adaptive filter: experimental and computational evidence," *Nature Reviews Neuroscience*, vol. 11, pp. 30-43, 2010.
- [121] S. Vijayakumar and S. Schaal, "Locally weighted projection regression: An  $o(n)$  algorithm for incremental real time learning in high dimensional space," in *Proceedings of the seventeenth international conference on machine learning (ICML 2000)*, 2000, pp. 288-293.
- [122] T. Sinkjaer, C. Wu, A. G. Barto, and J. C. Houk, "Cerebellar control of endpoint position—a simulation model," in *1990 IJCNN International Joint Conference on Neural Networks*, 1990, pp. 705-710.
- [123] F. Chao, D. Zhou, C.-M. Lin, L. Yang, C. Zhou, and C. Shang, "Type-2 fuzzy hybrid controller network for robotic systems," *IEEE Transactions on Cybernetics*, vol. 50, pp. 3778-3792, 2019.
- [124] C. Nie, Z. Zheng, and M. Zhu, "Three-dimensional path-following control of a robotic airship with reinforcement learning," *International Journal of Aerospace Engineering*, vol. 2019, 2019.
- [125] J. Cao, W. Liang, J. Zhu, and Q. Ren, "Control of a muscle-like soft actuator via a bioinspired approach," *Bioinspiration & biomimetics*, vol. 13, p. 066005, 2018.
- [126] S. Tolu, M. Vanegas, J. A. Garrido, N. R. Luque, and E. Ros, "Adaptive and predictive control of a simulated robot arm," *International journal of neural systems*, vol. 23, p. 1350010, 2013.
- [127] A. Lenz, S. R. Anderson, A. G. Pipe, C. Melhuish, P. Dean, and 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)*, vol. 39, pp. 1420-1433, 2009.
- [128] H. T. Kalidindi, T. G. Thuruthel, C. Laschi, and E. Falotico, "Cerebellum-inspired approach for adaptive kinematic control of soft robots," in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, 2019, pp. 684-689.
- [129] A. Bouganis and M. Shanahan, "Training a spiking neural network to control a 4-dof robotic arm based on spike timing-dependent plasticity," in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 2010, pp. 1-8.
- [130] A. Antonietti, D. Martina, C. Casellato, E. D'Angelo, and A. Pedrocchi, "Control of a humanoid nao robot by an adaptive bioinspired cerebellar module in 3d motion tasks," *Computational intelligence and neuroscience*, vol. 2019, 2019.

- [131] S. Wang, W. Chaovaitwongse, and R. Babuska, "Machine learning algorithms in bipedal robot control," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, pp. 728-743, 2012.
- [132] J. P. Hwang and E. Kim, "Robust tracking control of an electrically driven robot: adaptive fuzzy logic approach," *IEEE Transactions on Fuzzy systems*, vol. 14, pp. 232-247, 2006.
- [133] Z. Bing, C. Meschede, K. Huang, G. Chen, F. Rohrbein, M. Akl, *et al.*, "End to end learning of spiking neural network based on r-stdp for a lane keeping vehicle," in *2018 IEEE international conference on robotics and automation (ICRA)*, 2018, pp. 4725-4732.
- [134] D. D. Ligutan, A. C. Abad, and E. P. Dadios, "Adaptive robotic arm control using artificial neural network," in *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, 2018, pp. 1-6.
- [135] R. de Azambuja, A. Cangelosi, and S. V. Adams, "Diverse, noisy and parallel: a new spiking neural network approach for humanoid robot control," in *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 1134-1142.
- [136] M. Hernando and E. Gambao, "Teleprogramming: Capturing the Intention of the Human Operator," in *Advances in Telerobotics*, ed: Springer, 2007, pp. 303-320.
- [137] K. G. Shin and X. Cui, "Computing time delay and its effects on real-time control systems," *IEEE Transactions on control systems technology*, vol. 3, pp. 218-224, 1995.
- [138] A. Sorribes, "Sistemas dinámicos con retardos temporales," Tesis doctoral, UPV, Valencia, Espana, 2011.
- [139] V. A. Oliveira, L. V. Cossi, M. C. Teixeira, and A. M. Silva, "Synthesis of PID controllers for a class of time delay systems," *Automatica*, vol. 45, pp. 1778-1782, 2009.
- [140] M. Bozorg and F. Termeh, "Domains of PID controller coefficients which guarantee stability and performance for LTI time-delay systems," *Automatica*, vol. 47, pp. 2122-2125, 2011.
- [141] H. Xu, A. Datta, and S. Bhattacharyya, "PID stabilization of LTI plants with time-delay," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, 2003, pp. 4038-4043.
- [142] J. E. Normey-Rico and E. F. Camacho, "Dead-time compensators: A survey," *Control engineering practice*, vol. 16, pp. 407-428, 2008.
- [143] O. J. Smith, "Closed control of loop with dead time," *Chemical engineering progress*, vol. 53, pp. 217-219, 1957.

- [144] M. Bowthorpe, M. Tavakoli, H. Becher, and R. Howe, "Smith predictor-based robot control for ultrasound-guided teleoperated beating-heart surgery," *IEEE Journal of biomedical and Health Informatics*, vol. 18, pp. 157-166, 2013.
- [145] A. Manitius and A. Olbrot, "Finite spectrum assignment problem for systems with delays," *IEEE transactions on Automatic Control*, vol. 24, pp. 541-552, 1979.
- [146] I. Vörös, B. Várszegi, and D. Takács, "Lane keeping control using finite spectrum assignment with modeling errors," in *Dynamic Systems and Control Conference*, 2019, p. V003T18A002.
- [147] J. Eyre, S. Miller, and V. Ramesh, "Constancy of central conduction delays during development in man: investigation of motor and somatosensory pathways," *The Journal of physiology*, vol. 434, pp. 441-452, 1991.
- [148] J. Stein, "Cerebellar forward models to control movement," *The Journal of Physiology*, vol. 587, p. 299, 2009.
- [149] I. Abadia, F. Naveros, J. A. Garrido, E. Ros, and N. R. Luque, "On robot compliance: a cerebellar control approach," *IEEE transactions on cybernetics*, vol. 51, pp. 2476-2489, 2019.
- [150] M. L. Latash, "Muscle coactivation: definitions, mechanisms, and functions," *Journal of neurophysiology*, vol. 120, pp. 88-104, 2018.
- [151] T. Hirano, "Long-term depression and other synaptic plasticity in the cerebellum," *Proceedings of the Japan Academy, Series B*, vol. 89, pp. 183-195, 2013.
- [152] R. Kettner, S. Mahamud, H.-C. Leung, N. Sitkoff, J. Houk, B. Peterson, *et al.*, "Prediction of complex two-dimensional trajectories by a cerebellar model of smooth pursuit eye movement," *Journal of neurophysiology*, vol. 77, pp. 2115-2130, 1997.
- [153] R. S. Sutton and A. G. Barto, "Toward a modern theory of adaptive networks: expectation and prediction," *Psychological review*, vol. 88, p. 135, 1981.
- [154] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE transactions on systems, man, and cybernetics*, pp. 834-846, 1983.
- [155] J. L. Raymond and S. G. Lisberger, "Neural learning rules for the vestibulo-ocular reflex," *Journal of Neuroscience*, vol. 18, pp. 9112-9129, 1998.
- [156] V. Lev-Ram, S. B. Mehta, D. Kleinfeld, and R. Y. Tsien, "Reversing cerebellar long-term depression," *Proceedings of the National Academy of Sciences*, vol. 100, pp. 15989-15993, 2003.

- [157] J. Duchateau and S. Baudry, "Maximal discharge rate of motor units determines the maximal rate of force development during ballistic contractions in human," *Frontiers in human neuroscience*, vol. 8, p. 234, 2014.
- [158] I. Abadía, F. Naveros, E. Ros, R. R. Carrillo, and N. R. Luque, "A cerebellar-based solution to the nondeterministic time delay problem in robotic control," *Science Robotics*, vol. 6, p. eabf2756, 2021.
- [159] S. Ostojic and N. Brunel, "From spiking neuron models to linear-nonlinear models," *PLoS computational biology*, vol. 7, p. e1001056, 2011.
- [160] T. Yamazaki and S. Tanaka, "Neural modeling of an internal clock," *Neural computation*, vol. 17, pp. 1032-1058, 2005.
- [161] T. Yamazaki and S. Tanaka, "The cerebellum as a liquid state machine," *Neural Networks*, vol. 20, pp. 290-297, 2007.
- [162] T. Yamazaki and S. Tanaka, "A spiking network model for passage-of-time representation in the cerebellum," *European Journal of Neuroscience*, vol. 26, pp. 2279-2292, 2007.
- [163] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *trans. ASME*, vol. 64, 1942.
- [164] I. Herreros, X. Arsiwalla, and P. Verschure, "A forward model at Purkinje cell synapses facilitates cerebellar anticipatory control," *Advances in Neural Information Processing Systems*, vol. 29, 2016.
- [165] A. Mukherjee, "On the dynamics and significance of low frequency components of Internet load," 1992.
- [166] N. A. Ali, E. Ekram, A. Eljasmy, and K. Shuaib, "Measured delay distribution in a wireless mesh network test-bed," in *2008 IEEE/ACS International Conference on Computer Systems and Applications*, 2008, pp. 236-240.
- [167] C. Pei, Y. Zhao, G. Chen, R. Tang, Y. Meng, M. Ma, *et al.*, "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*, 2016, pp. 1-9.
- [168] J. Dugan, S. Elliott, B. A. Mah, J. Poskanzer, and K. Prabhu, "iPerf-The ultimate speed test tool for TCP," *UDP and SCTP*, 2014.
- [169] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A survey of research on cloud robotics and automation," *IEEE Transactions on automation science and engineering*, vol. 12, pp. 398-409, 2015.
- [170] G. Hu, W. P. Tay, and Y. Wen, "Cloud robotics: architecture, challenges and applications," *IEEE network*, vol. 26, pp. 21-28, 2012.

- [171] M. Rohde and M. O. Ernst, "Time, agency, and sensory feedback delays during action," *Current Opinion in Behavioral Sciences*, vol. 8, pp. 193-199, 2016.
- [172] M. Rohde, G. Altan, and M. O. Ernst, "When vision lags, motor prediction follows," *bioRxiv*, 2020.
- [173] S. Franklin, J. Česonis, R. Leib, and 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)*, 2019, pp. 1517-1520.
- [174] A. Giovannucci, A. Badura, B. Deverett, F. Najafi, T. D. Pereira, Z. Gao, *et al.*, "Cerebellar granule cells acquire a widespread predictive feedback signal during motor learning," *Nature neuroscience*, vol. 20, pp. 727-734, 2017.
- [175] M. Shidara, K. Kawano, H. Gomi, and M. Kawato, "Inverse-dynamics model eye movement control by Purkinje cells in the cerebellum," *Nature*, vol. 365, pp. 50-52, 1993.
- [176] H. Imamizu, S. Miyauchi, T. Tamada, Y. Sasaki, R. Takino, B. PuÈtz, *et al.*, "Human cerebellar activity reflecting an acquired internal model of a new tool," *Nature*, vol. 403, pp. 192-195, 2000.
- [177] C. Casellato, A. Antonietti, J. A. Garrido, A. Pedrocchi, and 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*, 2014, pp. 813-818.
- [178] E. D. Wilson, T. Assaf, M. J. Pearson, J. M. Rossiter, P. Dean, S. R. Anderson, *et al.*, "Biohybrid control of general linear systems using the adaptive filter model of cerebellum," *Frontiers in neurorobotics*, vol. 9, p. 5, 2015.
- [179] J. Porrill and P. Dean, "Recurrent cerebellar loops simplify adaptive control of redundant and nonlinear motor systems," *Neural computation*, vol. 19, pp. 170-193, 2007.
- [180] S. Tolu, M. C. Capolei, L. Vannucci, C. Laschi, E. Falotico, and M. V. Hernández, "A cerebellum-inspired learning approach for adaptive and anticipatory control," *International journal of neural systems*, vol. 30, p. 1950028, 2020.
- [181] M. Asadi-Eydivand, M. M. Ebadzadeh, M. Solati-Hashjin, C. Darlot, and N. A. Abu Osman, "Cerebellum-inspired neural network solution of the inverse kinematics problem," *Biological Cybernetics*, vol. 109, pp. 561-574, 2015.
- [182] H. T. Kalidindi, L. Vannucci, C. Laschi, and E. Falotico, "Cerebellar adaptive mechanisms explain the optimal control of saccadic eye movements," *Bioinspiration & Biomimetics*, vol. 16, p. 016004, 2020.

- [183] A. Salimi-Badr, M. M. Ebadzadeh, and 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*, vol. 111, pp. 421-438, 2017.
- [184] P. Van Der Smagt, M. A. Arbib, and G. Metta, "Neurorobotics: From vision to action," in *Springer Handbook of Robotics*, ed: Springer, 2016, pp. 2069-2094.
- [185] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*: MIT press, 2018.
- [186] D. Narain, E. D. Remington, C. I. D. Zeeuw, and M. Jazayeri, "A cerebellar mechanism for learning prior distributions of time intervals," *Nature communications*, vol. 9, pp. 1-12, 2018.
- [187] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business & information systems engineering*, vol. 6, pp. 239-242, 2014.
- [188] D. M. Lofaro, A. Asokan, and 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)*, 2015, pp. 519-526.
- [189] R. Imai and R. Kubo, "Cloud-based remote motion control over FTTH networks for home robotics," in *2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE)*, 2014, pp. 565-566.
- [190] N. Tian, A. K. Tanwani, K. Goldberg, and S. Sojoudi, "Mitigating network latency in cloud-based teleoperation using motion segmentation and synthesis," in *International Symposium on Robotics Research*, 2019.
- [191] F. Voigtländer, A. Ramadan, J. Eichinger, C. Lenz, D. Pensky, and A. Knoll, "5G for robotics: Ultra-low latency control of distributed robotic systems," in *2017 International Symposium on Computer Science and Intelligent Controls (ISCSIC)*, 2017, pp. 69-72.
- [192] J. C. Mankins, "Technology readiness levels," *White Paper, April*, vol. 6, p. 1995, 1995.
- [193] B. A. Richards and T. P. Lillicrap, "Dendritic solutions to the credit assignment problem," *Current opinion in neurobiology*, vol. 54, pp. 28-36, 2019.
- [194] A. Suvrathan, H. L. Payne, and J. L. Raymond, "Timing rules for synaptic plasticity matched to behavioral function," *Neuron*, vol. 92, pp. 959-967, 2016.
- [195] A. Berthoz and R. Llinás, "Afferent neck projection to the cat cerebellar cortex," *Experimental Brain Research*, vol. 20, pp. 385-401, 1974.
- [196] D. Menghini, M. Di Paola, R. Murri, F. Costanzo, C. Caltagirone, S. Vicari, *et al.*, "Cerebellar vermis abnormalities and cognitive functions in individuals with

- Williams syndrome," *Research in developmental disabilities*, vol. 34, pp. 2118-2126, 2013.
- [197] C. J. Stoodley and J. D. Schmahmann, "Functional topography of the human cerebellum," *Handbook of clinical neurology*, vol. 154, pp. 59-70, 2018.
- [198] C. J. Stoodley, E. M. Valera, and J. D. Schmahmann, "Functional topography of the cerebellum for motor and cognitive tasks: an fMRI study," *Neuroimage*, vol. 59, pp. 1560-1570, 2012.
- [199] R. Apps and M. Garwicz, "Anatomical and physiological foundations of cerebellar information processing," *Nature Reviews Neuroscience*, vol. 6, pp. 297-311, 2005.
- [200] J. Merel, M. Botvinick, and G. Wayne, "Hierarchical motor control in mammals and machines," *Nature communications*, vol. 10, pp. 1-12, 2019.
- [201] C. L. Ebbesen and M. Brecht, "Motor cortex—to act or not to act?," *Nature Reviews Neuroscience*, vol. 18, pp. 694-705, 2017.
- [202] H. J. Groenewegen, "The basal ganglia and motor control," *Neural plasticity*, vol. 10, pp. 107-120, 2003.
- [203] J. Korpelainen, H. Huikuri, K. Sotaniemi, and V. Myllylä, "Abnormal heart rate variability reflecting autonomic dysfunction in brainstem infarction," *Acta Neurologica Scandinavica*, vol. 94, pp. 337-342, 1996.
- [204] J. C. Smith, A. P. Abdala, A. Borgmann, I. A. Rybak, and J. F. Paton, "Brainstem respiratory networks: building blocks and microcircuits," *Trends in neurosciences*, vol. 36, pp. 152-162, 2013.
- [205] A. Rowald, S. Komi, R. Demesmaeker, E. Baaklini, S. D. Hernandez-Charpak, E. Paoles, *et al.*, "Activity-dependent spinal cord neuromodulation rapidly restores trunk and leg motor functions after complete paralysis," *Nature Medicine*, vol. 28, pp. 260-271, 2022.
- [206] P. Perrini, G. Tiezzi, M. Castagna, and R. Vannozzi, "Three-dimensional microsurgical anatomy of cerebellar peduncles," *Neurosurgical review*, vol. 36, pp. 215-225, 2013.
- [207] M. Glickstein and K. Doron, "Cerebellum: connections and functions," *The Cerebellum*, vol. 7, pp. 589-594, 2008.
- [208] R. M. Brownstone, T. V. Bui, and N. Stifani, "Spinal circuits for motor learning," *Current opinion in neurobiology*, vol. 33, pp. 166-173, 2015.
- [209] P. Matthews, "Muscle spindles and their motor control," *Physiological Reviews*, vol. 44, pp. 219-288, 1964.

- [210] Y. Laporte and D. P. Lloyd, "Nature and significance of the reflex connections established by large afferent fibers of muscular origin," *American Journal of Physiology-Legacy Content*, vol. 169, pp. 609-621, 1952.
- [211] E. Jankowska and A. Lundberg, "Interneurones in the spinal cord," *Trends in Neurosciences*, vol. 4, pp. 230-233, 1981.
- [212] S. L. Delp, F. C. Anderson, A. S. Arnold, P. Loan, A. Habib, C. T. John, *et al.*, "OpenSim: open-source software to create and analyze dynamic simulations of movement," *IEEE transactions on biomedical engineering*, vol. 54, pp. 1940-1950, 2007.
- [213] K. R. Saul, X. Hu, C. M. Goehler, M. E. Vidt, M. Daly, A. Velisar, *et al.*, "Benchmarking of dynamic simulation predictions in two software platforms using an upper limb musculoskeletal model," *Computer methods in biomechanics and biomedical engineering*, vol. 18, pp. 1445-1458, 2015.
- [214] A. Prochazka, "Quantifying proprioception," *Progress in brain research*, vol. 123, pp. 133-142, 1999.
- [215] F. E. Zajac, "Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control," *Critical reviews in biomedical engineering*, vol. 17, pp. 359-411, 1989.
- [216] J. M. Winters and S. L. Woo, "Multiple muscle systems," 1990.
- [217] G. Ganesh, A. Melendez-Calderon, M. Haruno, M. Kawato, and E. Burdet, "Transition between reciprocal activation and co-contraction during wrist posture control," 2020.
- [218] C. Ott, R. Mukherjee, and Y. Nakamura, "Unified impedance and admittance control," in *2010 IEEE international conference on robotics and automation*, 2010, pp. 554-561.
- [219] Ö. Ekeberg, "A combined neuronal and mechanical model of fish swimming," *Biological cybernetics*, vol. 69, pp. 363-374, 1993.
- [220] D. Kostadinov and M. Häusser, "Reward signals in the cerebellum: origins, targets, and functional implications," *Neuron*, 2022.
- [221] T. Valency and M. Zacksenhouse, "Accuracy/robustness dilemma in impedance control," *J. Dyn. Sys., Meas., Control*, vol. 125, pp. 310-319, 2003.
- [222] K. Melo, M. Garabini, G. Grioli, M. Catalano, L. Malagia, and A. Bicchi, "Open source VSA-CubeBots for rapid soft robot prototyping," in *Robot Makers-Workshop in conjunction with 2014 Robotics Science and Systems Conference, Berkeley, California, USA*, 2014.

- [223] M. Vespignani, K. Melo, M. Mutlu, and A. J. Ijspeert, "Compliant snake robot locomotion on horizontal pipes," in *2015 IEEE international symposium on safety, security, and rescue robotics (SSRR)*, 2015, pp. 1-8.
- [224] Á. González-Redondo, F. Naveros, E. Ros, and J. A. Garrido, "A Basal Ganglia Computational Model to Explain the Paradoxical Sensorial Improvement in the Presence of Huntington's Disease," *International Journal of Neural Systems*, vol. 30, p. 2050057, 2020.

