Bio-Inspired Motor Learning Models for Robot Control

Silvia Tolu
Department of Computer Architecture and Technology
University of Granada

A thesis submitted for the degree of
Doctor of Philosophy (PhD) in Computer Science

March 2012

Editor: Editorial de la Universidad de Granada
Autor: Silvia Tolu
D.L.: GR 2199-2012
ISBN: 978-84-9028-128-4

Declaración

Profesor Dr. Eduardo Ros Vidal, Catedrático de Universidad del Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada, Dra. Jo-Anne Ting, investigadora de Robert Bosch Research and Technology Center en Palo Alto, Estados Unidos, y Dr. Antonio Cañas Vargas, docente del Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada,

CERTIFICAN:

Que la memoria titulada "Bio-Inspired Motor Learning Models for Robot Control", ha sido realizada por Dña. Silvia Tolu bajo nuestra dirección en el Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada para optar al grado de Doctor por la Universidad de Granada.

Granada, 13 de febrero 2012

Fdo. Eduardo Ros Vidal

Fdo. Jo-Anne Ting

Fdo. Antonio Cañas Vargas

To Mauricio for his great love and to our baby for giving us the strength to go forward and chase our dreams.

To Sensei for his daily guidance.

"In qualsiasi circostanza vi troviate non cedete alla sconfitta. Non abbandonate ciò che avete raggiunto se vi trovate a un punto morto. Un grande futuro vi attende. Per questo motivo dovete perseverare e studiare."

Daisaku Ikeda.

Acknowledgements

Although many helped me during the process of making this PhD dissertation, it could not have been completed without the support, the patience, and the love of my husband Mauricio. I am deeply grateful to him because he understood all the times I needed to vent my frustration and my disappointment and he always encouraged me to look for my inner determination to succeed. I owe my mother many thanks for being always present in my life every day and for her huge heart.

I must thank my advisor Professor Eduardo Ros for his guidance, his inspiration, and his optimism every time I did not see any way out. He has provided me the tools to gradually become a researcher in a friendly and pleasant working environment. I would like to express my gratitude for my thesis director Dr. Jo-Anne Ting for listening to me in those hard days in Edinburgh and giving me several helpful suggestions.

I would like to thank Professor Sethu Vijayakumar, from the Institute of Perception, Action and Behaviour (IPAB) at the University of Edinburgh's School of Informatics and Dr. Stefan Klanke for helping me getting started with the LWPR algorithm and for their research advice and feedback.

I would like also to thank all my department colleagues (I do not want to name them for fear of omitting any of them) for all the moments that we shared and enjoyed together that distracted me from research and made me take the things easier. I express a special thank to Juan Manuel Gómez and Miguel Méndez for all the outdoor adventures (climbing and padel) that made remind me there are much more interests in life than thesis.

I am most grateful to my friend Ana for standing me in my bad moments during four years living together in Granada. I could not have survived those years without going shopping and dancing together, without trips and adventures, and without our long chats. Thanks to Ana for introducing me

into the fascinating world of the "pollos"; from this group of friends I learnt how to take life with a carefree, whereby I thank them.
Likewise, I must thank my family Soka of Spain and Italy and all my companions of faith of Granada and Genoa for being always together despite life's adversities, for encouraging and learning from each other to make us stronger and wiser.
Lastly, I thank the following projects/parties for providing funding that allowed me to make this thesis: the EU project SENSOPAC (IST-028056) and the Master and Back Program by the autonomous region of Sardinia (P.O.R. FSE 2007-2013).

Abstract

We propose two bio-inspired architectures for representing the role of the adaptive cerebellar microcircuit in correcting the motor behavior based on current errors: the feedforward and the recurrent schemes. While in the first architecture, the cerebellum adds adaptive torque corrections and its weights are adjusted depending on the motor-error signal, in the second one it adds sensory correction terms to the controller and its weights depend on the sensory error.

Biological systems perform adaptation of dynamics and kinematics models for accurate control with low power actuators, while in robotics, robots normally achieve very high precision and high speed motion with high forces and high energy consumption. This industrial approach cannot be used in the framework of human interaction applications because it is potentially dangerous in case of malfunctioning. Thus, adoption of compliant strategies of dynamics and kinematics models abstraction and adaptive control schemes in real time in the field of robotics are required. Furthermore, the cerebellum is able to acquire intrinsic models through experience by a perceptual feedback that allows the motor learning to proceed. These models are called internal models. Motivated by this, we implemented a cerebellum model able to adapt its corrections and store the sensory consequences or feedforward motor commands for predicting appropriate actions when needed. We addressed this study with a machine learning approach (LWPR algorithm) embedded in the control loops in which the LWPR module abstracts the functionality of the input layer to the cerebellar cortex. LWPR provides optimal input representation to the granular layer in terms of neural resources, that is, it adapts its neural resources incrementally and according to the input data structure. Moreover, we built compliant control systems combining the feedback error learning approach and adaptive predictive control.

This dissertation first introduces a description of the cerebellar microcircuitry and of the role of internal models in motor control. Then, we present the control systems based on inverse and forward internal model for controlling a robot arm in which a feedback error controller leads to a precise, compliant and stable control during manipulation of objects and the cerebellar-machine learning synergy makes the robot adaptable to changing conditions. Finally, we show the principal results obtained of the performance evaluation of both systems. We demonstrate the validity and efficiency of the models with experiments on a 3-DOF and 7-DOF light-weight robot arm.

Resumen

Proponemos dos arquitecturas bio-inspiradas para representar el papel del microcircuito adaptativo del cerebelo en la corrección del comportamiento motor basada en errores actuales: el esquema feedforward y recurrente. Mientras que en la primera arquitectura, el cerebelo añade correcciones adaptativas y sus pesos se ajustan en función de la señal de error motora, en la segunda añade términos sensoriales correctivos al controlador y sus pesos dependen del error sensorial.

Los sistemas biológicos realizan la adaptación de los modelos dinámicos y cinemáticos para un control preciso con actuadores de baja potencia, mientras que en robótica, los robots normalmente logran una precisión muy alta y un movimiento de alta velocidad con fuerzas muy grandes y un alto consumo de energía. Este enfoque industrial no puede ser utilizado en el marco de las aplicaciones de interacción humana, ya que es potencialmente peligroso en caso de mal funcionamiento. Así, se necesita adoptar estrategias *dóciles*[1] de modelos de abstracción cinemáticos y dinámicos y esquemas de control adaptativos en tiempo real en el campo de la robótica. Además, el cerebelo es capaz de adquirir los modelos intrínsecos a través de la experiencia dada por la retroalimentación perceptiva que permite al aprendizaje motor poder avanzar. Estos modelos se denominan modelos internos. Motivados por esto, hemos implementado un modelo de cerebelo capaz de adaptar sus correcciones y almacenar las consecuencias sensoriales o los comandos motores para predecir las acciones apropiadas cuando sea necesario. Hemos abordado este estudio con un enfoque de aprendizaje automático (algoritmo LWPR) incrustado en el lazo de control en el que el módulo LWPR abstrae la funcionalidad de la capa de entrada de la corteza cerebelar. LWPR proporciona una representación de entrada óptima para la capa granular en

[1]Traducción de la palabra inglesa **compliant**.

términos de recursos neuronales, es decir, adapta sus recursos neuronales progresivamente y de acuerdo con la estructura de entrada de los datos. Además, hemos construido los sistemas de control *dócil* combinando el enfoque de aprendizaje del error de retroalimentación y el control adaptativo predictivo.

Esta disertación primero presenta una descripción del microcircuito del cerebelo y del papel que desempeñan los modelos internos en el control motor. A seguir, se presentan los sistemas de control basados en los modelos inverso y directo para el control de un brazo robótico en el que un controlador de error de retroalimentación conduce a un control preciso, compatible y estable durante la manipulación de objetos y la sinergia de aprendizaje entre cerebelo y control automático hace que el robot sea adaptable a las condiciones cambiantes. Finalmente, se muestran los principales resultados obtenidos con la evaluación del rendimiento de ambos sistemas. Se demuestra la validez y eficacia de los modelos con los experimentos en un brazo robot de tres y siete grados de libertad.

Contents

List of Figures

List of Tables

Glossary

AFEL Adaptive Feedback Error Learning architecture. This is a control system based on inverse internal model learning, on a feedback error learning mechanism, on a machine learning algorithm and on a bio-inspired module for controlling a robot arm.

CF Climbing fiber. It is a type of nerve fiber that carries impulses to the Purkinje cells of the cerebellar cortex cells from the Inferior Olive cells.

CNS Central nervous system. It is the part of the nervous system that integrates the information and coordinates the activity of all parts of the body.

Compliance The ability of a robot to tolerate and compensate for misaligned parts. It is also the ability of distending in response to applied pressure.

DOF Degree of freedom. It is a defined mode in which a mechanical device or system can move. The number of degrees of freedom is given by the total number of independent displacements of motion.

GC Granule cell. It is one of the small neurons of the cortex of the cerebellum.

LF Learning feedback controller. It is a controller that generates adaptive feedback commands from the sensory errors avoiding high gains and complex reference structures.

MF Mossy fiber. It is one of the major inputs to cerebellum.

PC Purkinje cell. It is a class of neuron located in the cerebellar cortex.

PF Parallel fiber. It is a type of nerve fiber that arises from granule cells in the cerebellar cortex. It is the axon of a Granular cell.

RAFEL Recurrent Adaptive Feedback Error Learning architecture. This is a control system based on forward internal model learning, on a feedback error learning mechanism, on a machine learning algorithm and on a bio-inspired module for controlling a robot arm.

RF Receptive field. It is a region of space in which the presence of a stimulus will alter its output.

ULM Unit Learning Machine. A modular unit contained in the cerebellum that consists of a uniform set of neuronal circuits that is capable of learning.

1

Introduction

The main issues in the field of biomorphic and biomimetic motor control are how human movements are controlled by the motor system and how stability is maintained. A practical way to understand the human motor control is simulating a system endowed with the high level knowledge acquired until the moment and specific working hypothesis about biologically plausible control schemes. In biological systems motor learning takes many forms, such as learning new skills or adapting those already known to maintain performance, learning what movements to make and when to make them. The cerebellum seems to play a key role in modulating accurate and coordinated movements. For this reason, understanding the cerebellar mechanisms of learning and emulating them through bio-inspired architectures are two reciprocal processes of fundamental importance to develop sophisticated robotic systems able to carry out complex and accurate operations and movements in unstructured scenarios.

1.1 Motor Control and Motor Learning

Primary aim of robotics research is to build machines that could operate autonomously. First fully autonomous robots appeared in the second half of the 20th century and they are usually employed in dirty, heavy or dangerous jobs that people normally do in their absence. However, robots can not plan and control physical actions as flexibly as human beings do. Nowadays, researchers study how people control their bodies, how they make coordinated movements under changing conditions of the environment, how they hold an object or they do multiple tasks in order to endow robots with these human like features. For this, robotics and neuroscience play a conjoint role in developing bio-inspired approaches or models for high nonlinear systems. The goal is to understand the mechanisms of motor control and motor learning of the central nervous system (CNS) at the cell and neuronal network level to create coordinated and precise movements and to store the information processed.

Automatic control, particularly the application of feedback, has been fundamental in this sense and a great deal in this field is related to stability in order to design safer workplaces for interactions between humans and robots, to implement more effective methods or better robotics tools. Human motor control regards many activities, such as walking, looking (eye movements), reaching for objects, drawing and writing, speech production and posture; In this dissertation, we will focus on robot arm movements and we address both the problems of achieving compliant nonlinear control and simulating the neural processes in a high dimensional input space in real-time.

1.1.1 Perception and Motor Control

Our goal is to build a system architecture to move a robot arm and correct the error due to interactions between multiple links or changing dynamic or kinematic conditions.
One of the main functions of the brain is to extract information from sensory input, to organize and to use it to respond to a particular situation. In this process, sensory-motor complexes (experienced during a movement or any interaction with the environment) are organized in intrinsic models. These intrinsic models acquired through experience allow efficient and accurate movements. Models are acquired with a perceptual feedback that allows the motor learning to proceed as we argue in section 1.1.2. This is achieved

through repetitive training by means of which the motor skills are improved in terms of accuracy and smoothness of movements. The feedback indicates how much the movement is effective and subsequent movements are performed to reduce the error between the desired and the actual output. The control strategy employed in the cerebellum that takes advantage of the sensorial experience is called closed-loop or feedback control (e.g. the optokinetic nystagmus [2]). One open issue is how to use feedback effectively for motor control. The choice of gains is of particular concern in a system due to the very important question of stability and robustness of the system itself.

In this dissertation, a feedback error learning approach based on repetitive iterations of the same task is devised with very low gains for compliant control.

1.1.2 Prediction and Motor Learning

The motor learning is the "acquisition of information about movements (and other motor outputs), including what output to produce as well as how and when to produce it. Motor learning results in the formation of motor memories" (Shadmehr and Wise [2]). The problem to solve in controlling a dynamical system is to find out the input to the system that will achieve the desired behavior as output even under disturbances or changing environments. The cerebellum acts in this sense, in fact is able to adapt its output in every condition by capturing internal models which are mechanisms that can mimic the input-output characteristics, or their inverses, of the motor apparatus [3].

Given our goal of making compliant control, one important challenge is to combine feedback error learning approach and adaptive predictive control by implementing a cerebellum model able to adapt its corrections and store the sensory consequences or feedforward motor commands for predicting appropriate actions when needed.

1.2 The Cerebellum as an Adaptive Controller

Since the late 1960s, Marr [4] (1969), Albus [5] (1971), Sejnowski [6] (1977), Fujita [7] (1982), and other authors published their papers talking about the cerebellum as an adaptive filter controller and many of recent works used artificial neural networks as models for the cerebellum [8]. In computing an internal model, the CNS operates a

transformation from an input variable to an output variable encoding a map. Doing that, the CNS adjusts or optimizes its own parameters automatically. On the basis of samples encoded in the map, the cerebellum as adaptive controller predicts the future signal and generates a signal for controlling an object (a robot arm) along a desired trajectory. The main advantage of this approach is to allow the design of control systems that can operate in an unknown or changing environment, when the dynamical robot model is unknown. In other words, adaptive control allows monitoring and adapting the behavior of the controller, achieving better performance and compensating changes in the environment. When the CNS experiences novel dynamics and the internal model cannot predict the necessary forces or the sensory consequences, then it needs to be trained again. After that, it adapts and predicts correct outputs in this new scenario.

Internal models of dynamics are instantiated based on context [9] as they are what has been learned about a specific motor control for a specific machine. Apart from adaptation, another issue is the retention of internal models which is the CNS capability of recalling the appropriate internal model and using it to make predictions to make the movement. Therefore, after training and adaptation the internal model becomes encoded into a long-term memory. The progress of memorization is called consolidation. An important aspect to remark about the anatomy of the cerebellum is that its structure appears to be very regular and modular (see Chapter 3 for further details). So, the functions of the different parts of the cerebellum just depend on their input and output connections rather than on their local anatomy.

In this dissertation, we take into consideration all these concepts to develop a mechanism of adaptation and retention of internal models of dynamics by a machine learning algorithm (LWPR) [10, 11] for online function approximation. This is specially suitable in case of noise or partial data, to optimize the input space reducing the number of active neurons, and to generalize unseen contexts. The main disadvantages are given by the fact that it implements supervised learning, that means that we need a very good teacher, and a high number of parameters to tune. Nevertheless, LWPR allows online incremental learning in a robotic platform since it spatially exploits localized linear models to approximate nonlinear functions at a low computational cost. Therefore, the evaluation of the prediction value is quite fast allowing real-time learning. Besides, the

incremental training permits the acquisition and retention of different models without any restriction of contexts and without interferences among them. Furthermore, we exploit the LWPR capability of transforming the inputs in an optimal basis of clean and accurate signals to the Purkinje layer of a cerebellar like architecture (see section 3.2 of the Chapter 3) for improving the learning of the internal model.

1.2.1 Bio-Inspired Motor Learning Models

As we have previously commented, the adaptive cerebellar microcircuit is repeated throughout the cerebellar cortex and can be used to correct future motor behavior based on current errors. Two alternative models have been proposed for representing the role of the above mentioned circuit in motor plant compensation [12, 13, 14]. These are the feedforward and the recurrent architectures (Figs. 3.2a and 3.2b in Chapter 3). While in the first architecture, the cerebellum adds adaptive torque corrections and its weights are adjusted depending on the motor-error signal, in the second one the cerebellum output is used as a sensory correction to the controller and it depends on the sensory error. Thus, the forward architecture needs the right motor error which is given by the difference between desired and actual motor commands and it is not an easy value to compute specially in real motor task of complex biological systems [8]. This problem does not exist in the recurrent architecture which seems to be more biologically plausible according to neurophysiological evidence [8, 15].

In this dissertation, we present both bio-inspired architectures in which we implement the feedback error learning approach referred in subsection 1.1.1 and the neural processes mentioned in section 1.2; we also demonstrate how to solve the motor-error problem stated by Porrill and Dean [8] (2007) for compensation of plants of high degrees of freedom (DOF) in the forward architecture.

1.3 Thesis Motivation. Working Hypothesis

The nervous system has developed model-abstraction mechanisms that allow an accurate control based on previously experimented movements (dynamics and kinematics abstraction processes). The cerebellum is one of the most important nervous centre involved in tasks of precise and coordinated control. It seems a nervous centre able to

abstract models of objects that are being manipulated. It is important to note that biological systems achieve accurate control with low power actuators, while in robotics, robots normally achieve very high precision and high speed motion with high forces and high energy consumption. This allows controlling strategies that are rather independent of the dynamics models of the plant and object being manipulated. These robots are potentially dangerous in the field of human interaction because they lack real-time sensory feedback and the force of their engines is dangerous if the system becomes unstable. Thus, the field of robotics and assistance aimed at human interaction needs the development of new technologies and control schemes with low power actuators. Adoption of strategies of dynamics and kinematics models abstraction and adaptive control schemes in real time are required. For this, we study biological systems that perform adaptation of dynamics and kinematics models for accurate control and machine learning approaches for model abstraction. There are two principal issues to consider for the further development of this study: internal models and feedback control.

Ito [16] (2008) examined how a movement of a body part is controlled in the brain. He stated that a feedback controller in the motor cortex generates a command that drives the body part (robot arm) accordingly to the desired instruction. Then, the body part model is captured into an internal model in the cerebellum in order to precisely perform the control of the robot arm by referring to it. How the internal model is learnt in the cerebellum? Again, Ito [16] (2008) stated that the cerebellum is composed of many modules called microcomplexes, each of which is a Unit Learning Machine (ULM) made up of a structured neuronal circuits and it encodes an internal model. The input-output relationship of a ULM is adaptively modified by the climbing fibers (CFs) that convey the error signal. The output neuron of the cerebellar cortex is the Purkinje cell (PC), which integrates information carried by one CF. Motivated by these concerns, in our approach, we took advantage of the LWPR algorithm for nonlinear function approximation to incrementally learn and store the internal models of the robot arm and to solve the lack of a perfect analytical robot model. Furthermore, Schweighofer et al. [17] (2001) hypothesized that the cerebellar learning is facilitated by a GC sparse code, i.e. a neural code in which the ratio of active neurons is low at any time. Besides, Porrill and Dean [8] (2007) stated that both accuracy and learning speed could greatly

improved by optimizing the choice of the centers and transforming the internal sensory-motor representation to an optimal basis of receptive fields (RFs). Motivated by that, we exploited the LWPR features to implement a kind of plastic process in the granular layer, in order to efficiently minimize the required resources for the input-space mapping placing and adapting kernels to better represent the input-space with a limited number of them. In our models, LWPR RFs are used as a representation of the granular and molecular layers delivering clean and accurate signals to the Purkinje cell layer.

With regard to the feedback control, Kawato [18] (2009) stated that a feedback error learning mechanism is needed to compute the necessary commands from a desired movement. In fact, a feedback controller that generates good commands drives the supervised learning in the cerebellum for acquisition of the internal model through the CF in conjunction with the PF-PC synapses. The major drawback of feedback error learning according to Porrill and Dean [8] (2007) is that it requires complex reference structures for generic redundant and nonlinear systems. Some authors attempted to avoid this problem using high gains in the feedback loop [19]. Moreover, an analytical computation of the dynamics is complex and in the case of a high number of DOFs, precise dynamics parameters may be unknown. In this case, adaptive models are required for an accurate and stable control during manipulation. Traditional methods are no longer suitable for controlling the complex dynamics of the new generation of robots since the movement is influenced by the state variables of all the joints and the control becomes very complex and highly non-linear [20]. Nonlinearities can dominate the robot dynamics and the feedback gains have to be increased to compensate the resulting tracking error [21] for accurately following a predefined desired trajectory. This is dangerous regarding the system stability and implies non-compliant movements. Furthermore, high gains are unacceptable in autonomous and biological systems because they introduce destabilizing components provided the inherent feedback sensorimotor delays [8]. Furthermore, high gains generate large forces or in other words potentially dangerous non-compliant movements [22], making the robot less safe for the environment, mainly in the framework of human-robot interaction applications, and compromising the closed-loop stability [23]. In this work, we implement a Learning Feedback (LF) controller that generates adaptive feedback commands from the sensory errors avoiding

classic PID with high gains and complex reference structures [8], thus solving the "distal teacher problem" [24].

1.4 Objectives

This thesis is focused on studying the cerebellar internal models role in motor control and motor learning of robot arm movements. The main aim is to develop bio-inspired control schemes for complex robot systems with low power actuators that require adaptive models for accurate movements during the manipulation of objects or interaction with the world. Therefore, in this thesis specifically addresses the following objectives:

- Study and development of low-gains control schemes.
- Implementation of model abstraction schemes during object manipulation. Extraction of knowledge through experimentation.
- Study of the organization of models from sensorimotor representations.
- Schemes of bio-inspired control: integration of machine learning methodologies with biologically plausible control schemes.

1.5 Project Framework

The work described in this document has been developed within the European project "SENSOrimotor structuring of Perception and Action for emergent Cognition" (SENSOPAC) [25].
SENSOPAC project (funded under the EU Framework 6 IST Cognitive Systems Initiative) extended from January, 2006 to July, 2010 in collaboration with 12 institutions from 9 different countries. SENSOPAC project combined machine learning techniques and modeling of biological systems to develop a machine capable of abstracting cognitive notions from sensorimotor relationships during interactions with its environment, and of generalizing this knowledge to novel situations. Through active sensing and exploratory actions the machine discovers the sensorimotor relationships and consequently learns the intrinsic structure of its interactions with the world and unravel predictive and causal relationships. The project demonstrated how a robotic system can bootstrap its development by constructing generalization and discovering sensor-based abstractions,

based on neuroscience findings on tactile sensory data representation and processing. The project relies on the synergy between multiple scientific institutions who are leaders in their fields, building on interaction between neuroscience experimentalists, theoreticians, and roboticists, leading to a complete artificial cognitive system using biomimetic actuation, and bio-inspired sensing. The research group at the University of Granada was mainly involved in the development of the spiking neuron computation environment (EDLUT) [26, 27] and in the development of real-time biologically and computing control systems capable of carrying out manipulation tasks under various contexts. Figure 1.1 shows the module organization of the SENSOPAC project. University of Granada (and this work as a part) focused in the fourth module dealing between neurophysiologist (module 5) and more abstract and robotic systems (modules 1, 2, and 3).

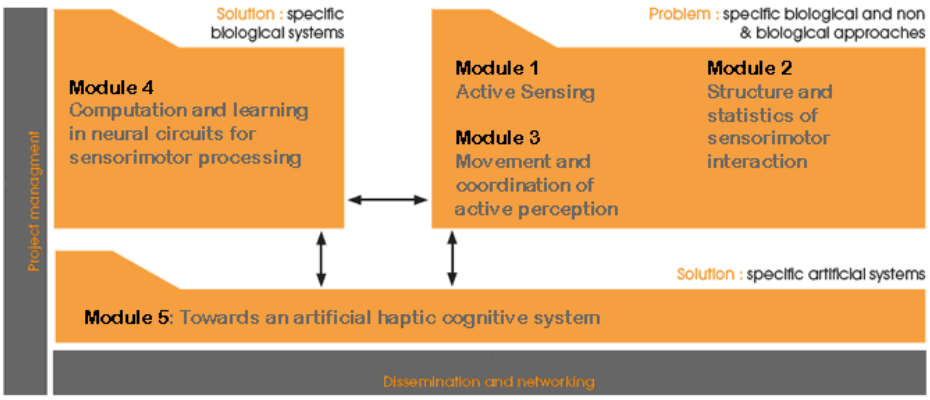


Figure 1.1: Module structure of the SENSOPAC European Project - Module diagram showing the main tasks developed in the framework of the project. The research group at the University of Granada was mainly involved in Module 4.

This work is highly interdisciplinary as it combines biological models with machine learning schemes. There are many efforts that are based on bio-inspired models, but in this project we adopt strategies from other fields such as machine learning and study its feasibility and benefits in the context of biologically plausible models. This project proposes the development of control schemes and acquisition of models based on an engine of abstraction of models and their validation with biomorphic robots. This represents a significant advance in the field of biomorphic control agents with low power actuators. The algorithm of machine learning used within the partners of the Sensopac project is the LWPR algorithm developed by Stefan Klanke, Sethu Vijayakumar and Stefan

Schaal. The doctorate spent 3 months at the University of Edinburgh through a research program exchange in order to get started with the LWPR algorithm and the robot configuration of the 7-DOF Light Weight Robot (LWR) arm designed by the Institute of Robotics and Mechatronics at the German Aerospace Centre (DLR).

1.6 Dissertation Outline

Chapters 4 and 5 describe the two approaches developed in this work. They both include a brief introduction and a material and methods section. Though they are very similar, we have kept them in both chapters to provide self-contained chapters (similar to papers describing these two developments).
The remainder of this dissertation is organized as follows:

- **Chapter 2**. This is the Spanish translation of the Introduction.
- **Chapter 3**. Adaptive control. Description of the cerebellar microcircuitry and of the role of internal models in motor control; presentation of the adaptive architectures that emulate the role of the cerebellum in motor adaptation.
- **Chapter 4**. Feedforward corrective actions. Adaptive Feedback Error Learning Architecture for Motor Control. This is a control system based on inverse internal model learning, on a feedback error learning mechanism, on a machine learning algorithm and on a bio-inspired module for controlling a robot arm. Finally, we present the experiments performed and its corresponding outcome evaluation.
- **Chapter 5**. Feedback corrective contributions. Presentation of a recurrent adaptive architecture based on forward internal model learning in which a feedback error controller leads a precise, compliant and stable control during manipulation of objects. The cerebellar-machine learning synergy makes the robot adaptable to changing conditions. Finally, we present the experiments performed and its corresponding outcome evaluation.
- **Chapter 6**. Concludes with a summary of contributions presented in this thesis and a discussion of future work.
- **Chapter 7**. This is the Spanish translation of the Conclusion.

2

Introducción en español

Los movimientos humanos y la estabilidad de los mismos son el motivo de estudio del control motor biomórfico y biomimético. Una manera práctica de entender el control motor humano es la simulación de esquemas de control biológicamente plausibles definidos a partir de los conocimientos de alto nivel adquiridos hasta el momento y su aplicación en hipótesis de trabajo específicas. En los sistemas biológicos, el aprendizaje motor se presenta como el aprendizaje de nuevas habilidades o la adaptación de las que ya son conocidas, con el fin de mantener el rendimiento y aprender qué movimientos hacer y cuándo hacerlos. El cerebelo parece jugar un papel clave en la modulación de movimientos precisos y coordinados. Por esta razón, la comprensión de los mecanismos de aprendizaje del cerebelo y su emulación a través de arquitecturas bio-inspiradas son dos procesos recíprocos de fundamental importancia para desarrollar sofisticados sistemas robóticos capaces de realizar operaciones complejas de forma precisa en escenarios no estructurados o previamente conocidos.

2.1 Control motor y aprendizaje motor

El objetivo principal de la investigación en robótica es crear máquinas que puedan funcionar de forma autónoma. Los primeros robots totalmente autónomos aparecieron en la segunda mitad del siglo XX y son generalmente empleados en trabajos sucios, pesados o peligrosos, que la gente normalmente hace en su ausencia. Sin embargo, los robots no pueden planificar y controlar las acciones físicas tan flexiblemente como lo hacen los seres humanos. Hoy en día, los investigadores estudian cómo las personas controlan sus cuerpos, cómo hacen los movimientos coordinados bajo condiciones variables del entorno, cómo sostienen un objeto o hacen varias tareas con el fin de dotar a los robots con dichas características humanas. Para ello, la robótica y la neurociencia aportan conocimiento de manera conjunta para el desarrollo de enfoques bio-inspirados o modelos de sistemas altamente no lineales. El objetivo es comprender los mecanismos de control motor y de aprendizaje motor del sistema nervioso central (CNS) a diferentes niveles, que van desde la célula hasta redes neuronales con el fin de crear movimientos coordinados y precisos y, al mismo tiempo, almacenar la información procesada.

El control automático, en particular la aplicación de la retroalimentación, ha sido fundamental en este sentido; el diseño de lugares de trabajo seguros con robots interactuando con humanos, depende de la estabilidad de los mismos y para ello es necesario implementar métodos más eficaces o mejores herramientas de robótica. El control motor humano se refiere a muchas actividades, como caminar, mirar (movimientos oculares), alcanzar objetos, dibujar y escribir, la producción del habla y la postura. En esta tesis, nos centraremos en los movimientos de un brazo robótico y abordaremos tanto los problemas de lograr un control *dócil* no lineal como la simulación de los procesos neuronales en un espacio de entrada multidimensional en tiempo real.

2.1.1 Percepción y control motor

Nuestro objetivo es desarrollar un sistema para mover un brazo robótico y corregir el error debido a las interacciones entre sus varios enlaces o a condiciones dinámicas o cinemáticas variables.

Una de las principales funciones del cerebro es extraer información a partir de la entrada sensorial, organizarla y utilizarla para responder a una situación particu-

lar. En este proceso, las complejas señales sensoriales motoras (registradas durante un movimiento o cualquier interacción con el medio ambiente) se organizan en modelos intrínsecos. Estos modelos intrínsecos adquiridos mediante la experiencia permiten movimientos eficientes y precisos. Los modelos se adquieren con una retroalimentación perceptiva que permite el aprendizaje motor para proceder tal y como se argumenta en la sección 1.1.2. Esto se logra a través de un entrenamiento repetitivo gracias al cual las habilidades motoras se mejoran en términos de precisión y suavidad de los movimientos. La retroalimentación indica hasta qué punto el movimiento es efectivo y los movimientos sucesivos son realizados para reducir el error entre lo deseado y la salida real. La estrategia de control empleada en el cerebelo que se aprovecha de la experiencia sensorial se denomina circuito cerrado o control de retroalimentación (por ejemplo, el nistagmo optocinético [2]). Un problema aún en investigación es cómo usar la retroalimentación de modo eficaz para el control motor. La elección de las ganancias es de particular interés en un sistema debido a su importante papel en la estabilidad y solidez del mismo.

En esta disertación un enfoque de aprendizaje del error de retroalimentación basado en iteraciones repetitivas de la misma tarea con ganancias muy bajas para obtener un control *dócil*.

2.1.2 Predicción y aprendizaje motor

El aprendizaje motor es la "adquisición de la información sobre los movimientos (y otras salidas motoras), incluyendo qué salida producir, así como cuándo y cómo producirla. El aprendizaje motor genera la formación de memorias motoras"[1] (Shadmehr and Wise [2]). El problema a resolver en el control de un sistema dinámico es encontrar la entrada al sistema que logre la conducta deseada como respuesta, incluso bajo perturbaciones o cambios en el entorno. El cerebelo actúa en este sentido, de hecho es capaz de adaptar su respuesta bajo cualquier condición capturando los modelos internos; es precisamente este mecanismo el que permite imitar las características de entrada-salida, o sus inversas, del aparato locomotor [3].

[1]Traducción de la cita literal en inglés: "acquisition of information about movements (and other motor outputs), including what output to produce as well as how and when to produce it. Motor learning results in the formation of motor memories" (Shadmehr and Wise [2]).

Teniendo en cuenta nuestro objetivo de hacer un control *dócil*, un desafío importante es la combinación del enfoque de aprendizaje del error de retroalimentación y del control adaptativo predictivo mediante la aplicación de un modelo de cerebelo capaz de adaptar sus correcciones, y asimismo almacenar las consecuencias sensoriales o las órdenes motoras hacia adelante para predecir acciones apropiadas cuando sea necesario.

2.2 El cerebelo como controlador adaptativo

Desde finales de los 60, Marr [4] (1969), Albus [5] (1971), Sejnowski [6] (1977), Fujita [7] (1982), y otros autores publicaron artículos hablando sobre el cerebelo como un controlador de filtro adaptativo. Muchos de los trabajos más recientes utilizan las redes neuronales artificiales como modelos para el cerebelo [8]. En el cómputo de un modelo interno, el CNS transforma una variable de entrada en una variable de salida codificando de este modo un mapa. En este proceso, el CNS ajusta u optimiza sus propios parámetros de forma automática. Sobre la base de las muestras codificadas en el mapa, el cerebelo, como controlador adaptativo, predice una señal futura y genera una señal para el control de un objeto (un brazo robótico) a lo largo de una trayectoria deseada. La principal ventaja de este enfoque es que permite el diseño de sistemas de control que pueden funcionar en un entorno desconocido o variable, cuando el modelo dinámico del robot es desconocido. En otras palabras, el control adaptativo permite la monitorización y la adaptación del comportamiento del controlador, obteniendo un mejor rendimiento y compensando los cambios en el entorno. Cuando el CNS experimenta nuevas dinámicas y el modelo interno no puede predecir las fuerzas necesarias o las consecuencias sensoriales, entonces éste tiene que ser entrenado de nuevo. Después de eso, se adapta y predice valores correctos en este nuevo escenario.

Los modelos internos dinámicos se crean dependiendo del contexto [9], ya que estos son lo que se han aprendido acerca de un control motor específico para una máquina específica. Además de la adaptación, otra cuestión es la retención de los modelos internos, es decir, la capacidad del CNS de recordar el modelo interno apropiado y usarlo para hacer predicciones para el movimiento. Por lo tanto, después del entrenamiento y adaptación, el modelo interno se codifica en una memoria a largo plazo. La evolución

de la memorización se llama consolidación. Un aspecto importante a resaltar acerca de la anatomía del cerebelo es que su estructura parece ser muy regular y modular (véase el capítulo 3 para más detalles). Por lo tanto, las funciones de las diferentes partes del cerebelo dependen de sus conexiones de entrada y de salida más que de su anatomía local.

En esta tesis, se tienen en cuenta todos estos conceptos para desarrollar un mecanismo de adaptación y de retención de los modelos internos dinámicos mediante un algoritmo de aprendizaje automático (LWPR) [10, 11] para la aproximación de funciones en tiempo de ejecución. Esto es especialmente adecuado en caso de ruido o datos parciales, para optimizar el espacio de entrada reduciendo el número de neuronas activas y generalizar los nuevos contextos. Las principales desventajas vienen dadas por el hecho de que se implementa el aprendizaje supervisado, lo que significa que necesitamos un muy buen entrenador y un gran número de parámetros a ajustar. Sin embargo, el LWPR permite un aprendizaje incremental en una plataforma robótica, ya que explota espacialmente modelos lineales localizados para aproximar funciones no lineales con un bajo coste computacional. Por lo tanto, la evaluación de la predicción es bastante rápida, lo que permite un aprendizaje en tiempo real. Además, el entrenamiento incremental permite la adquisición y retención de los diferentes modelos sin ninguna restricción de contextos y sin interferencias entre ellos. Asimismo, aprovechamos la capacidad del LWPR de transformar las entradas en una base óptima de señales claras y precisas para la capa de Purkinje de la pseudo arquitectura del cerebelo (véase capítulo 3 sección 3.2) para mejorar el aprendizaje del modelo interno.

2.2.1 Modelos de aprendizaje motor bio-inspirados

Como hemos comentado anteriormente, el microcircuito adaptativo del cerebelo se repite a lo largo de la corteza cerebelosa y se puede utilizar para corregir el comportamiento motor futuro a partir de los errores actuales. Para la compensación motora de la planta se han propuesto dos modelos alternativos para representar el papel del circuito mencionado anteriormente [12, 13, 14]. Estas son las arquitecturas hacia adelante y recurrente (figuras 3.2a y 3.2b en el capítulo 3). Mientras que en la primera arquitectura, el cerebelo añade pares de torsión correctivos adaptativos y sus pesos se ajustan en función de la señal de error motora, en la segunda la salida del cerebelo se utiliza

como una corrección sensorial al controlador y depende del error sensorial. Por lo tanto, la arquitectura hacia adelante necesita el error motor correcto que viene dado por la diferencia entre las órdenes motoras deseadas y las reales. Este no es un valor fácil de calcular, especialmente en una tarea motora real de sistemas biológicos complejos [8]. Este problema no existe en la arquitectura recurrente que parece ser biológicamente más plausible de acuerdo a las pruebas neurofisiológicas [8, 15].

En esta tesis, se presentan las dos arquitecturas bio-inspiradas en las que implementamos el enfoque de aprendizaje del error de retroalimentación al que se hace referencia en la sección 2.1.1 y los procesos neuronales mencionados en la sección 2.2; También se muestra cómo resolver el problema del error motor declarado por Porrill and Dean [8] (2007) para la compensación de las plantas de muchos grados de libertad (DOF) en la arquitectura del control hacia adelante.

2.3 Motivación de la tesis. Hipótesis de trabajo

El sistema nervioso ha desarrollado mecanismos de abstracción de modelos que permiten un control preciso sobre la base de los movimientos previamente experimentados (procesos de abstracción dinámicos y cinemáticos). El cerebelo es uno de los centros nerviosos más importantes que intervienen en las tareas de control preciso y coordinado. Parece un centro nervioso capaz de abstraer modelos de los objetos que están siendo manipulados. Es importante tener en cuenta que los sistemas biológicos logran un control preciso con actuadores de baja potencia, mientras que en la robótica, los robots normalmente logran una precisión muy alta y un movimiento de alta velocidad con fuerzas grandes y un alto consumo de energía. Esto permite el control de estrategias que son bastante independientes de los modelos dinámicos de la planta y el objeto que está siendo manipulado. Estos robots son potencialmente peligrosos en el campo de la interacción humana porque carecen de reacción sensorial en tiempo real y la fuerza de sus motores es peligrosa si el sistema se vuelve inestable. Por lo tanto, el campo de la robótica y la asistencia destinada a la interacción humana necesita el desarrollo de nuevas tecnologías y esquemas de control con actuadores de baja potencia. Se necesita adoptar estrategias de abstracción de modelos dinámicos y cinemáticos y esquemas de control adaptativos en tiempo real. Para ello, estudiamos los sistemas biológicos que

llevan a cabo la adaptación de los modelos dinámicos y cinemáticos para un control preciso y los enfoques de control automático para la abstracción de modelos. Hay dos cuestiones principales a considerar para el desarrollo de este estudio: los modelos internos y el control de retroalimentación.

Ito [16] (2008) examinó cómo el movimiento de una parte del cuerpo está controlado por el cerebro. Él dijo que un controlador de retroalimentación en la corteza motora genera un comando que mueve la parte del cuerpo (brazo robótico) de acuerdo a la instrucción deseada. Entonces, el modelo de la parte del cuerpo se almacena como un modelo interno en el cerebelo con el fin de realizar con precisión el control del brazo del robot al referirse a este. Cómo se aprende el modelo interno en el cerebelo? Una vez más, Ito [16] (2008) declaró que el cerebelo se compone de varios módulos llamados microcomplexes, siendo cada uno de ellos una máquina de aprendizaje unitaria (ULM), formada por una estructura de circuitos neuronales, que codifica un modelo interno. La relación entrada-salida de una ULM es modificada de modo adaptativo por las fibras trepadoras (CFs) que transmiten la señal de error. La neurona de salida de la corteza cerebelosa es la célula de Purkinje (PC), que integra la información transportada por una CF. Motivados por estos conceptos, en nuestro enfoque, nos aprovechamos del algoritmo de aproximación de función no lineal LWPR, para aprender de forma incremental, almacenar los modelos internos del brazo del robot y resolver la falta de un modelo de robot analítico perfecto. Además, Schweighofer et al. [17] (2001) trabajaron sobre una hipótesis en la que el aprendizaje del cerebelo se ve facilitado por un código disperso en las señales de las neuronas GC, es decir, un código neuronal en el que la proporción de neuronas activas es baja en cualquier momento. Por otra parte, Porrill and Dean [8] (2007) declararon que tanto la precisión como la velocidad de aprendizaje se pueden mejorar en gran medida mediante la optimización de la representación interna sensorial-motora a partir de una base óptima de los campos receptivos (RFs). Motivados por esto, hemos explotado las características del LWPR para implementar una especie de proceso adaptativo en la capa granular, con el fin de minimizar de manera eficiente los recursos necesarios para la asignación del mapeo del espacio de entrada de colocación y adaptar los núcleos para representar mejor el espacio de entrada con un número limitado de ellos. En nuestros modelos, los RF del LWPR se utilizan como una representación de las capas granular y molecular que entregan señales claras y precisas

a la capa de células de Purkinje.

En lo que respecta al control de retroalimentación, Kawato [18] (2009) afirmó que un mecanismo de aprendizaje del error de retroalimentación es necesario para calcular las órdenes necesarias a partir de un movimiento deseado. De hecho, un controlador de retroalimentación que genera buenas órdenes guía el aprendizaje supervisado en el cerebelo para la adquisición del modelo interno a través de las CF conjuntamente con las sinapsis PC-PF. El principal inconveniente del aprendizaje del error de retroalimentación Porrill and Dean [8] (2007) es que requiere estructuras complejas de referencia para sistemas genéricos redundantes y no lineales. Algunos autores trataron de evitar este problema utilizando altas ganancias en el lazo de retroalimentación [19]. Por otra parte, un cálculo analítico de las dinámicas es complejo y en el caso de un alto número de DOF, pueden desconocerse los parámetros precisos de la dinámica. En este caso, los modelos adaptativos son necesarios para un control preciso y estable durante la manipulación. Los métodos tradicionales ya no son adecuados para controlar las dinámicas complejas de la nueva generación de robots, ya que el movimiento se ve influido por las variables de estado de todas las articulaciones y el control se vuelve muy complejo y altamente no lineal [20]. Las no linealidades pueden dominar la dinámica del robot y las ganancias de realimentación tienen que ser aumentadas para compensar el error de seguimiento resultante [21] para seguir una trayectoria predefinida deseada. Esto es peligroso en relación con la estabilidad del sistema e implica movimientos no *dóciles*. Además, son inaceptables altas ganancias en los sistemas autónomos y biológicos ya que introducen elementos de desestabilización dados los retrasos sensorimotores de realimentación inherentes [8]. Adicionalmente, las altas ganancias generan grandes fuerzas, o en otras palabras, movimientos no *dóciles* potencialmente peligrosos [22], haciendo al robot menos seguro para el entorno, principalmente en el marco de las aplicaciones de interacción entre humanos y robot, y asimismo compromete la estabilidad en lazo cerrado [23]. En este trabajo, implementamos un controlador de aprendizaje de realimentación (LF) que genera comandos adaptativos a partir de los errores sensoriales, evitando de este modo un PID clásico con ganancias altas y complejas estructuras de referencia [8], solucionando por lo tanto el problema distal [1] [24].

[1] Traducción de "distal teacher problem" mencionado por Jordan and Rumelhart [24] (1992).

2.4 Objetivos científicos

Esta tesis se centra en el estudio del papel de los modelos internos del cerebelo en el control motor y el aprendizaje motor de los movimientos del brazo robótico. El objetivo principal es desarrollar esquemas bio-inspirados de control para sistemas complejos robóticos con actuadores de baja potencia que requieren modelos adaptativos para movimientos precisos durante la manipulación de objetos o la interacción con el mundo. Por lo tanto, esta tesis se centra específicamente en los siguientes objetivos:

- Estudio y desarrollo de sistemas de control de bajas ganancias.
- Implementación de esquemas de abstracción de modelos durante la manipulación de objetos. Extracción del conocimiento a través de la experimentación.
- Estudio de la organización de los modelos a partir de representaciones sensorimotoras.
- Esquemas bio-inspirados de control: integración de las metodologías de aprendizaje automático con sistemas de control biológicamente plausibles.

2.5 Marco del proyecto

El trabajo descrito en este documento ha sido desarrollado en el marco del proyecto europeo "SENSOrimotor structuring of Perception and Action for emergent Cognition" (SENSOPAC) [25].

El proyecto SENSOPAC (financiado por el 6 Programa Marco de la UE en su iniciativa de sistemas cognitivos) se extendió desde enero de 2006 a julio de 2010, en colaboración con 12 instituciones de 9 países diferentes. Dicho proyecto combinaba conceptos de inteligencia artificial y técnicas de modelado de sistemas biológicos para desarrollar un sistema capaz de abstraer nociones cognitivas de las relaciones sensorimotoras durante las interacciones con su entorno, y generalizar este conocimiento a situaciones nuevas. A través de sensores activos y acciones de exploración, el sistema descubre las relaciones sensorimotoras y, por consiguiente, aprende la estructura intrínseca de sus interacciones con el mundo y desentraña las relaciones causales y predictivas. El proyecto ha demostrado que un sistema robótico puede arrancar su desarrollo mediante procesos de generalización y el descubrimiento de abstracciones basadas en sensores,

mediante los resultados obtenidos de estudios de neurociencia sobre la representación y el procesamiento de datos sensoriales táctiles. El proyecto se basa en la sinergia entre varias instituciones científicas que son líderes en sus campos, agregando la interacción entre experimentadores de neurociencia, teóricos y expertos en robótica, dando lugar a un completo sistema cognitivo artificial que utiliza la actuación biomimética, y la detección bio-inspirada. El grupo de investigación de la Universidad de Granada ha participado activamente en el desarrollo del entorno de computación de neuronas de spikes (EDLUT) [26, 27] y en el desarrollo de sistemas de control informáticos y biológicos en tiempo real capaces de llevar a cabo tareas de manipulación. La figura 2.1 muestra la organización en módulos del proyecto SENSOPAC. La Universidad de Granada (y este trabajo como una parte) se centró en el cuarto módulo, a medio camino entre la neurofisiología (módulo 5) y los sistemas más abstractos y la robótica (módulos 1, 2, y 3).

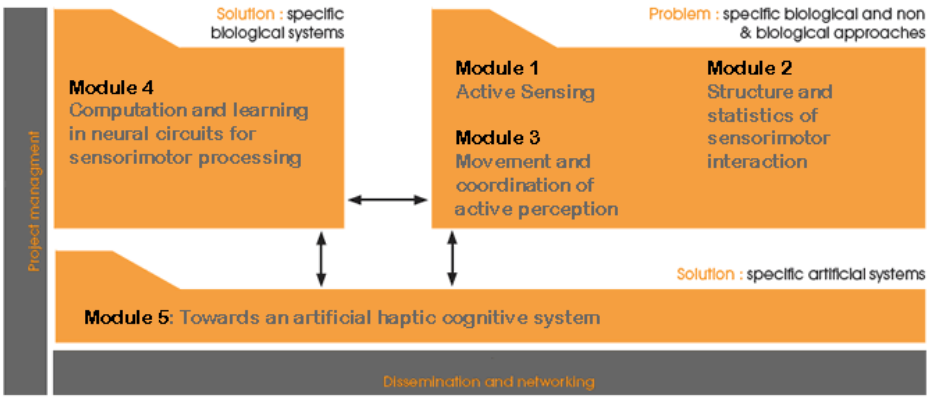


Figure 2.1: Estructura de módulos del proyecto europeo SENSOPAC - diagrama que muestra las principales tareas desarrolladas en el marco del proyecto. El grupo de investigación de la Universidad de Granada ha participado principalmente en el módulo 4.

Este trabajo es muy interdisciplinario, ya que combina los modelos biológicos con los sistemas de aprendizaje automático. Hay muchos esfuerzos que se basan en modelos bio-inspirados, pero en este proyecto adoptamos estrategias de otros campos como el aprendizaje automático y estudiamos su viabilidad y los beneficios en el contexto de los modelos biológicamente plausibles. Este proyecto propone el desarrollo de sistemas de control y la adquisición de los modelos basados en un motor de abstracción de los modelos y su validación a través de los robots biomórficos. Esto representa un avance

significativo en el campo de los agentes de control biomórficos con actuadores de baja potencia.

El algoritmo de aprendizaje automático utilizado entre los socios del proyecto Sensopac es el algoritmo LWPR desarrollado por Stefan Klanke, Sethu Vijayakumar y Stefan Schaal. Parte del trabajo desarrollado en esta tesis se hizo en la Universidad de Edimburgo, a través de un programa de intercambio de investigación que duró tres meses, con el fin de empezar a trabajar con el algoritmo LWPR y con la configuración del robot LWR de 7 grados de libertad diseñado por el Instituto de Robótica y Mecatrónica en el Centro Aeroespacial Alemán (DLR).

2.6 Organización de los capítulos

Los capítulos 4 y 5 describen los dos métodos desarrollados en este trabajo. Ambos incluyen una breve introducción y una sección de métodos y herramientas. A pesar de que son muy similares, los hemos mantenido en los dos capítulos separados para que sean autosuficientes (similares a los artículos científicos que describen estos dos hechos) El resto de esta tesis se organiza de la siguiente manera:

- **Capítulo 3**. Control adaptativo. Descripción del microcircuito del cerebelo y del papel de los modelos internos en el control motor; presentación de las arquitecturas de adaptación que emulan el papel del cerebelo en la adaptación motora.

- **Capítulo 4**. Acciones correctivas hacia adelante. Arquitectura adaptativa de aprendizaje del error de retroalimentación para el control motor. Este es un sistema de control basado en el aprendizaje del modelo interno inverso, en un mecanismo de aprendizaje del error de retroalimentación, en un algoritmo de aprendizaje automático y en un módulo bio-inspirado para controlar un brazo robótico. Finalmente se presentan los experimentos realizados y su correspondiente evaluación de resultados.

- **Capítulo 5**. Contribuciones correctivas retroalimentadas. Presentación de una arquitectura recurrente adaptativa basada en el aprendizaje del modelo interno directo en el que un controlador de error de retroalimentación lleva un control preciso, *dócil* y estable durante la manipulación de objetos. La sinergia de aprendizaje del cerebelo y el control automático hace que el robot sea adaptable a

entornos cambiantes. Finalmente se presentan los experimentos realizados y su correspondiente evaluación de resultados.

- **Capítulo 6**. Concluye con un resumen de los trabajos presentados en esta tesis y una descripción de los trabajos futuros.
- **Capítulo 7**. Esta es la traducción al español del capítulo conclusiones.

3

Adaptive Control

The cerebellum has cognitive functions, as well as motor and autonomic functions. It is a set of adaptive modules that contribute in the control of actions over time. Each microzone in which the cerebellar cortex is divided performs an algorithm based on an adaptive microcircuit. Adaptive means that the synaptic weights are adjusted by error signals. Furthermore, the cerebellar learning capability is allowed by internal models which are neural processes that mimic a behavior of input-output characteristics or their inverse of the motor apparatus. Internal models help the brain to perform a task precisely and increase the system's compliance. Inverse and forward models are used in different manners in the context of motor control as they have different input/output characteristics. From that, there are two main neuronal mechanisms in the brain based on feedforward and recurrent (or feedback) corrective actions respectively. In the first one, the cerebellum supplies corrections terms to the motor commands, while in the second it supplies corrections terms to the desired trajectory. After a brief description of the cerebellar microcircuitry and of the role of internal models in motor control, we present the adaptive architectures that emulate the role of the cerebellum in motor adaptation.

3.1 Introduction

Originally, feedback error learning was proposed to establish a computational model of the cerebellum for learning motor control with internal models in the CNS [28]. From a control viewpoint, the feedback error learning can be seen as an adaptive control technique [19]. In fact, "The goal of adaptive control is to achieve asymptotic tracking to the desired trajectory under the presence of unknown parameters in the plant dynamics by adjusting them during operation from input-output data while guaranteeing stability of the closed-loop system" (Nakanishi and Schaal [29]). In the brain, some of the neural connections have plasticity, which is thought to be the neural basis for adaptive behavior [28]. It is of interest to this work to study how the CNS is able to control the movements by making use of local synaptic plasticity. Marr [4] and Albus [5] proposed learning network models of the cerebellum, and Fujita [7] expanded the Marr-Albus model and proposed an adaptive filter model of the cerebellar cortex. The uniform structure of the neuronal circuitry that is repeated over the entire cerebellar cortex is presented in the following paragraph 3.2.

This work is focused on the investigation of the control of voluntary arm movements of many DOFs, which are characterized to have nonlinear dynamics. In the cerebellum, the controller is the motor cortex that is driven from higher motor centers through an instruction signal. In turn, the controlled object is the motor apparatus that is connected to lower motor centers driven by the motor cortex. In considering the control system structure for voluntary movements, Ito [30] depicted two different block diagrams of the connections between the cerebral cortex and the cerebellum which have both anatomical and physiological basis: the forward-model-based and the inverse-model-based control system schemes. A remarkable feature of these diagrams is that they make use of internal models, i.e. forward and inverse models. Internal models represent the cerebellar learning capability through which the cerebellum accurately performs a movement even in the absence of the visual feedback [30]. With the aim of acquiring internal models of the robot arm, the control system architectures we present in this thesis contain an internal feedback, which acts to apply feedforward adaptive corrections with the inverse model and feedback adaptive contributions with the forward model. In fact, the former plays the role of a feedforward controller that replaces the

motor cortex serving as a feedback controller [28, 30], while the latter fits the recurrent decorrelation control algorithm presented by Porrill et al. [14].

In this chapter, we focus on the description of the cerebellar neuronal microcircuitry and on its function on motor control and learning through internal models. Then, we present two models of the role of the cerebellum in motor adaptation: the feedforward and the recurrent architectures.

3.2 The Cerebellar Microcircuit

Since the 1960s, analysis on neuronal circuit structures of the cerebellum has revealed its involvement in the control of actions and in the acquisition of specific motor skills [4, 5, 31, 32]. In other words, the cerebellum plays an important role in accurate motor learning, motor adaptation, and cognition [31], e.g., computing the inverse dynamics of a body component [13, 33], delivering feedforward [12, 21] and feedback terms to the crude control commands from the motor cortex. Moreover, evidence has uncovered that the cerebellum also contributes to higher cognitive functions, but there is not a consensus about how it processes that activity [34].

Recent research works [15] describe the cerebellum as a set of adaptive modules, also called cerebellar microcomplexes, embedded in the motor control system to improve coordinated movements over time. There are even works that specifically study how spiking cerebellar-like neural structures can efficiently contribute in control tasks within biologically plausible control schemes [35, 36, 37].
The cerebellar cortex is divided into a large number of distinct microzones that correspond to the minimal functional unit. Throughout the cerebellum the internal microcircuitry of microzones is similar, there are three cortical layers in which the processing capability of the neural structure is distributed to perform an algorithm for a general signal-transforming activity as a complex adaptive filter [15]. This cerebellum microcircuit based on Marr and Albus' model [4, 5] consists of five main cell types localized between the granular, PC and molecular layers (see Fig. 3.1).
The output neuron of the cerebellar cortex is the PC, which integrates information carried by one climbing fiber (CF), that is the axon of the inferior olive neuron, and

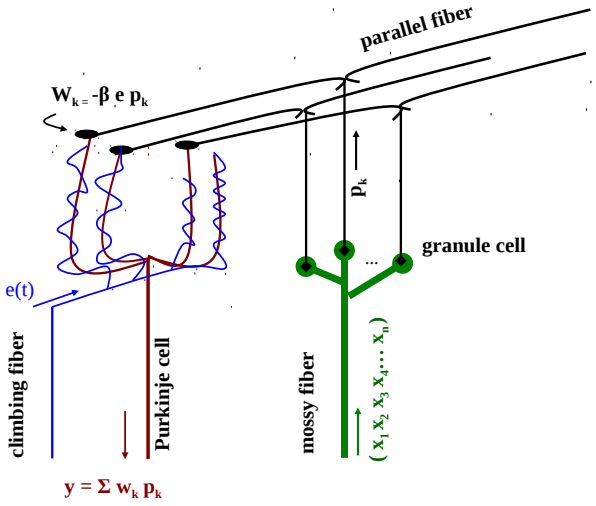


Figure 3.1: Basic structure of the cerebellar microcircuit.

receives inputs from many parallel fibers (PFs). Each single CF synapses a large number of dendrites of the target PC, thus affecting it significantly. The PC produces a *complex spike* [38] each time it receives input from a CF. PFs are the forks of the axons of the granule cells (GCs) situated in the granular layer, below the Purkinje layer. Besides CFs, the other main afferent of the cerebellar cortex is the mossy fiber (MF) that makes excitatory synaptic contacts mainly with GCs. The PCs in a microzone receive CFs driven by the same input and have a specific output effect [15], i.e. they control one specific movement component. Unlike the MF, CF is a rather unique afferent system of the cerebellum in the CNS. In fact, there is not a generally accepted view regarding its functions yet [39]. In the context of movement control, several studies [3, 16, 39, 40] revealed that the CF may signal the presence of an unexpected sensory stimulus during an action, so it may represent an error signal that helps to adjust the synaptic weights using the covariance learning rule proposed by Sejnowski [6].

3.3 Internal Models

Different research groups support that in the cerebellar cortex are allocated two different types of internal models [3, 13, 16, 30], the inverse and forward internal models, which are formed and adjusted through a supervised learning process as the movement is repeated [16] to mimic the behaviour of a natural process [41], to facilitate more precise

coordinated movements [15] and increase the system's control compliance [42]. Once the internal model is learned, it helps the brain to perform the task precisely without referring to feedback. The outcome is that the desired motions are predicted and only small correction forces are required, thus increasing the system's control compliance. The evidence for internal models of controlled objects in the brain is not yet confirmed, however it has successfully been applied to produce a robot that acquires a motor skill by learning [37, 43, 44, 45]. As a matter of fact, an increasing number of psychophysical experiments support the fact that humans make use of forward models [46, 47].
The inverse internal model reproduces the inverse dynamics model of a body part [16], in other words it produces the motor command in terms of torques to be applied. On the contrary, the second one reproduces the forward dynamics [16] and it plays an important role in controlling reaching movements especially in arm control [24, 48].

Learning an inverse dynamics model presents the following problems: the motor command error is not directly available to the CNS, and the movement errors need to be converted into motor errors in order to be used for the training [13, 15]. Previously, some authors such as Kawato [12], Gomi and Kawato [21] and Nakanishi and Schaal [29] proposed a cerebellar feedback error learning for adaptive control to solve this problem by a conventional feedback controller. In fact, the feedback controller transforms the trajectory error into motor command errors. This is then used as training signal and the feedback motor commands are summed with the feedforward commands given by the inverse model to act on the controlled arm.
Conventional feedback controllers are based on the principle of negative feedback, in other words they perform corrections on the basis of the error computed in the previous movement. The general feedback control law depends on the proportional, derivative and integral gain factors. In fact, they concern the stability of the system, and the compensation for errors. If the gains are high, the command torques will be strong making the robot stiff and dangerous, i.e., not compliant (not valid for human-robot interactions). However, small gains will reduce the stiffness and increase the robot's flexibility with respect to disturbances, but the movement may result imprecise. Then, the gains have to be tuned in order to achieve a balance between accuracy and stability. These are the main concerns to face up in order to obtain compliant and accurate motions through a combination of a reliable internal model that predicts correct command

torques and a feedback controller that leads both the motor control and the motor learning of the inverse internal model.

Since the actions based on sensory feedback are slow (due to sensory-motor pathways delays) and risky, the brain has developed mechanisms to estimate predictions [49]. This is achieved by forward models, which are employed by the sensorimotor system to provide predicted estimates of the state of the controlled object, and thus they overcome the significant delays of real sensory feedback [41, 49]. “In other words, by including the forward model within an internal negative feedback loop, it provides an internal feedback signal that is available much more rapidly than the actual feedback signals resulting from the movement” (Miall and Wolpert [41] (1996)). Other uses of the forward models have been proposed, such as anticipating and canceling sensory reafference of self initiated actions, providing an internal feedback to overcome time delays, and estimating the state of the controlled object some time into the future. Finally, during learning itself, forward models may be used to generate sensory error signals obtained by the difference between predicted feedback minus real feedback to guide the learning of inverse models. This can help to solve the "distal error" problem [24]. In fact, Jordan and Rumelhart [24] proposed the forward and inverse modeling approach to solve the problem of converting the error from task-oriented coordinates to the motor command space.

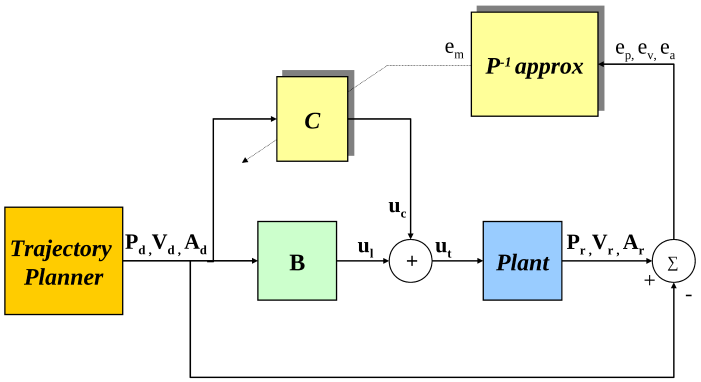
3.4 Adaptive Control Architectures

To form an adaptive control system, that is a system that can learn, internal models are essential. According to the control theory, in the brain, the motor cortex assumes the role of the controller for voluntary movements. It generates command signals to drive a controlled object represented by neurons in the temporoparietal cortex [16]. Lastly, the cerebellar hemispheres provide the internal models. The controller can be fed with the output of the controlled object (external feedback). The forward model can replace the external feedback, as it mimics the controlled object, while the inverse model replaces the controller as it provides the inverse dynamics of the robot arm without receiving feedback. Internal models are formed and updated by error signals carried by CFs. The forward model receives motor commands in MFs and sensory signals in CFs. In contrast,

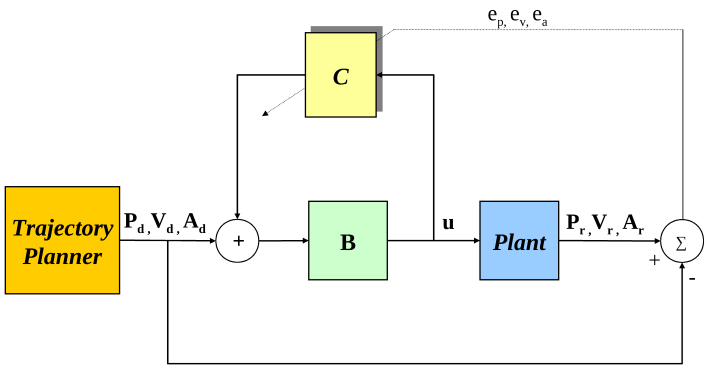
the inverse model receives instruction signals in MFs and motor signals in CFs [14, 30]. Thus, the output of the inverse model represents motor commands, while in the forward model, represents sensory signals. Motor errors can be derived from the controller, and the sensory errors from the output of the controlled object and the desired output. So, there are two main neuronal mechanisms in controlling and manipulating an object in the brain based on feedforward and recurrent corrective actions respectively. In the first the cerebellum supplies corrections to the motor commands, while in the second it supplies corrections to the desired trajectory.

Porrill et al. [14], [8] described and presented the main advantages and disadvantages of the forward loop and the recurrent architectures shown in figure 3.2. The cerebellar filter C is adaptive as it has adjustable weights. In the feedforward loop, the training signals are the errors obtained from the difference between the actual and the desired motor commands (motor error). Due to the fact that the desired motor commands are not directly observable, the motor error is not a desirable training signal since complex reference structures are required to compute it [14]. However, the recurrent architecture presents a solution to the motor error problem as it uses sensory errors instead of motor errors. Furthermore, this architecture seems to be more biologically plausible than the first architecture in the context of plant compensation [8].
In feedback error learning, Gomi and Kawato [21] presented a scheme in order to rescue the feedforward control system, in which they used the estimated motor error as both a training signal and a feedback error term. Anyway, as we have mentioned above, specially in autonomous and high nonlinear systems, the main difficulty is to balance the choice of the gains to avoid destabilizing effects. Despite this, feedback error learning still remains a useful approach under some circumstances. For example, Haith and Vijayakumar [50] highlighted the fact that the feedforward scheme is not affected by changes in the kinematics, but is impaired under changes in the dynamics in contrast with the recurrent architecture.

In the following chapters, we present the adaptive architectures we have implemented based on the internal-model control and on feedback error learning in order to avoid the difficulties related to the control of high nonlinear robotic system with many DOFs.



(a) The feedforward loop architecture.



(b) The recurrent architecture.

Figure 3.2: Schematic of the feedforward and recurrent architectures. In both architectures, the adaptive cerebellar weights are adjusted by an error-driven signal, the motor error in the feedforward scheme and the sensory error in the recurrent scheme. In Fig. 3.2a the correction term is provided by means of a corrective torque that is added to the torque computed by module B. While, in Fig. 3.2b the correction term is delivered in terms of spatial coordinates correcting the actual trajectory that is received by module B.

4

Feedforward Corrective Actions. Adaptive Feedback Error Learning Architecture for Motor Control.

Internal models form an adaptive control system since they are are formed by neural networks inside the brain that learn the input-output characteristics of some dynamic processes. Among them, in this chapter we study the inverse model and its role as a feedforward controller that replaces the feedback controller represented by the motor cortex in the brain. Then, we propose a control system based on inverse internal model for controlling a robot arm based on a feedback error learning mechanism (Learning feedback controller (LF)), a machine learning algorithm (LWPR) and a bio-inspired module (cerebellar-like engine (C)). The LF controller is driven by feedback errors and it generates motor commands to drive the supervised learning for acquisition of the inverse model through the CFs in conjunction with the PF - PC synapses. The LWPR engine incrementally acquires the inverse internal model of the robot arm, while the C module allows a faster control and a more precise movement thus compensating for interaction torques. We examine the contribution of the different components of the proposed scheme comparing the obtained performance with alternative approaches. Then, we show that the presented architecture can be used for accurate manipulation of different objects when their physical properties are not directly known by the controller. We evaluate how the scheme scales for plants of high DOFs (7-DOF robot arm).

4.1 Introduction

The problem of controlling a robot of many DOFs is that of determining the forces or torques to be developed by the joint actuators to obtain the right execution of the commanded task [51]. Traditional methods are no longer suitable for controlling the complex dynamics of the new generation of light-weight robots [52, 53] since the movement is influenced by the state variables of all the joints and the control becomes very complex and highly non-linear [20]. Nonlinearities can dominate the robot dynamics and the feedback gains have to be increased to compensate the resulting tracking error [21] for accurately following a predefined desired trajectory. This is dangerous regarding the system stability and implies non-compliant movements. Furthermore, high gains are unacceptable in autonomous and biological systems because they introduce destabilizing components provided the inherent feedback sensorimotor delays [8]. Therefore, classic feedback control seems to be inappropriate because high gains result in large forces generating potentially dangerous non-compliant movements [22], making the robot less safe for the environment, mainly in the framework of human-interaction applications, and compromising the closed-loop stability [23]. The major drawback of feedback error learning according to Porrill and Dean [8] (2007) is that it requires complex reference structures for generic redundant and nonlinear systems in order to compute an accurate motor error. Some authors attempted to avoid this problem using high gains in the feedback loop [19]. Otherwise, Gomi and Kawato [21] (1992) described a conventional feedback controller, Proportional Derivative and Acceleration (PDA), for a simple linear case, as an inverse reference model to convert the trajectory error into motor error.
Moreover, an analytical computation of the dynamics is complex and in the case of a large number of DOFs, precise dynamics parameters may be unknown. In this case, adaptive models are required for an accurate and stable control during manipulation. Here, we address these problems setting an appropriate inverse reference model in the feedback controller proposing a Learning Feedback (LF) controller, which improves the system behavior and self-adapts by a learning rule through consecutive iterations of the same trajectory.

Wolpert et al. [13] (1998) proposed a modular organized structure of internal models (forward and inverse models). The first type predicts the consequences of actions

under different contexts while the second one provides commands to achieve desired trajectories (see paragraph 3.3 in the Chapter 3). This chapter relates to the function of inverse internal models of memorizing a map of the motor apparatus in terms of an inverse dynamics model with a state-space representation. Accordingly, this inverse dynamics model will provide precise command torques over the input state-space and new trajectories are efficiently controlled based on previously learned primitives [3].

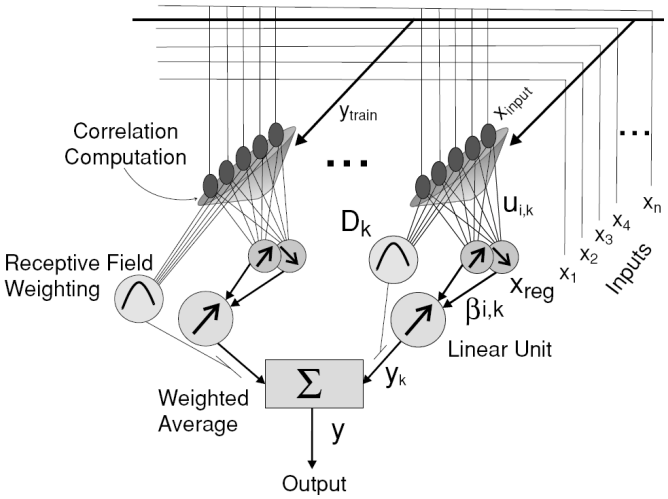
Recent approaches treat the solution of inverse dynamics (also referred to as inverse internal model) as a function approximation problem [22, 45]. A robot arm produces a vast amount of data (joint angles, velocities, accelerations, and torques) during its movements, which can be used as training data for the LWPR algorithm [10, 11]. In this sense, machine learning algorithms and the robot systems may help us to develop an understanding of how motor learning takes place in biological systems. Human motor skills are adaptive to changes in the body's physical morphology and the nature of the task being performed, the biological system is able to accomplish compliant and precise motion. Function approximation can solve the lack of a perfect analytical model; however, the learned dynamics function represents only a part of the dynamic robot model. In fact, there are some difficulties to know the exact model in the case of miscalibrations of the joints, changes of context, objects under manipulation, etc; thus, the learned function will not be able to prevent all the uncertainties. For this reason, the task of the cerebellar microcircuit is to compensate for these changes while the LWPR incrementally learns the adapted dynamic models. The learned function which emulates the inverse dynamics model of the arm together with the feedback learning module make the robot arm capable of performing movements that are both precise and compliant at the same time and also adaptable to changing situations. In fact, it is well-known that the human motor system is able to generate accurate control commands under different environment changing conditions.

Ito [16] (2008) stated that a feedback controller generates a command in the motor cortex that drives the controlled body part accordingly to the desired instruction. Then, the body inherent characteristics are captured in an internal model in the cerebellum in order to precisely perform the control of the robot arm by referring to it. That means that the feedback control is replaced by the forward control that reproduces the

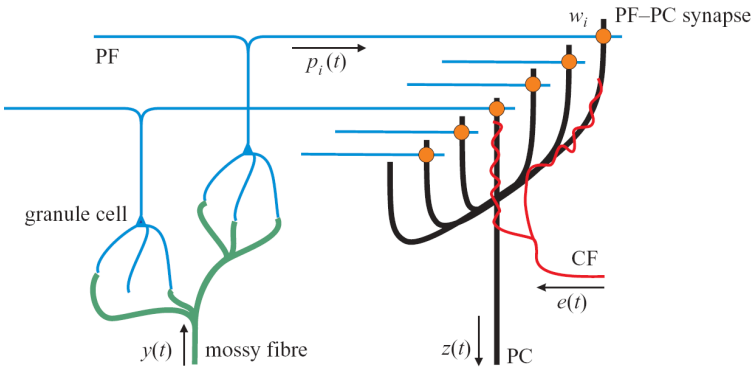
dynamics of the robot arm.
In this chapter, we present an architecture in which the cerebellar cortex is embedded in a feedforward loop and the basic cerebellar microcircuit is based on the Marr and Albus' model [4, 5]. The LF controller generates adaptive feedback commands from the sensory errors avoiding classic PID with high gains and complex reference structures [8] thus solving the motor error problem [24], and the LWPR incrementally learns the internal inverse model of the robot arm.

We have done a generalization experiment in order to evaluate the functional structure of our LWPR internal model. After sufficient learning of trajectories defined in Equations (4.16) and (4.17), LWPR predicts accurate torques when the robot arm has to follow a trajectory given by the summation of previous learned trajectories keeping the desired performance. In fact, the motor commands are predicted by a map from the state-space input composed of positions, velocities and accelerations of desired trajectory, and positions and velocities of the computed trajectory. So, the more the new trajectory belongs to the known state-space the generalization of previous learning will work better. In practice, LWPR generalization performance (that supports our approach's generalization capability) has already been evaluated in Schaal et al. [54] (2002).

Among the global nonlinear function approximators, such as Gaussian Process Regression (GPR) [55], or Support Vector Regression (SVR) [56], LWPR has been successfully used for online incremental learning in robotic platforms [11, 22, 54, 57] since it spatially exploits localized linear models to approximate nonlinear functions at a low computational cost. Therefore, the evaluation of the prediction value is quite fast allowing real-time learning. Besides, the incremental training permits the acquisition and retention of different tasks without restriction in the number of tasks and without interferences among them [45]. Figure 4.1a summarizes the basic process of LWPR learning. Bearing in mind Fig. 4.1, we exploit the similarity between the LWPR learning mechanism and the cerebellar circuitry to fuse their functionalities and take advantage both of the potentiality of the machine learning algorithm and of the cerebellum's role to make fine adjustments to the way an action is performed.



(a) The LWPR processing unit
(Reproduced from Vijayakumar et al. [11] (2005)).



(b) The microcircuit of the cerebellum
(Reproduced from Porrill et al. [14] (2004)).

Figure 4.1: Parallelism between the LWPR processing unit and the cerebellum microcircuit.

In the LWPR, the RF weighting kernels encode the input space like the cerebellar GCs expansively encode the information coming from the MFs. Previous simulation studies have widely developed the theory of the cerebellar granular layer as a liquid state machine, where the PFs generate a finite but very long sequence of active neuron populations without recurrence [58]. In a similar way, RF weighting kernels could adapt their weights to select different outputs depending on the current state of the robot arm.

Schweighofer et al. [17] (2001) hypothesized that the cerebellar learning is facilitated by a GC sparse code, i.e. a neural code in which the ratio of active neurons is low at any time. Porrill and Dean [8] (2007) stated that both accuracy and learning speed could greatly improve by optimizing the choice of the centers and transforming to an optimal basis of RFs. According to these hypotheses, we exploited the LWPR capabilities to emulate the granular layer with a limited number of resources; LWPR places and adapts efficiently its internal kernels to better represent the input-space with a limited number of them. In fact, unlike the cerebellum, LWPR automatically evaluates the required number of local correlation modules to optimize the network size by incremental learning. In this sense, each LWPR module and its associated RF weights can be seen as providing the firing rate of a PF, while the set of active RF weights can be seen as the current state of the granular layer processing module [58]. This state would be propagated through PFs and interneurons to produce more accurate signals at the PCs [59]. The major strength of the LWPR is the use of incremental calculation methods during the training and therefore, it does not require the input data to be stored. Furthermore, the algorithm can cope with highly redundant and irrelevant data input dimensions without any prior knowledge of the data, because it uses an incremental version of the Partial Least Squares regression (PLS).

It is important to remark that recent studies highlight the importance of other elements such as interneurons for learning consolidation [59]. As input to the Purkinje layer we use machine learning modules (LWPR kernels) whose adaptive input RFs can be seen as an abstraction of the granular and molecular layer modules (including also interneurons [59]) that efficiently and accurately deliver clean signals to the Purkinje layer. In our approach the cerebellar module (C) includes only short term adaptation while consolidation of learned primitives takes place at the machine learning module (LWPR) as also indicated in the results and conclusions sections. There have been

some applications of cerebellar models to the control of robot manipulators, all in simple systems such as Gomi and Kawato [21] (1992), Porrill and Dean [8] (2007), Haith and Vijayakumar [50] (2009) and in real robot systems such as Shibata and Schaal [60] (2001).

The major contribution of the presented model is that it manages to learn different non-linear dynamics with a hybrid approach that uses a machine learning engine (LWPR) and a bio-inspired module (cerebellar-like network). In other words, we exploit the RF weighting of each local model in the LWPR as granular and molecular layer microzones (complexes) in the cerebellum approach. Therefore, the cerebellum module instead of receiving inputs by means of MFs, receives pre-processed signals from the LWPR RFs. This takes advantage of the optimized engine for a compact sensorimotor representation provided by the LWPR. The LWPR incrementally learns and stores the inverse internal model of the robot arm, while the C module allows a faster control and a more precise movement [17]. As Ito [16] (2008) stated, the cerebellum is composed of many modules called microcomplexes, each of which is a ULM made up of a structured neuronal circuits and it encodes an internal model. The input-output relationship of each ULM is adaptively modified by the CFs that convey the error signal. So, the dynamics of the robot are encoded in the ULM which carries out the role of an internal model, as illustrated in Fig. 4.3. Each microcomplex adapts the corrections for any possible miscalibration, e.g. on account of interaction torques, by a teaching inverse reference signal or appropriate motor command. The latter is learned by the LF controller on consecutive iterations of the task and also assures the stability of the system without using high gains. Once the internal model is learned, the ULM performs the movement precisely with a low contribution from the feedback.

In the following paragraphs, we will present the advantages of the control architecture system according to the next structure. Firstly, each block of the proposed architecture is presented with high regard to the learning rule of the LF controller and the connection between the cerebellum and the LWPR algorithm. Secondly, we will demonstrate the validity and efficiency of the model with experiments on a 3-DOF and 7-DOF Light Weight Robot (LWR) arm (see Fig. 4.2). This is the third generation of a 7-DOF robot arm designed by the Institute of Robotics and Mechatronics at the

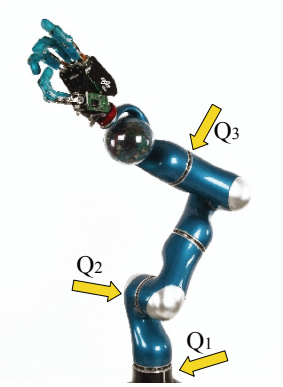


Figure 4.2: Light Weight Robot (LWR) arm and hand consisting of seven revolute joints. The three joints used in our 3DOF experiments are explicitly indicated. Figure adapted from Albu-Schaffer et al. [1] (2007).

German Aerospace Centre (DLR) [52, 53].

4.2 Control Architecture

In this section, the Adaptive Feedback Error Learning (AFEL) architecture, shown in Fig. 4.3, is presented. It consists of the LF controller which generates the u_{fb} feedback joint torques and the ULM which provides the u_{ff} feed-forward joint torques. This u_{ff} term is a combination of an u_l prediction from the LWPR and an u_c prediction from the cerebellum model. The trajectory planner block computes the desired joint angles, velocities, and accelerations (Q_d, $\dot{Q}_d$, $\ddot{Q}_d$) by inverse kinematics.

The field of nonlinear control theory is very large, therefore, we will focus our attention on a particular method called feedforward nonlinear control [61]. Considering the analytical robot model, we calculate the joint torques required for a particular trajectory using the following dynamic Equation (4.1) of the robot:

$$\tau = M(Q)\ddot{Q} + V(Q,\dot{Q}) + G(Q) + F(Q,\dot{Q}), \tag{4.1}$$

where $M(Q)$ is the inertia matrix of the manipulator, $V(Q,\dot{Q})$ represents the centrifugal and Coriolis terms, $G(Q)$ is the gravity term, $F(Q,\dot{Q})$ is the model of friction, and Q, $\dot{Q}$, $\ddot{Q}$ are the joint angles, velocities, and accelerations of the robot arm.

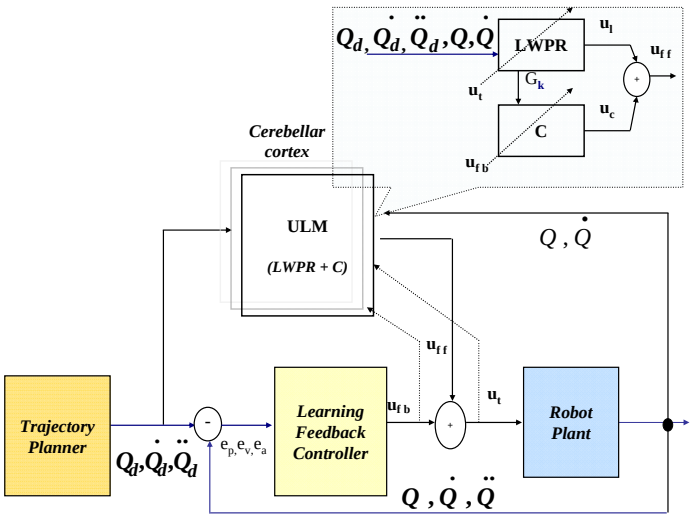


Figure 4.3: Block diagram for the AFEL scheme.

In the architecture, the u_t global torque (4.2) is the summation of the u_{ff} predicted motor command which comes out of the ULM and allows the robot to follow a desired trajectory (Q_d, $\dot{Q}_d$, $\ddot{Q}_d$), and the u_{fb} motor feedback command, generated by the LF controller, which ensures the stability of the trajectory.

$$u_t = u_{ff} + u_{fb}. \tag{4.2}$$

If the adaptive model is accurate, the resulting u_{ff} feed-forward term will cancel the robot nonlinearities. However, if the inverse dynamic model is not exact, there will be an error between the desired signal and the output of the controlled robot arm (Q, $\dot{Q}$, $\ddot{Q}$). This is also called feedback error and the LF controller output will reflect it. Then, the cerebellum receives the signal which will activate the process of learning [16]. Inside the ULM block, the LWPR algorithm plays the important role of internal model, or in other words, it learns the inverse model of the robot arm while corrections are applied following the trajectory. In addition to this, the ULM also consists of a set of uniform cerebellar circuits which are capable of learning the input-output relationship of dynamic processes by means of the long-term depression induced in the synapses between PFs and PCs. The error signal conveyed by the CF adaptively modifies this relationship. Analogously, this plastic site is represented by the C module - in function

rather than form - in our system, and it is quite sensitive to the representation of the input space [8, 17]. The error signal computed by the LF and sent to the C module is the effect of the system action that is minimized through the Hebbian rule (4.13) by analogy with the *distal error* problem formulated by Jordan and Rumelhart [24] (1992). Further details about the ULM are given in Section 4.2.2.

Summarizing, the cerebellum leads the model abstraction engine (LWPR), captures through optimized representation the sensorimotor complexes and produces u_c corresponding torques to reduce the u_{fb} teaching signal to the least possible amount (error related estimate). The LWPR engine incrementally learns from the u_t global torques incrementally by abstracting the whole model.

4.2.1 Learning Feedback Controller

The LF controller overcomes the lack of a precise robot arm dynamic model, ensures the stability of the system, and enables the control architecture to get a better performance. This is achieved by adding a feedback control torque to the one which is provided by the known part of the model. The $\hat{u}_{fb_r}$ feedback torque, shown in the following Equation (4.3), is adjusted through a learning rule after consecutive repetitions of the same task, $r = 0, 1, ...$ The dynamics of the robot can be written as:

$$\tau = \hat{M}(Q_d)\ddot{Q}_d + \hat{V}(Q_d, \dot{Q}_d) + \hat{G}(Q_d) + \hat{u}_{fb_r}. \tag{4.3}$$

Keeping in mind the dynamic model described in Equation (4.1), considering a non-modeled $\hat{u}_{fb_r}$ friction term to be added to the estimated terms, and substituting Equation (4.3), we obtain error Equations (4.4) and (4.5) of the closed loop of the control system in Fig. 4.3:

$$\ddot{e} = M^{-1}(F - \hat{u}_{fb_r}) \tag{4.4}$$

or in a more compact form:

$$\ddot{e} = (B - B_r), \tag{4.5}$$

where $\ddot{e} = \ddot{Q}_d - \ddot{Q}$, $B = M^{-1}F$, $B_r = M^{-1}\hat{u}_{fb_r}$. For every joint, the Equation (4.5) becomes:

$$\ddot{e}_i = (B_i - B_{ir}). \tag{4.6}$$

In the last Expression (4.6), term B_i is constant during iterations over time, while term B_{ir} changes on consecutive iterations of the task. We propose the following learning rule for each i joint, as indicated in Expression (4.7):

$$\hat{B}_{i(r+1)} = \hat{B}_{ir} + P * e_{ir}, \tag{4.7}$$

where $P * e_{ir}$ is the convolution between the impulse response filter P and the error in iteration r. Among filters, we chose the one given by Equation (4.8):

$$P(s) = s^2 + (K_{vi} - \mu)s + (K_{pi} - \mu), \tag{4.8}$$

where μ is a constant.
P is a non-causal filter, so it uses the errors of the previous iterations and its convergence depends on μ. Further specifics on the LF controller analysis are provided in Appendix A.

4.2.2 Unit Learning Machine

LWPR algorithm creates N linear local models and feeds the inputs into them. Here, a weighting kernel computes a weight $p(k, i)$ for each x_i data point according to the distance from the c_k center of the kernel in each k local unit. The weight is a measure of how often an item of x_i data falls into the region of validity of each linear model. The kernel function is defined as a Gaussian kernel:

$$p_k = exp\left(-\frac{1}{2}(x_i - c_k)^T D_k (x_i - c_k)\right), \tag{4.9}$$

where D_k is a positive definite matrix which is called distance matrix.

This measure is updated iteratively (on-line) by using an incremental gradient descent based on stochastic leave-one-out cross validation criterion. This is very important in the performance of our system since the remaining sites of plasticity strongly depend on a good lower-dimensional manifold of the input space. A good representation of the input space as the one that could be achieved by means of Equation (4.9) leads to sparse code in the GC layer in analogy with the Marr and Albus theory.

Here, we assume that there are N local linear models combining their individual

prediction $\overline{y}_k$ to make the global output $\hat{y}$ (4.10). In other words, the total output of the network is the weighted mean of all linear models:

$$\hat{y} = \frac{\sum_{k=1}^{N} p_k \overline{y}_k}{\sum_{k=1}^{N} p_k}. \tag{4.10}$$

Regarding the learning process, the number of local models increases with the complexity of the input space. If a data sample falls into the validity region of a model, its own distance matrix and regression parameters will be updated; furthermore, the update of each local model is independent from all the other models. As mentioned before, the inverse model is trained using a feedback error learning strategy. The LF controller converts trajectory errors into motor commands to be used as a training signal for the cerebellar network.

Comparing the LWPR processing unit and the cerebellar microcircuit shown in Fig. 4.1, we take advantage of the LWPR kernels as granular and molecular layer microzones (complexes) in the cerebellum approach. Therefore, the cerebellum module, instead of receiving inputs by means of MFs, receives preprocessed signals from the LWPR RF filters. This benefits from the optimized engine for a compact sensorimotor representation provided by the LWPR. The LWPR sends the inputs $(x_1, x_2, ..., x_i)$ to each local model to produce the $\overline{y}_k$ signals. Likewise, in the cerebellum, the inputs enter through every MF which sends them to a bank of filters G_k (GCs) to produce the $p_k(t)$ signals, defined in (4.11), which are driven by the kth PFs and the interneuron contributions. Then, the specific PF and interneuron pathway to the Purkinje layer carry the $p_k(t)$ signal to the PC synapse:

$$p_k = G_k(x_1, x_2, ..., x_i), k = 1..N, i = 1..M. \tag{4.11}$$

PC output $z(t)$, defined in (4.12), is modeled as a weighted linear combination of the $p_k(t)$

$$z(t) = \sum_k w_k p_k(t). \tag{4.12}$$

The synaptic weights w_k of the kth PF-PC synapse (see Fig. 4.1b) are updated using the heterosynaptic covariance learning rule (4.13) [6] in the continuous form [8], and adjusted by an e_t teaching or error signal (the CF input). Because of the adaptation

of the synaptic weights, Fujita [7] (1982) introduced the concept of adaptive filter to cerebellar modeling.

$$\delta w_k = -\beta e(t) p_k(t), \tag{4.13}$$

where β is a small positive learning rate and $e(t)$ is the error signal carried out by the CF. In this approach, $e(t)$ is the feedback error torque $\hat{u}_{fb_r}$. β is 0.05 in all the experiments carried out.

In order to perform an optimal function approximation, the LWPR incrementally divides the input space into a set of RFs defined by the center c_k and a Gaussian area characterized by the particular kernel width D_k, as shown in Equation (4.9). During each iteration, all RFs calculate their weight activation in order to assign the new input, x_i, to the closest RF and consequently, the center and the kernel width are incrementally updated. The optimized choice of centers and widths gives the optimal basis of RFs, so that the accuracy and the learning speed of the ULM are improved. In other words, Equation (4.11) represents the bank of G_k filters for the GCs in the cerebellum and their response is both used to compute the cerebellar output $z(t) = u_c$, as defined in Equation (4.12) and to update the synaptic weights (4.13).

From Fig. 4.3, we see that the internal model within the cerebellar cortex will retain the global control torque as target signal u_t. Given that, Equation (4.3) can now be completed:

$$u_t = M(Q_d)\ddot{Q}_d + V(Q_d, \dot{Q}_d) + G(Q_d) + \hat{u}_{fb} + \hat{u}_c. \tag{4.14}$$

The function defined in Equation (4.15) represents the nonlinear function to be approximated by means of linear regressions and it depends on the desired angular position, velocity and acceleration, and on the real angular position and velocity of the joints of the arm.

$$u_l = \Phi(Q_d, \dot{Q}_d, \ddot{Q}_d, Q, \dot{Q}), \tag{4.15}$$

where function Φ can be learned online and offline. Further details about the method of learning are given in Subsection 4.3.1.

4.3 Simulation Results

We have verified the performance of the AFEL architecture in adapting to dynamic and kinematic changes of the controlled object on two physically realistic models of robotic arms. In the first setup, the LWR arm was simulated considering a reduced configuration to 3 DOFs in order to get fewer input dimensions to the machine learning engine. Specifically, the first (we will refer to it as Q_1), second (Q_2), and fifth joint (Q_3) have been used and the others have been kept fixed. The three non-fixed joints used in our experiments are indicated in Fig. 4.2. This reduces the amount of training data required, and expedites the initial learning process. Afterwards, all 7 DOFs of the LWR III shown in Fig. 4.2 were involved in the simulation, as described in 4.3.3. To simulate dynamic changes, we considered that the manipulated object was the last link of the arm, so we changed the physical properties of the tip of the arm when emulating manipulation of different objects. Furthermore, to simulate a kinematic modification, we changed and fixed a certain orientation shift of the end-effector. Simulations were setup in the Matlab robotics toolbox [62]. The task for the experiments with the LWR arm was to follow a planned trajectory in a 3-dimensional task space.

4.3.1 Control Performance Evaluation

The robot end-effector traced out a target trajectory shown in Fig. 4.4b, defined by (4.16):

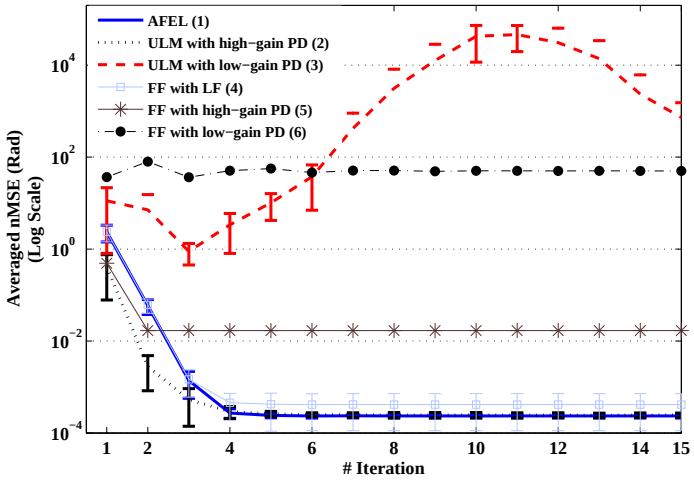
$$\begin{aligned} Q_1 &= Dsin(2\pi t), \\ Q_2 &= Dsin\left(2\pi t + \frac{\pi}{4}\right), \\ Q_3 &= Dsin\left(2\pi t + \frac{\pi}{2}\right), \end{aligned} \tag{4.16}$$

where D is a constant, and Q_1, Q_2, and Q_3 are the joint coordinates, respectively.

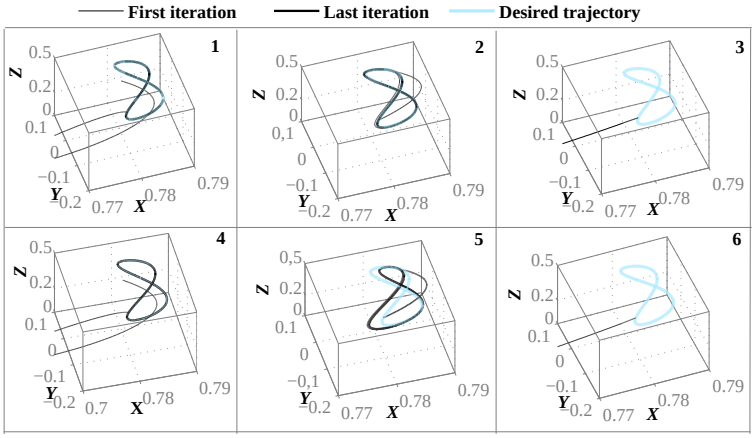
To approximate the nonlinear function described in Equation (4.14), a sequence of 16 eight-like shape movements was simulated to collect enough target points (8000) for training. Next, 15 iterations of the trajectory were repeated using the learned inverse dynamic model. An analytical model of the 3-DOF LWR arm (4.1) generated the feed-forward joint data torques given the desired joint angles, velocities, and accelerations. In the training stage, the LWPR algorithm approximated Equation (4.14)

which contains the terms of the u_{fb} feedback joint torques and the u_c cerebellar joint torques besides the feedforward joint torques supplied by the analytical model. Then, the learned inverse dynamic model defined in Equation (4.14) is tested: the analytical model is no longer used, the LWPR module predicted the joint torques to be applied to the robot plant, the cerebellum still optimized the execution of the trajectory and the LF supervised the system, meanwhile learning of the u_t global torques was proceeding. The LWPR training took place for each DOF separately (a LWPR module for each i joint) with a training and a test set of [5 x number of joints] inputs (desired joint angles, velocities, and accelerations (Q_d, $\dot{Q}_d$, $\ddot{Q}_d$) and the current joint angles, velocities (Q, $\dot{Q}$)), and 1 target (joint torque u_{ti} of joint i) (according to Equation (4.15)). As the LWPR has learned the inverse dynamics model, the movement is performed more precisely with a lower contribution command from the feedback and from the cerebellar circuitry. Therefore LWPR works also as a memory consolidation module.

To evaluate the AFEL architecture performance, we examined how the tracking errors became compensated following the desired trajectory (4.16). In order to highlight the advantages of this novel adaptive control system, we set up six different architectures by substituting one block of the AFEL scheme for other alternatives, as shown in Fig. 4.4a. The thicker solid line in Fig. 4.4a is referred to the novel AFEL architecture performance (case 1). The LF controller was replaced with a high-gain PD (case 2) and with a low-gain PD (case 3) while the ULM module was substituted for an analytical dynamics method called Feed-Forward (FF) module (case 4). Lastly, both the LF controller and the adaptive ULM module were substituted for a high-gain PD (case 5) and for a low-gain PD (case 6).



(a)



(b)

Figure 4.4: Control architecture tracking performances manipulating a 6 kg load at the last joint. Figure 4.4a displays the nMSE averaged over 3 joints for six different architectures described in Subsection 4.3.1. Error bars represent the standard deviation of the mean of the nMSE of the three joints. As a result of the simulation, Fig. 4.4b shows the eight-like figure-shape (i.e. the desired and actual trajectories before and after learning, indicated as first iteration and last iteration relative to the left panel learning process) obtained after 15 trials for the six case-studies (linked with the numbers on the top-right of the panels, referred to the control architectures indicated in Fig. 4.4a) in the task space. The low-gain PD controller (cases 3 and 6) yields a very large tracking error (therefore, the actual initial and final trajectories lay out of the plot).

The used system accuracy measure is the normalized mean squared error (nMSE) between the desired joint angle (Rad) and the actual joint angle (Rad) obtained from the robot plant. The nMSE is defined as the MSE divided by the variance of the target data values. From Fig. 4.4a, we can see that the proposed AFEL architecture (case 1) achieves a very good performance with a low standard deviation. In order to guarantee a low tracking error in the system with a PD controller instead of the LF controller (cases 2, and 5), the PD gains have to be set to high values, which results in a potentially dangerous non-compliant movement because the manipulator could damage the environment if it comes into contact with it. As a result of this, the maximum torque (Nm) applied by joint actuators is too high. Table 4.1 reflects this effect in terms of maximum torque u_t applied at each joint during a single trajectory.

	Maximum absolute Torques (Nm)			**Minimum absolute Torques (Nm)**			**RMS (Nm)**		
	Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3
AFEL	88	206	106	0	0	0	52	112	56
ULM with high-gain PD	620	1037	908	0	0	0	56	119	60
ULM with low-gain PD	501	812	689	0	0	0	56	117	60
FF with LF	116	213	130	0	0	0	52	112	56
FF with high-gain PD	477	755	642	0	0	0	56	115	59
FF with low-gain PD	74	177	59	0	0	0	42	92	31

Table 4.1: The values in the first two columns are the maximum and minimum absolute torques applied at joints in adapting to the different dynamics models for the six different architectures. The third column contains the quadratic mean or root mean square (RMS) of torques applied during all the iterations of the executed eight-like trajectory.

As a matter of fact, for ULM with high-gain PD architecture (case 2), the maximum torque gets up to 1000 Nm among the 3 joints, while for the AFEL architecture, the maximum torque was limited to around 200 Nm. To achieve the same rate of performance as the AFEL system (see Fig. 4.4), gains were multiplied by a factor of 250. Finally, by substituting the ULM module with an analytical model (case 4), the system still achieves good performance, but the standard deviation is higher, which means that the error is not equally diminished for all the joints at the same time. In other words, the cerebellum does optimize the miscalibration and gets adapted to novel dynamics. During the experiment, we set the LF controller gains to very low values for a compliant control observing the sufficient condition provided by Nakanishi and Schaal [29] (2004) in order to ensure stability of the FEL scheme into account.

Removing the cerebellar circuitry from the ULM module (LWPR alone), we obtain the result shown in Fig. 4.5, which is compared with the performance (dashed line) of the AFEL architecture. In both cases, the arm manipulates a 6 kg load at the last joint. This is the demonstration of how the cerebellum makes the LWPR learn an optimized inverse dynamics model of the robot arm and makes fine adjustments to the way the trajectory is performed. In fact, the nMSE is similar for each joint, as indicated by the error bars.

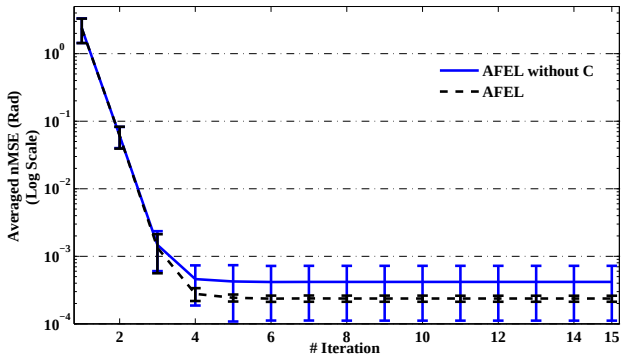


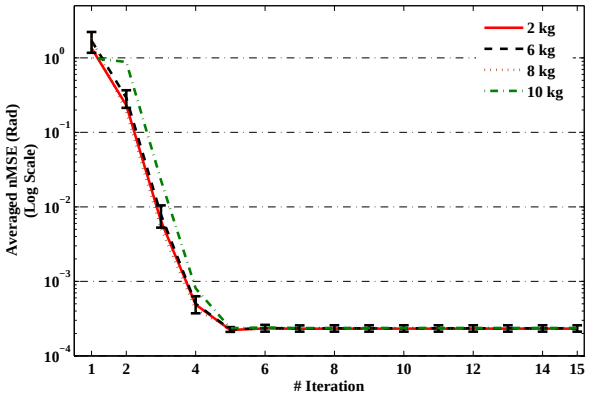
Figure 4.5: The dashed line represents the nMSE averaged over 3 joints related to the proposed AFEL architecture. The solid line shows the tracking error performance obtained by removing the cerebellar structure from the ULM module in the AFEL scheme. Comparing them, we make clear that the cerebellum drives the model abstraction engine (LWPR). In this way, the LWPR incrementally abstracts the whole model. Error bars represent the standard deviation above and below the mean of the nMSE of the three joints.

4.3.2 Dynamics and Kinematics Changes

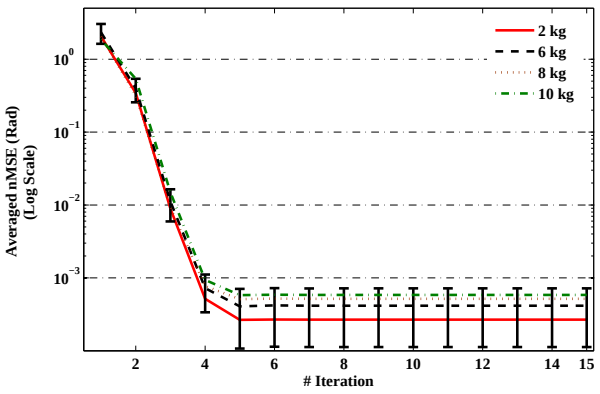
The dynamics of the robot arm changes as the robot manipulates different objects or different contexts. In this section, four contexts are simulated by attaching objects with different masses at the last link of the arm. The masses are 2, 6, 8, and 10 kg, respectively. 15 iterations of the trajectory were executed using the inverse dynamics model of the four arm+object instantiation previously learned by the LWPR. We ran the experiment ten times with different initial positions around the trajectory, defined in Equation (4.16), on each trial. We computed the robot arm tip position error in the different trials and averaged it over ten times.

With the LWR 3-DOF arm the gains of the LF controller have been set to $K_p = 12.5$;

$K_v = 5$; $mu = 1.5$ for the four objects and for all the robot joints. However, with the LWR 7-DOF arm the gains of the LF controller have been set to $K_p = 6$; $K_v = 3$; $mu = 0.75$. Nakanishi and Schaal [29] (2004) provided a strictly positive real (SPR) condition, that is $K_v^2 > K_p$, for choosing feedback gains in order to ensure stability of the feedback error learning mechanism. The choice of feedback gains we have made satisfies the mentioned condition, which implies that the stability of the AFEL architecture is guaranteed.



(a) Outcome of the AFEL architecture.



(b) Outcome of the Feed-Forward architecture.

Figure 4.6: Adaptation for the 3-DOF LWR arm using the AFEL architecture (a) and the Feed-Forward architecture (b) that contains an analytical model instead of the ULM module. Both figures show the average of the nMSE for three joints and over ten trials. Different traces indicate the response to different contexts. For the sake of clarity, error bars are plotted only for the 6 kg context and indicate the standard deviation between the trials.

Comparing Fig. 4.6a, related to the AFEL architecture, with Fig. 4.6b, related to the Feed-Forward architecture with the LF controller, the importance of the ULM module (LWPR + Cerebellum) becomes clear. In fact, in the second case, error bars are larger and the nMSE becomes higher as the load increases. However, results in Fig. 4.6a indicate the high quality of the estimate of the ULM output. In table 4.2 we present the maximum torque applied at joints for adaptation to different contexts. As the load at the last joint is increased, compliance is gradually achieved by gradually increasing corrective joint torques. As mentioned before, we also tested the performance of the

AFEL	**Maximum absolute Torques (Nm)**			**Minimum absolute Torques (Nm)**			**RMS (Nm)**		
	Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3
2 kg	80	189	76	0	0	0	45	97	35
6 kg	91	206	107	0	0	0	52	112	56
8 kg	99	231	122	0	0	0	56	128	65
10 kg	109	252	133	0	0	0	60	142	71

Table 4.2: The first two columns contain the maximum and minimum absolute torques applied at joints in adapting to the different dynamics models for the AFEL control system. The third column is the quadratic mean (RMS) of torques applied during all the iterations of the executed eight-like trajectory.

AFEL architecture in adapting to kinematics changes as well. The outcome is plotted in Fig. 4.7, which shows that performance is not affected either by changes in kinematics or by changes in dynamics. In this experiment, kinematics transformations applied at the robot plant consisted of different angles of fixed rotation of the end-effector (λ = [30,90]).

4.3.3 Self-adaptive Learning

Using an analytical method is not always possible in order to obtain a sufficiently accurate dynamics model which is needed for compliant robot control. In this case, it is necessary to adopt a new strategy. The u_{fb} feedback joint torques are given by the LF controller to control the arm. The LWPR modules receive their feedback command combined with its own prediction and the cerebellar output to form the feedforward motor command as a training signal. Then, in this experiment, there is no preliminary learning from an analytical model as in the previous approach. We repeat the trajectory 16 times for each context specified in Section 4.3.2. So, the LWPR still learns the u_{fb} global torque for the whole simulation while the LF adaptively controls the trajectory

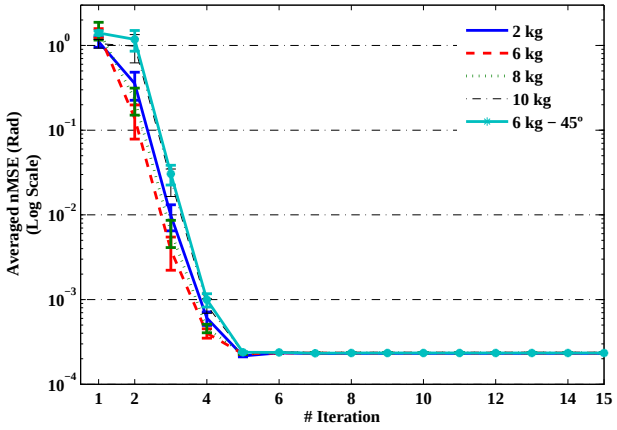


Figure 4.7: Robustness of the AFEL architecture under kinematics and dynamics transformations. For the sake of clarity, only $\lambda = 45°$, which is representative of all values of λ tested, as kinematics transformation is plotted. The average nMSE for three joints is averaged over ten trials. Error bars indicate the standard deviation between ten trials.

execution and the cerebellum optimizes the corrections. The task of this experiment is to follow the trajectory specified in Equations (4.17)

$$\begin{aligned} Q_1 &= Asin(2\pi t), \\ Q_2 &= Asin\left(2\pi t + \frac{\pi}{4}\right), \\ Q_3 &= Acos\left(2\pi t + \frac{\pi}{2}\right). \end{aligned} \tag{4.17}$$

In order to evaluate the relative relevance of the feedback contribution along the learning process, the used performance measure is the ratio of torque components to the total joint torque applied to the robot plant. Therefore, we defined the ratios in Equations (4.18) and (4.19):

$$R_{fb} = \frac{u_{fb}}{u_{fb} + u_{ff}}, \tag{4.18}$$

$$R_c = \frac{u_c}{u_{fb} + u_{ff}}. \tag{4.19}$$

Equation (4.18) represents the ratio of u_{fb} feedback torque to the u_t global torque and Equation (4.19) is the ratio of u_t cerebellar torque to the u_t global torque.

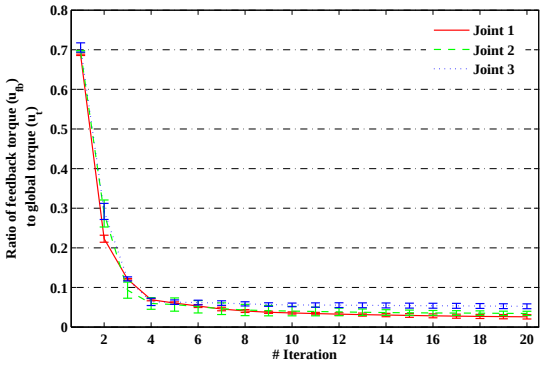
Figures 4.8a and 4.8b show how the average values (within each iteration) of these ratios evolve along the learning process. If the learned inverse model is accurate, the

ratios will be small as the error-correcting torque decreases over consecutive iterations. At the beginning of the simulation, the amount of ratio R_{fb} is higher than R_c, which means that the LF controller output contributes more to the global torque than the cerebellar torque and decreases significantly according to the reduction of errors. However, the second ratio, R_c, depends on the LWPR learning performance because they are connected by the RF weights of the LWPR local models which are the cerebellar granular weighting kernels. As a matter of fact, R_c decreases (see Fig. 4.8b) as the LWPR incorporates u_{fb} and u_c to its global output torque during the learning process.

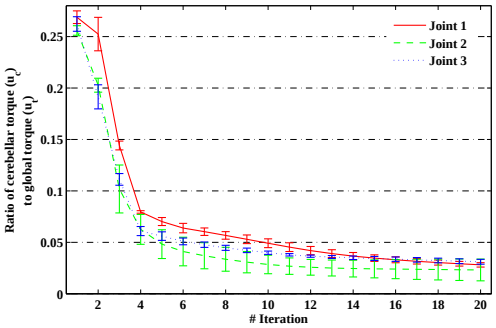
Figure 4.8c shows the ratio of the LWPR torque u_l to the global torque u_t, as defined in Equation (4.20):

$$R_l = \frac{u_l}{u_{fb} + u_{ff}}. \tag{4.20}$$

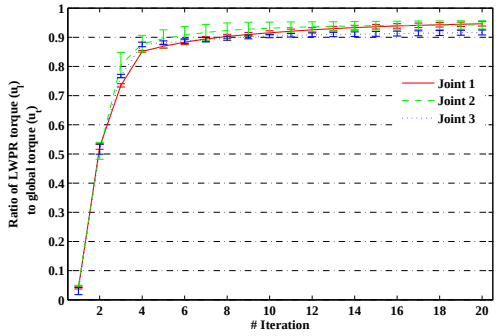
We can see that the LWPR algorithm progressively learns the u_t global torque which means that it really acts as internal model for the robot arm inverse dynamics.



(a) Contribution of correcting feedback commands R_{fb} computed as defined in Equation (4.18).



(b) Contribution of correcting cerebellar commands R_c computed as defined in Equation (4.19).



(c) Indirect measurement of how well the inverse dynamic model learned by the LWPR approximates the actual dynamics. Values are computed as defined in Equation (4.20).

Figure 4.8: Ratios of individual joint torque contributions to the global torque. Results are averaged over 4 trials (four contexts) and the error bars indicate the standard deviation between the trials.

In short, the more accurate the learned model is the finer will be the contribution of the cerebellum, because the LWPR optimally allocates the RFs for an efficient input mapping. In any case, both torque correction quantities, u_{fb} and u_c, vary depending on the nature of the system and on the error due to miscalibrations, contexts, noise, etc. In all cases, the u_{fb} feedback component and the u_c cerebellar torque will decrease as the LWPR incorporates their contributions in its internal model as a memory consolidation process. Furthermore, as displayed in Fig. 4.9, the tracking absolute error is very low and with low variance between trials (contexts). Again, the fact that the error slopes down is a result of the control stability provided by the LF controller and of the cerebellar optimization that improved the dynamic inverse model to be learned.

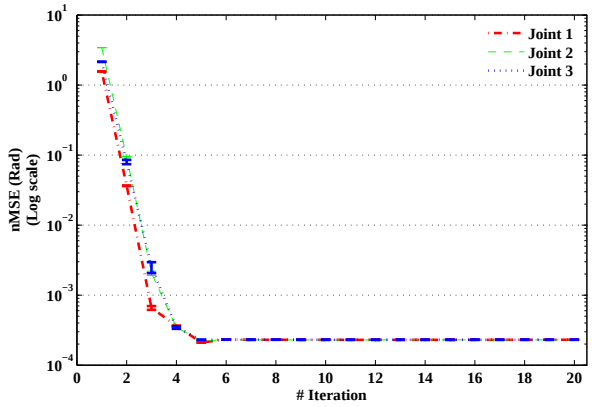


Figure 4.9: nMSE averaged over four trials (four contexts, i.e. movements manipulating 2,4,6 and 10 kgs).

Previous experiments are characterized by the fact that the LWPR learned the contexts separately. Anyway, the LWPR is capable of learning different forward dynamics models and retaining them in the same regression model. The figure 4.10 shows that after training forward dynamics models corresponding to the dynamics of the robot arm manipulating three different loads (2, 6, and 10 kg) at the last robot segment, the LWPR was tested with three unseen loads of 1, 4, and 8 kg to study its generalization capability to predict the plant behavior under new contexts. The robot was expected to follow the trajectory defined in Equations 4.17 in all cases.

We also tested other trajectories obtained from Equations (4.17) by changing the phase or the amplitude or summing two of them. In more details, we composed four trajectories with the following coefficients indicated in table 4.3. The performance

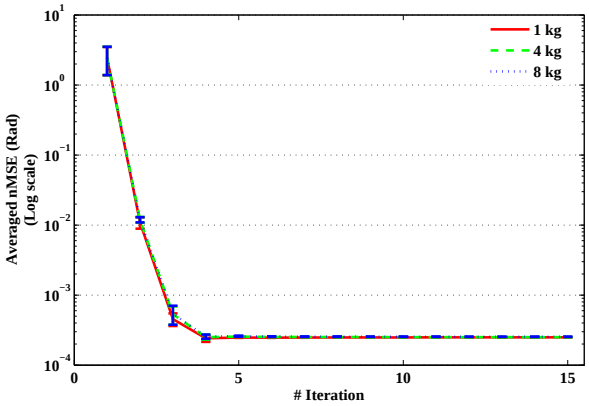


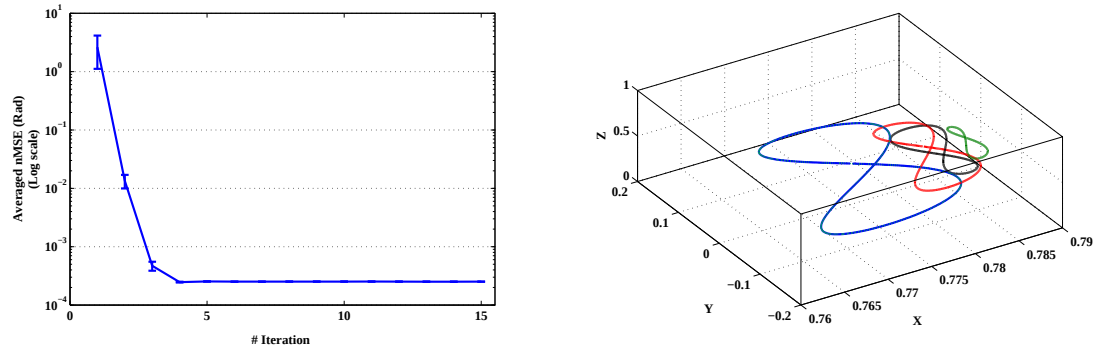
Figure 4.10: The AFEL architecture still has a good performance under testing unseen dynamics contexts. This is expressed by the nMSE value plotted in the figure. Error bars represent the standard deviation between three joints.

	A	ϕ
Traj. 1	0.1	$\pi/4$
Traj. 2	0.1	$\pi/2$
Traj. 3	0.05	$\pi/4$
Traj. 4	Traj. 1 + Traj. 3	

Table 4.3: Coefficients of the trajectories tested.

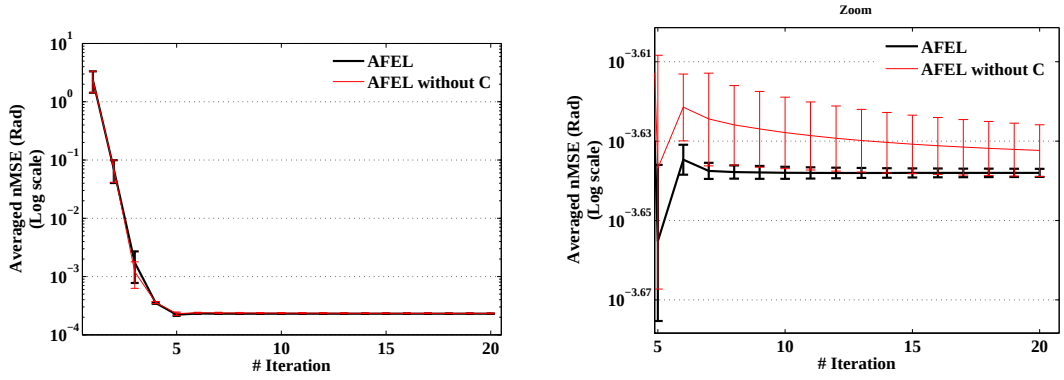
results of tracking four precomputed trajectories under manipulation of 6 kg load in the last robot segment are shown in Fig. 5.7a and their eight-like figure-shape are plotted in Fig. 4.11b which correspond to the final iteration number 15.

Evaluating the AFEL performance plotted in Figs. 4.9 and 4.10, we notice that the self-adaptive learning works and it is of high interest in case of unavailability of an analytical dynamics model. In addition, we can say that the cerebellum not only drives the LWPR learning engine to learn an optimized internal model, but it also contributes to deliver finer and more effective corrections for all contexts. For this purpose, we examine performances between the proposed AFEL system and an identical one in which the ULM module does not contain the cerebellar microcircuit. Results are plotted in Fig. 4.12.



(a) (b)

Figure 4.11: 4.11a. The AFEL architecture still has a good performance when performing the test stage with trajectories whose coefficients are defined in table 4.3. The robot arm manipulated 6 kg load at the end-effector-segment. The line indicates average nMSE value, averaged over the three joints first for each trajectory and then over the four trajectories. Error bars represent the standard deviation above and below the mean of the nMSE of the four trajectories. 4.11b. The eight-like figure-shapes obtained after 25 trials for the four precomputed trajectories (they are accurately approximating the desired trajectories).



(a) (b)

Figure 4.12: Averaged nMSE over three joints. Error bars represent the standard deviation above and below the mean of the nMSE of the three joints. The thicker solid line is related to the proposed AFEL architecture, while the other is related to the AFEL architecture without the cerebellar structure in the ULM module. Comparing them, we notice that the cerebellum optimizes the tracking error performance and drives the joints to a better convergence.

Finally, we have verified that the self-adaptive learning works efficiently on a more complex robotic platform too. Then, we have repeated the same experiment for the 7-DOF LWR arm, measuring the outcome performance in terms of nMSE and computing the ratios described in Equations (4.18), (4.19), and (4.20). The eight-like target trajectory to be followed by the arm tip is defined in Equations (4.21):

$$y = 0.15 \cdot sin(2t), \tag{4.21}$$
$$z = 0.6 + 0.2 \cdot cos(t),$$

and the variable x is a constant.

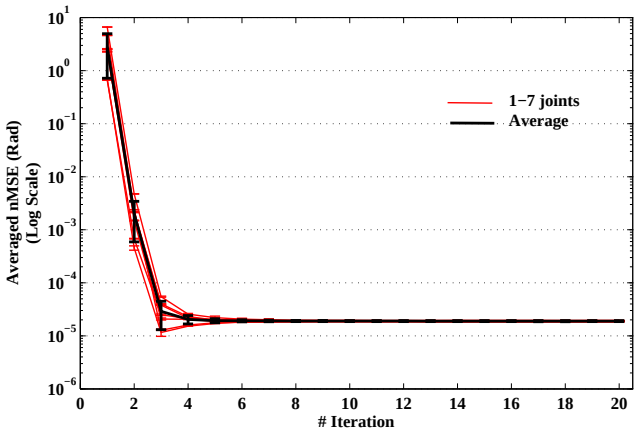
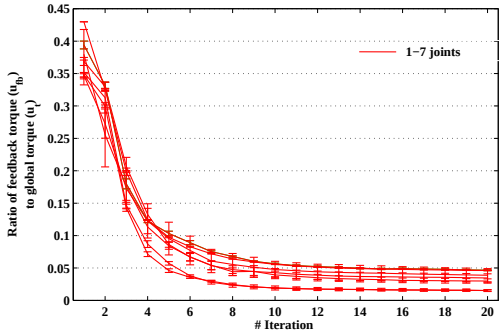


Figure 4.13: nMSE averaged over four trials (four contexts). The thicker and darker line is the average over the 7 joints.

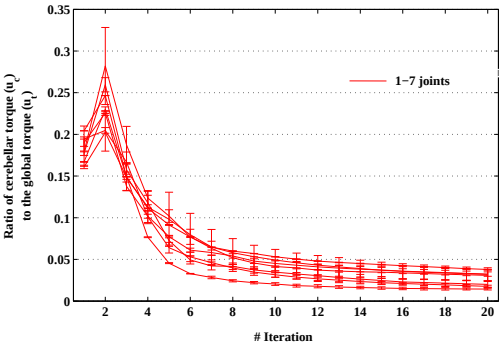
Results in Fig. 4.13 indicate that the AFEL architecture also works for high DOFs. The nMSE is low and after 5 iterations, the system's behavior becomes stable. The LWPR has approximated the dynamics model of the LWR arm well and this model achieves a high performance.
Figure 4.14 shows the ratios of each individual component torque, u_{fb} and u_c, with respect to the global joint torque, u_t, while Fig. 4.14c reveals that the LWPR output increases according to its gradual learning of the global torque, u_t.

To conclude, even though the improvement in the case of inclusion of the C module is not so significant, the AFEL (LWPR + C module) architecture presents a better error performance since the cerebellar module effectively contributes to drive the learning

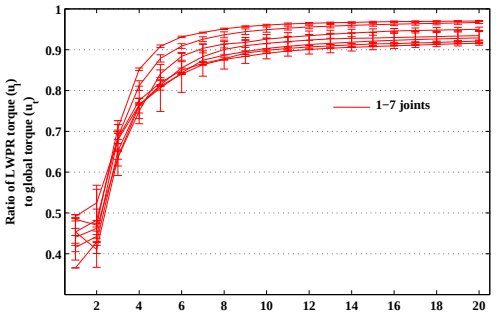
process at LWPR faster (conjoint convergence among the different joints to a smaller error value) (see Fig. 4.15). Anyway, in this case, we attribute the negligible improvement in terms of performance to the fact that the representation of the input space was not dissimilar enough to be sparse. More importantly, this result also indicates that the AFEL scheme (including a cerebellar module in the ULM) is scalable in terms of number of joints. For the 7-DOF LWR arm, we obtained good results in terms of error, just as for the 3DOF case. However, it should be noted that the dynamics of a real system will significantly be more complicated than the simulated dynamics, as there are important nonlinear effects that are not simulated, such as actuator dynamics or elasticity. Furthermore, as already indicated by other authors, it has been shown that learning of dynamics using LWPR on real world high DOF robotic platforms works very efficiently [11].



(a) Contribution of correcting feedback commands R_{fb}
computed as defined in Equation (4.18).

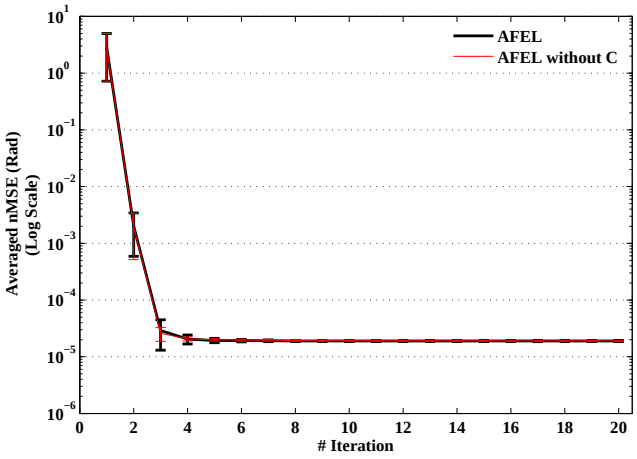


(b) Contribution of correcting cerebellar commands R_c
computed as defined in Equation (4.19).

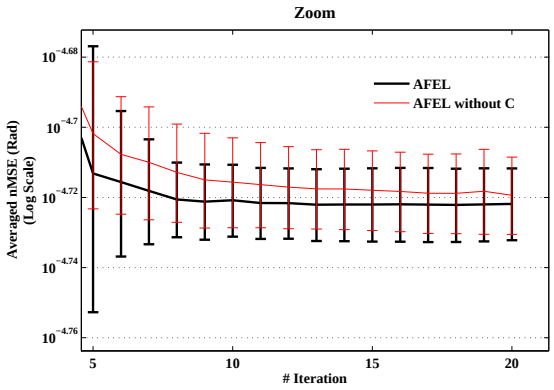


(c) Contribution of LWPR feed-forward commands R_l to the u_t global torque computed as defined in Equation (4.20).

Figure 4.14: Ratios of individual torque contributions to the u_t global torque. Results are averaged over the four trials (four contexts) and the error bars indicate the standard deviation between the trials.



(a)



(b)

Figure 4.15: Normalized mean square error (nMSE) averaged over seven joints. Error bars represent the standard deviation above and below the mean of the nMSE of the seven joints. The thicker line represents the nMSE of the proposed AFEL architecture. The other line shows the tracking error performance obtained by removing the cerebellar circuit from the ULM module in the AFEL scheme.

4.4 Discussion

We implemented a model for the motor control of robotic arm movements in which machine learning and biologically inspired approaches co-exist and complement each other. The presented AFEL scheme, which takes advantage of the connection between the accurate regression method LWPR and a basic cerebellar structure, works properly. Furthermore, the cerebellar module takes full advantage of the LWPR for efficiently abstracting the high dimensional input space.

A potential role of the granular and molecular layers in biologically plausible cerebellar models is to provide accurate signals to the PCs for improving the learning of the current model. These signals seem to influence the DCN synapses capability in order to consolidate the learning [59, 63, 64]. In our model, the input LWPR RFs are used as a representation of the granular and molecular layers delivering clean and accurate signals to the Purkinje layer. Therefore, LWPR provides optimal input representation to the Purkinje layer in terms of neural resources (it adapts its neural resources incrementally and according to the input data structure). The importance of this efficient and clean contribution has been evaluated recently [59, 65] and other authors, such as Schweighofer et al. [17] (2001) and Porrill and Dean [8] (2007) hypothesized that the cerebellar learning is facilitated by optimizing the choice of the centers and the basis of RFs at the granular layer. As a matter of fact, LWPR creates around 16 locally linear models for the 3-DOF robot arm, 60 locally linear models for the 7-DOF robot arm, and it allows the selection of only predictions from those who elicit great activation for a query point.

It is also important to note that the LWPR also works as a memory consolidation module. Therefore the cerebellar module with its learning rule is focused on short term adaptation while long term memory consolidation takes place at the LWPR module. Thus, in terms of short/long term learning by analogy with biological systems, our cerebellar module, receiving inputs from the LWPR RFs, represents the MF-GrC/interneurons-PC pathway for short time learning while the LWPR adaptation kernels represent the MF-DCN adaptive pathway which is responsible of long term learning (memory consolidation) [59, 66].

We exploited the LWPR characteristics to acquire the dynamics of a robot arm through the learning process as the movement is repeated. Once the inverse model is learned, the system can perform the task precisely and small correction forces are required, thus increasing the compliance. In order to achieve compliant movements and ensure the system stability, high feedback gains must be avoided, specially in biological systems (or robotic platforms targeting at human machine interaction tasks). To avoid this, we propose the LF controller, which adapts the error-correcting feedback over consecutive iterations of the same task. The LF controller supplies the error to the cerebellar network. Furthermore, the LF controller will accurately guide the LWPR during the learning process using very low gains. Results show that the global architecture has a compliant performance which is suitable for robotic systems in human environments. Haddadin et al. [67] (2007) evaluated the influence of the mass and velocity of DLR-LWRIII in resulting injuries on human bodies. Their impact tests were carried out by using the Head Injury Criterion (HIC) since it is the most prominent indicator of head injury in automobile crash-testing; the results of this tests suggest that a robot, even with arbitrary mass moving not much faster than 2 m/s is not able to become dangerous to a non-clamped human head with respect to typical severity indices. In addition to this, their investigation revealed that the inertia properties of the LWRIII allow an impact velocity of up to 1 m/s without leading to soft-tissue injuries. With regard to our system, due to the fact that the linear velocity is the cross product between the angular velocity and the position of the end-effector with respect to origin, we have considered the worst case, i.e. the longest link and the maximum angular velocity in order to simplify the computation. The maximum linear velocity obtained during the self-adaptive experiment for the 3-DOF LWR arm manipulating 10 kg load at the last joint is 0.62 m/s, therefore the system performs in a compliance manner.

The performance obtained by the AFEL scheme in terms of error after learning is remarkable compared to other approaches [22, 45]. This is of high interest taking also into account that in this architecture, the LWPR does not require an analytical preliminary dynamic model to learn from it, as it learns directly from the feedback torques and the cerebellar compensatory torques. As a matter of fact, LWPR works as an internal model abstraction kernel whose learning process is guided by the LF instead

of having a reference analytical model.

Porrill and Dean [8] (2007) mentioned the motor error problem due to the complexity on the reference structures used to compute the error for the forward connectivity. This problem particularly affects biological nonlinear motor systems since the number of the reference structures is multiplicative in the dimension of the control and sensor space. In our feedback error learning approach we solved the motor error problem thanks to the LF controller, we proved its performance in the task of a simulated 3 and 7-DOF robot arm. We demonstrated that the combination of feedback and feedforward estimates does offer considerable advantages for robust online control. Furthermore, we ensured accuracy and enhanced the speed of learning by optimizing the choice of the centers and transforming to an optimal basis of rfs through the LWPR algorithm.
Schweighofer et al. [17] (2001) hypothesized that cerebellar learning is facilitated by a granule cell sparse code, i.e. a neural code in which the ratio of active neurons is low at any time. They proposed a diagram of the model of cerebellar control two-joint arm movements in which the cerebellum learns how to compensate for interaction torques that occur during reaching movements. In analogy with their approach, in our control system, there are three motor commands: feedforward motor commands (by the forward internal model), feedback torques (by the LF controller), and the cerebellar compensatory torques (by the C module) which are summed and sent to the robot arm plant. The cerebellar torque values are necessary for precise control and the cerebellar network is embedded in the control model. Due to the large number of DOFs and the pervasive nonlinearities of the seven degrees of freedom human arm, an internal model of the arm's dynamics is an extremely complex mapping between kinematic and dynamic variables and thus requires a large number of encoding states [17]. Despite this, we demonstrated that we achieved very good performances with a small number of states (or GCs) for a 7-DOf robot arm.

5

Feedback Corrective Contributions. Recurrent Adaptive Architecture.

The adaptive structure of the cerebellar microcircuit receives a teaching or error-driven signal, and sensory-motor signals as inputs for motor control in the process of acquiring internal models related with the input-output correlations. In brief, motor commands or sensory signals arrive to the GC layer and corrective motor commands or sensorial terms are also produced as output of the PC, while the CF adapts the synaptic weights and conveys motor or sensory errors. In this chapter, in the case of sensory signals, the cerebellum circuitry is represented in a recurrent loop which solves the motor error or distal error problem contrary to the scheme described in the previous chapter. Here, we focus on how to decompose the input into different components in order to facilitate the learning process using the LWPR algorithm. The LWPR automatically optimizes the network size by incremental learning of the forward model of the robot arm and provides the GC layer with optimal pre-processed signals. We present a recurrent adaptive architecture in which a LF controller leads a precise, compliant and stable control during manipulation of objects and the cerebellar-LWPR synergy makes the robot adaptable to changing conditions. Then, we evaluate how the scheme scales for plants of high number of DOFs using a model of a robot arm of the new generation of light-weight robots.

5.1 Introduction

Cerebellum's behavior has commonly been emulated in a feedforward control architecture based on feedback error learning (FEL), where it delivers feedforward corrective terms to the crude control commands from the motor cortex [12, 13, 21, 35, 36, 37, 68] and the teaching signal is computed by the difference between actual and correct/desired motor commands, that is the motor error. For example, Kawato [12] (1990) and Wolpert et al. [13] (1998) proposed an architecture based on feedback error learning emulating the role of the cerebellar microcircuit. The motor error is the difference between the desired and the actual motor commands. However, the correct motor command is typically unknown; only sensory errors are available, and how to use this information for motor learning represents the so-called *distal error* problem [24]. Porrill and Dean [8] (2007) stated that this kind of approach requires complex neural structures to estimate the motor error, thus they advocated *the recurrent architecture* being more biologically plausible. In fact, unlike in the feedforward architecture, in the recurrent control system, sensory or teaching error signals, which are needed for the learning process as they adaptively adjust the cerebellar weights [7], are physically available signals [8, 15]. They represent the mismatch between the desired and actual movement, as shown in Fig. 5.2. The role of the recurrent system solves the so-called motor error problem or distal error problem [24], that is the motor commands might not be known in advance and so cannot be used to generate error signals. In this scheme, in contrast with the forward connectivity, the cerebellum receives copies of the motor commands as input and it delivers correction terms to be added to the desired signal trajectory. Furthermore, both forward and recurrent pathways are complementary in adaptations problems [37, 50]; the former allow compensation for changes in upstream structures of the cerebellum and the latter in downstream structures [15].

In adaptive filters, a central feature is the decomposition of input into different components, which in the cerebellum has been assumed to take place at the granular layer. GCs receive direct input from multiple MFs and their output has to be processed in order to produce the signals carried by the PFs. Porrill and Dean [8] (2007) argued that the choice of appropriate basis seems to influence both speed of learning and accuracy. In our approach based on the recurrent adaptive control architecture, we

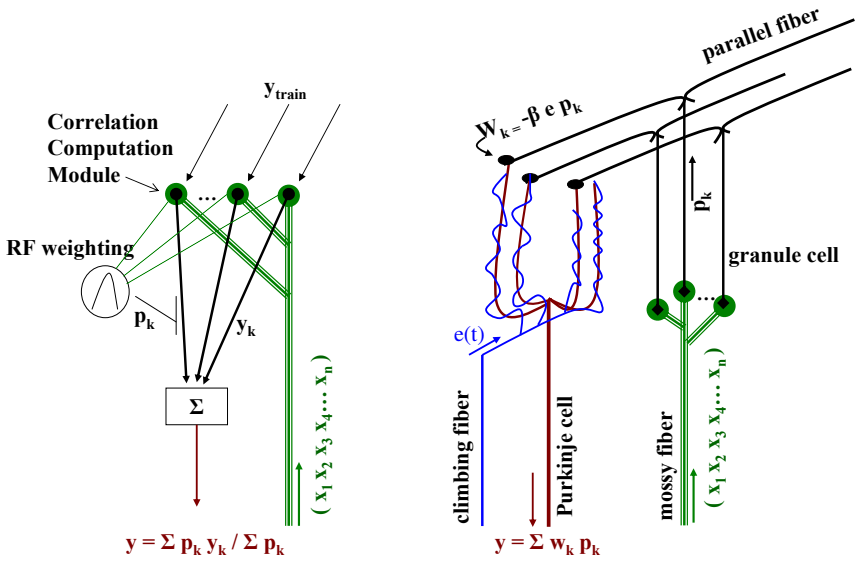


Figure 5.1: Comparison of the LWPR processing unit with the cerebellum microcircuit.

address the problem using the machine learning algorithm LWPR [10, 11], which learns and emulates the forward dynamic model of the arm during the simulation. Fig. 5.1.a shows the LWPR processing unit and working mechanism. Although we have indicated it already in the previous chapter, we repeat the similarities here to make the chapter self-contained. The LWPR creates k local models that have a region of validity (RF) parameterized as a distance metric D_k in a Gaussian kernel. Comparing it with the simplified cerebellar circuitry in Fig. 5.1.b, we notice that the RF weighting kernels encode the input space like the cerebellar GCs expansively encode the information coming from the MFs. Yamazaki and Tanaka [58] (2007) stated that the granular layer generates a finite but very long sequence of active neuron populations without recurrence. In a similar way, RF weighting kernels could adapt their weights p_k for each x_i data point in each k local unit to select different outputs depending on the efference motor command and the current state of the robot arm. Along with this analogy of the cerebellar circuit, each LWPR module and its associated RF weights can be seen as providing the firing rate of a PF, while the set of active RF weights can be seen as the current state

of the granular layer processing module [58]. This state would be propagated through PFs and interneurons to produce more accurate signals to the PCs [59]. With regard to our model, the LWPR learned function will be able to provide the cerebellum module with optimal pre-processed signals p_k and the LWPR automatically evaluates the required number of local correlation modules to optimise the network size by incremental learning. In this sense, machine learning algorithms and robot systems may help us improve our understanding of how learning processes take place in biological systems.

As also indicated in the previous chapter, with regard to the control experimental set-up, we use a model of a robot arm of the new generation of light-weight robots [52, 53] (see Fig. 4.2) whose major drawback is that its complex dynamics are no longer controllable by traditional methods since the movement is influenced by the state variables of all the joints and the control becomes very complex and highly non-linear [20]. Therefore, feedback control gains have to be increased to compensate nonlinearities and avoid the resulting tracking error, but they are dangerous regarding the system stability and unacceptable in autonomous and biological systems [8] as they introduce large forces generating potentially dangerous non-compliant movements [22]. Thus, we propose the LF controller which transforms the trajectory error in sensory coordinates, into a motor command, it improves the system behaviour and self-adapts by a learning rule through consecutive iterations of the same trajectory. It works as an adaptive control model and it leads to a precise and stable control during manipulation. In fact, the learned adapted dynamic arm model together with the LF controller make the robot arm capable of performing movements that are both precise and compliant at the same time and also adaptable to changing situations similar to what occurs in a biological system.

In this chapter, we will present a recurrent adaptive architecture for robust arm control and will focus on the role of forward models which consists of predicting the next state (positions, velocities, and accelerations) given the current state (positions, velocities and accelerations) and the motor command or efferent copy [23, 69]. In practice, the cerebellum provides appropriate sensory corrections in response to changes in the plant, while the LWPR incrementally learns the dynamic models of the arm+object receiving the torques or motor commands (efferent copy) as input data. The RFs weighting kernels encode this input space and send preprocessed signals to the synapses PFs-PCs in order

to update the synaptic weights and compute the PC output. The cerebellum-LWPR connection makes the robot also adaptable to changing situations; we have fused their functionalities to take advantage of the efficiency of the machine learning algorithm and of the cerebellum's role to make fine adjustments to the way an action is performed. In other words, the approach we present exploits both the efficiency of the machine learning engine (LWPR kernels) and the bio-inspired module (cerebellum circuitry) to learn different non-linear forward dynamics as an optimization of dynamic nonlinear-control scheme in a robot for robust control. Dynamics are encoded in the Forward Model Unit showed in Fig. 5.2. This unit predicts the next state from the current state of the robot arm and the efference copy or motor command u. The cerebellum module (block C in Fig. 5.2), instead of receiving inputs by means of MFs, receives pre-processed signals from the LWPR RFs (G_k) and it computes corrections (S_c, $\dot{S}_c$, $\ddot{S}_c$) to be added to desired positions, velocities, and accelerations (Q_d, $\dot{Q}_d$, $\ddot{Q}_d$) as shown in Fig. 5.2. Then, sensory feedback (Q, $\dot{Q}$, $\ddot{Q}$) (i.e. actual position, velocity and acceleration) is subtracted to it and sent to the LF controller that generates the motor torque u assuring the stability of the system without using high gains (this is done making use of a learning error rule).

LWPR has already been used for online incremental learning in robotic platforms [11, 22, 54, 57] since it exploits spatially localized linear models to approximate non-linear functions at low computational cost. The evaluation of the prediction value is rather fast allowing real-time learning. On the other hand, Gaussian Process Regression (GPR) [55], or Support Vector Regression (SVR) [56] have the advantage that they do not depend on a large number of parameters as in LWPR, and they are therefore easier to tune [70]. Anyway, the incremental training permits the acquisition and retention of different tasks without restriction in the number of tasks and without interferences among them [45]. The major strength of the LWPR is that the algorithm can cope with highly redundant and irrelevant data input dimensions, without any prior knowledge of the data, this is so because it uses an incremental version of Partial Least Squares regression (PLS). Sigaud et al. [70] (2011) provided a survey that focus on differences among the main regression methods. LWPR has been convincingly applied to the identification of diverse models (kinematics, velocity kinematics and dynamics, forward and inverse) of large mechanical systems [70]. In particular, considering the forward dynamic model learning case, LWPR has been used to learn the forward dynamic model

of a two DOFs planar robot [44].

In the following paragraphs, we will present the advantages of our system according to the next structure. In the first section, we will explain the Recurrent Adaptive Feedback Error Learning (RAFEL) control scheme showing its connections and system equations. Then, we will discuss the properties or characteristics of the forward model learning algorithm (LWPR), the cerebellar microcircuit and their relationship. Finally, we will demonstrate the performance of the presented model with experiments on a 3-DOF and a 7-DOF LWR arm (see Fig. 4.2).

5.2 Control System Architecture

The RAFEL block diagram is shown in Fig. 5.2. The architecture proposed here uses a feedback controller called LF which adapts the joint feedback torques u, also called efference copy, over consecutive iterations of the same task. Signal u represents the input for the learning and prediction of the forward model (LWPR). Command torques u are computed by the LF controller from the global errors (e_{pg}, e_{vg}, e_{ag}) (5.1) (where p, v and a stand for position, velocity and acceleration respectively, and g stands for global) obtained by the sum of the feedback errors (e_p, e_v, e_a) and the sensorial corrective terms (S_c, $\dot{S}_c$, $\ddot{S}_c$) produced by the simplified cerebellar microcircuit (5.2, 5.3).

$$u = LF(e_{pg}, e_{vg}, e_{ag}). \tag{5.1}$$

$$(e_{pg}, e_{vg}, e_{ag}) = (Q_d, \dot{Q}_d, \ddot{Q}_d) - (Q, \dot{Q}, \ddot{Q}) + C(p_k, e_p, e_v, e_a), \tag{5.2}$$

or in a more compact form:

$$(e_{pg}, e_{vg}, e_{ag}) = (e_p, e_v, e_a) + (S_c, \dot{S}_c, \ddot{S}_c). \tag{5.3}$$

As a matter of fact, the cerebellar circuitry applies corrections for miscalibration of the system adjusting its weights by the (e_p, e_v, e_a) feedback errors. These represent the difference between the desired state (Q_d, $\dot{Q}_d$, $\ddot{Q}_d$) and the output of the controlled robot arm ($Q, \dot{Q}, \ddot{Q}$) at time t (see expressions 5.2, 5.3). From a biological point of view, the block C in Fig. 5.2 consists of a set of uniform cerebellar circuits capable of learning, and the error signal conveyed by the CF adaptively modifies its input-output system's

characteristic function. Further details about C are given in subsection 5.2.2. In our approach, the LWPR algorithm is implemented to play the important role of the forward model, which means that it learns the forward dynamic model of the robot arm. The machine learning engine and the cerebellar cortical circuitry complement each other; the former takes advantage of the adaptive cerebellar corrections through the efference copy and the latter of the efficient reduction of the high dimensional input dimensions, of the incremental learning and of a compact sensory-motor representation by a bank of p_k filters.

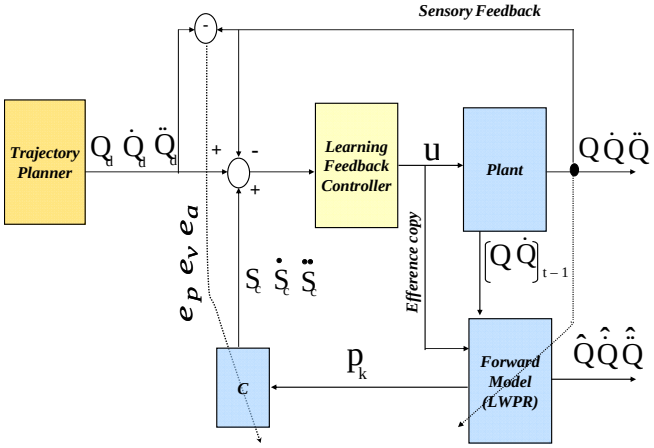


Figure 5.2: Block diagram for the RAFEL architecture.

The field of nonlinear control theory is very large, we will focus our attention on a particular method called feedforward nonlinear control [61]. Considering the analytical robot model, we calculate the joint torques required for a particular trajectory using the following dynamic expression (5.4) of the robot:

$$u = M(Q)\ddot{Q} + V(Q, \dot{Q}) + G(Q) + F(Q, \dot{Q}), \tag{5.4}$$

where $M(Q)$ is the inertia matrix of the manipulator, $V(Q, \dot{Q})$ represents the centrifugal and Coriolis terms, $G(Q)$ is the gravity term, $F(Q, \dot{Q})$ is the model of friction, and Q, $\dot{Q}$, $\ddot{Q}$ are the joint angles, velocities, and accelerations of the robot arm. In the

architecture, the u joint torques generated by the LF controller allow the robot following a desired trajectory ($\dot{Q}_d$, $\dot{Q}_d$, $\ddot{Q}_d$) and ensures the stability of the trajectory at the same time. The specifics and equations of the LF controller are presented in the next subsection 5.2.1.
If the cerebellar output is accurate and the forward dynamic model is exact, the LF controller will cancel the nonlinearities transforming the resulting global errors (e_{pg}, e_{vg}, e_{ag}) in the correct torque u. However, there will be a difference between the desired signal and the actual output of the controlled arm that will activate the cerebellar process of learning [16].

Summarizing, the cerebellum elaborates the sensorimotor complexes coming from the optimized representation done by the LWPR and produces the sensorial corrective terms (S_c, $\dot{S}_c$, $\ddot{S}_c$), which added to feedback errors enable the LF controller to adapt the u motor commands. The LWPR engine incrementally learns from the u torques and the current state of the robot arm (Q, $\dot{Q}$)$_{(t-1)}$ (see Fig. 5.2) and abstracts the whole forward dynamic model.

5.2.1 Learning Feedback Controller

In this subsection we address the problem of controlling a robot arm of many DOFs. Taking into account the fact that an analytical computation of the robot dynamics is complex and a large number of parameters may be unknown, we implement an adaptive model for an accurate and stable control during object manipulation: the LF controller. The LF controller ensures the stability of the system producing the appropriate joint torques to obtain the right execution of the desired task in a compliant way. From Equation (5.5), that represents the estimated robot arm dynamic model, we notice that the u_r feedback joint torque varies after consecutive repetitions of the same task, $r = 0, 1, ...$

$$\tau = \hat{M}(Q_{dc})\ddot{Q}_{dc} + \hat{V}(Q_{dc}, \dot{Q}_{dc}) + \hat{G}(Q_{dc}) + \hat{u}_r, \tag{5.5}$$

Where Q_{dc}, $\dot{Q}_{dc}$, and $\ddot{Q}_{dc}$ are the sum of the desired motions and the cerebellar joint corrections (see Fig. 5.2).

In the previous Equation (5.5), $\hat{u}_r$ joint torque is a non-modeled friction term added to the other estimated terms, this accounts for the dynamic model which is not well

known or not precise. Substituting Equation (5.5) in Equation (5.4), we obtain the error Equations (5.6) and (5.7) of the closed loop of the control system in Fig. 5.2:

$$\ddot{e} = M^{-1}(F - \hat{u}_r), \tag{5.6}$$

or in a more compact form:

$$\ddot{e} = (B - B_r), \tag{5.7}$$

where $\ddot{e} = (\ddot{Q}_{dc} - \ddot{Q})$, $B = M^{-1}F$, $B_r = M^{-1}\hat{u}_r$. For every joint, the Equation (5.7) becomes:

$$\ddot{e}_i = (B_i - B_{ir}). \tag{5.8}$$

In the last expression (5.8), term B_i is constant during iterations over time, while the term B_{ir} changes on consecutive iterations of the task. We propose the following learning rule (5.9) for each joint i:

$$\hat{B}_{i(r+1)} = \hat{B}_{ir} + P * e_{ir}, \tag{5.9}$$

where $P * e_{ir}$ is the convolution between the impulse response filter P and the error in iteration r. We choose the specific filter defined by equation (5.10):

$$P(s) = s^2 + (K_{vi} - \mu)s + (K_{pi} - \mu), \tag{5.10}$$

where μ is a constant.
P is a non-causal filter, so it uses the errors of the previous iterations and its convergence depends on μ. Further specifics on the LF controller analysis are provided in Appendix A.

5.2.2 The Role of the Forward Model in the Cerebellum

As we mentioned in section 5.2, LWPR works, in the presented architecture, as a model abstraction engine capable of learning the forward dynamic model of the controlled robot arm+object (the object being manipulated).
LWPR is an algorithm for nonlinear function approximation in high dimensional spaces with redundant and irrelevant input dimensions. In order to perform optimal function approximation, LWPR incrementally divides the input space into a set of k RFs defined by the center c_k and a Gaussian area characterized by the particular kernel width D_k as shown in Equation (5.11). Fig. 5.1.a represents its processing unit where M inputs enter

into K linear local models automatically created and located. For each data point, a weight $p(k,i)$ is computed inside each local unit to measure the input locality according to the distance from the center c_k of the kernel. In other words, the weight measures how often an item of data xi falls into the region of validity of each linear model. The kernel function is defined as a Gaussian kernel:

$$p_k = exp\left(-\frac{1}{2}(x_i - c_k)^T D_k (x_i - c_k)\right), \tag{5.11}$$

where D_k is called distance matrix and defines the size and shape of the region of validity of the k linear local model.
Every time a data sample falls into the validity region of the local model, its distance matrix and regression parameters are updated. This happens independently for each model. In fact, every model makes its own prediction and the combination of all the individual predictions $\overline{y}_k$ is the total output $\hat{y}$ (see Equation (5.12)) of the LWPR network. In other words, the LWPR prediction is the weighted mean of all linear models:

$$\hat{y} = \frac{\sum_{k=1}^{K} p_k \overline{y}_k}{\sum_{k=1}^{K} p_k}. \tag{5.12}$$

Forward models predict the next state of each joint (e.g. position, velocity and acceleration) given the current state and the efference copy [13, 33, 69]. So, LWPR input data consist of torques, current positions and velocities of all the joints, while the output data are the predictions of the outgoing positions, velocities and accelerations. More in detail, the LWPR training takes place for each DOF separately, we set up [3 x number of joints] modules, with a test set of [3 x number of joints] inputs (torques, current positions, current velocities) and 1 target. Each LWPR module retains either the next joint position, or the next joint velocity or the next joint acceleration as target signal (see Fig. 5.2). The goal of learning is to make prediction errors converge to zero, thus providing an optimized representation of the sensorimotor complexes to the cerebellum module.

Comparing the LWPR processing unit and the cerebellar microcircuit shown in Fig. 5.1, it is remarkable their similarity in the way they encode the input space. In the cerebellum, information coming from the MFs is expansively distributed over many

GCs to produce other signals carried on the PFs. The GCs can be considered as a bank of linear filters that must be sufficiently rich to provide correct adaptive synaptic weights. As mentioned before, LWPR optimizes the network size by encoding the input space through the RF kernels. In Fig. 5.1.a are shown inputs $(x_1, x_2, ..., x_i)$ that enter through every local model and each of them produces the $\hat{y_k}$ signal seen in Equation (5.12). Then, RF kernel functions can be thought as a bank of filters G_k defined in the following Equation (5.13), and the output signals p_k are driven by the PFs and interneurons by analogy with the cerebellum.

$$p_k = G_k(x_1, x_2, ..., x_i), k = 1..K, i = 1..M. \tag{5.13}$$

Then, we take advantage of the LWPR local kernels as granular and molecular layer microzones in the cerebellum model. This does benefit from the optimized engine for a compact sensory-motor representation provided by the LWPR.
PC output $z_k(t)$, defined in (5.14), is modeled as a weighted linear combination of the $p_k(t)$ computed by equation (5.13):

$$z_k(t) = \sum_k w_k p_k(t). \tag{5.14}$$

The adaptive synaptic weights w_k of the k_{th} PF-PC synapse (see Fig. 5.1.b) are updated using the heterosynaptic covariance learning rule (5.14) [6] in the continuous form [8], and adjusted by an $e(t)$ teaching or error signal (the CF input).

$$\delta w_k = -\beta e(t) p_k(t), \tag{5.15}$$

where β is a small positive learning rate and $e(t)$ is the error signal carried out by the CF. In this approach, $e(t)$ is the feedback error, which has three components e_p, e_v, and e_a (positions, velocities and accelerations) for each joint. β is 0.005 in all the experiments carried out. The cerebellum produces [3 x number of joints] position, velocity and acceleration corrections for every joint arm, which are updated by the corresponding weight w_k.

Summarizing, all RFs calculate their weight activation in order to assign the new input, x_i, to the closest RF and consequently the center and the kernel width are incrementally updated during the learning process. The optimized choice of centers and

widths gives the optimal basis of RFs, so that the accuracy and the learning speed of the cerebellar model are improved. In other words, Equation (5.13) represents the bank of filters for the GCs in the cerebellum and they are both used to compute the cerebellar outputs (S_c, $\dot{S}_c$, $\ddot{S}_c$) as a linear weighted combination as defined in Equation (5.14) and to update the synaptic weights w_k (5.15).
Finally, the function to be approximated online, during simulation, by the regression algorithm LWPR is shown in the following Equation (5.16):

$$\Phi(u, (Q, \dot{Q})_{t-1}) = (Q, \dot{Q}, \ddot{Q})_t \tag{5.16}$$

Where $(Q, \dot{Q})_{t-1}$ are the current state and $(Q, \dot{Q}, \ddot{Q})_t$ are the next state of the robot that corresponds to the efference copy of the motor command u. Further implementation details on the approximation process are given in the section 5.3.

5.3 Materials and Methods

In order to change the dynamics of the robot arm and to allow the forward model (LWPR) learning different context's dynamics, we considered four case-studies when emulating manipulation of different objects, in which the manipulated object was the last link of the arm with a changed tip in terms of mass. The masses are 2, 6, 8, and 10 kg, respectively. So, we have studied how the tracking errors are compensated by the cerebellum and the LF contributions following the desired trajectory under different contexts. Moreover, we have tested the stability of the RAFEL architecture under kinematic modifications obtained changing and fixing a certain orientation shift of the end-effector. In other words, we have evaluated the performance of the RAFEL architecture in adapting to dynamics and kinematics changes of the controlled object on two physically realistic models of robotic arms. In the first setup, the LWR arm was simulated considering a reduced configuration to 3 DOFs (fixing the rest of the joints) in order to get fewer input dimensions to the machine learning engine. The three non-fixed joints (Q_1, Q_2, Q_3) used in our experiments are indicated in Fig. 4.2 in Chapter 4. This reduces the amount of training data required, and expedites the initial learning process. Afterwards, all 7 DOFs of the DLR LWR III shown in Fig. 4.2 were involved in the simulation.
LWPR learning was carried out online and the learned forward models were adapted

to changes in dynamics in real-time at every time step. The initial LWPR's forgetting factor was set to $\lambda = 0.995$, the final λ was set to 0.99995, and the annealing constant of 0.999; in this way we created a trade-off between preserving what has been learned and quickly adapting to the non-stationarity.
However, for both configurations, the gains of the LF controller have been set to the same value for the four objects and for all the robot joints ($K_p = 6$; $K_v = 3$; $mu = 0.75$). Nakanishi and Schaal [29] (2004) provided a strictly positive real (SPR) condition, that is $K_v^2 > K_p$, for choosing feedback gains in order to ensure stability of the feedback error learning mechanism. The choice of feedback gains we have made satisfies the mentioned condition, which implies that the stability of the AFEL architecture is guaranteed.
The task for the experiments with the LWR 3-DOF arm was to follow a planned trajectory in a 3-dimensional task space defined by (5.17).

$$\begin{aligned} Q_1 &= Asin(\pi t), \\ Q_2 &= Asin(\pi t + \phi), \\ Q_3 &= Asin(\pi t + 2\phi), \end{aligned} \tag{5.17}$$

where A is a constant (0.1), ϕ is $\pi/4$ and Q_1, Q_2, and Q_3 are the joint coordinates, respectively. However, in the second setup, in order to scale the movement to a 7-DoF scenario, the arm had to follow the trajectory defined in Equation (5.18):

$$\begin{aligned} y &= 0.15sin(2t), \\ z &= 0.6 + 0.2cos(t). \end{aligned} \tag{5.18}$$

We ran 25 iterations of the previous desired trajectories, and trained the LWPR with 12500 samples in the first setup and 50000 for the second setup. With the purpose of highlighting the importance of the cerebellum in the system and the role of the forward model in making more effective cerebellar corrections, we have also compared the performance between the RAFEL architecture with another system configuration in which the cerebellum module has been removed and another architecture with a high-gains PD instead of the LF controller.
In order to evaluate the functional structure of the LWPR forward model and its capability to generalize among contexts, we have done a generalization experiment. We

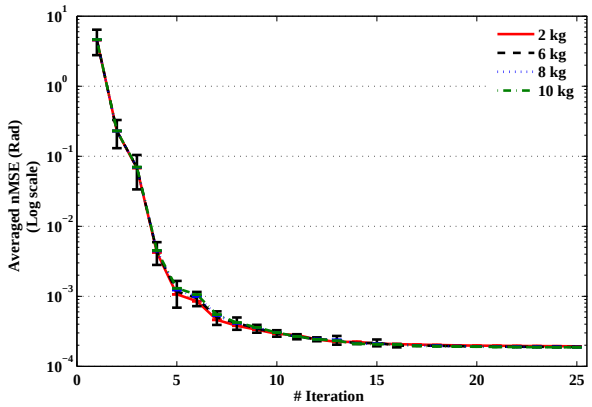
tested the LWPR capability to provide an optimal basis of RFs to compute the cerebellar corrections under unseen dynamic contexts. First, the LWPR training is done with some dynamic contexts (2,6, and 10 kg), second the testing is done with 1, 4, and 8 kg contexts. Furthermore, we tested the performance of the RAFEL architecture planning different desired trajectory obtained from Equation (5.17) by modifying its coefficients. We have demonstrated that the system is able to keep the desired performance thanks to the appropriate cerebellar and LF contributions together.
We have evaluated the system accuracy with a performance measure over a whole iteration of the planned trajectory: the nMSE in radians (Rad) between the desired joint angle Q_d and the actual joint angle Q obtained from the robot plant. The nMSE is defined as the MSE divided by the variance of the target data values.
All simulations were setup in the Matlab (v.R2008a) environment and we used the robotics toolbox [62] for Matlab. All experiments carried out for this work and the results obtained are described in the subsection 5.4.

5.4 Experiments and Results

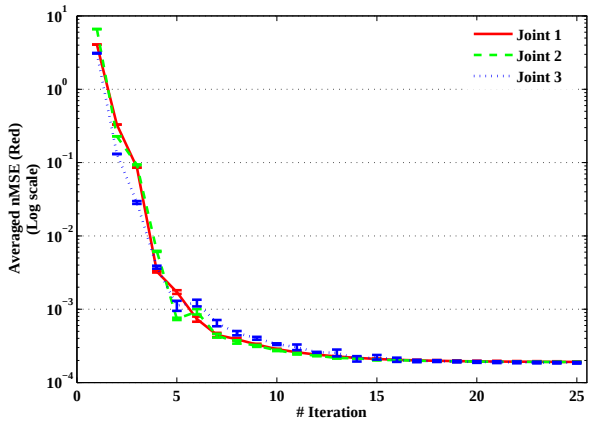
With the 3-DOF robot arm, the LWPR creates 22 locally linear models (RFs) for 2-6-8-10 kg contexts. Figure 5.3a shows the average of the nMSE over three joints; different traces indicate the response to different contexts during twenty-five iterations of the trajectory (5.17). Figure 5.3b shows the average of the nMSE over four contexts for the three robot joints. In this case, both panels in Fig. 5.3 prove that the RAFEL architecture is adaptable to dynamic transformations in the context of tracing a planned trajectory.
In the same way, Figs. 5.4a and 5.4b show that there are not remarkable variations of the averaged nMSE under kinematics transformations of the tip arm. These have been obtained shifting the orientation of the end-robot-segment by a factor $\lambda = [\pi/4, \pi/2]$ in radians. Then, the architecture robustness is not affected either by changes in dynamics or by changes in kinematics. Previous experiments are characterized by the fact that the LWPR learned the contexts separately. Anyway, the LWPR is capable of learning different forward dynamics models and retaining them in the same regression model. The figure 5.5 shows that after training forward dynamics models corresponding to the dynamics of the robot arm manipulating three different loads (2, 6, and 10 kg) at the

last robot segment, the LWPR was tested with three unseen loads of 1, 4, and 8 kg to study its generalization capability to predict the plant behavior under new contexts. The robot was expected to follow the trajectory (5.17) in all cases.

In order to highlight the advantages of the RAFEL architecture, we examined and compared how the tracking errors became compensated following the desired trajectory (5.17) using the three architectures. Apart from the RAFEL architecture, we considered a second one with the feedback loop (LF), but without cerebellar corrections (thicker solid line in Fig. 5.6) and a third one with the block C and a high-gain PD instead of the LF controller. In order to guarantee a low tracking error in the system with a PD controller, the gains used in the RAFEL architecture had to be multiplied by a factor of 500. Anyway, the solid line in Fig. 5.6 shows that the final behavior is not stable and the maximum torque applied at joints is 758 Nm which is dangerous in case of contact of the robot arm with the environment. Instead, from Fig. 5.6 we notice that the RAFEL scheme achieved a better performance and the maximum torque was limited to around 200 Nm.

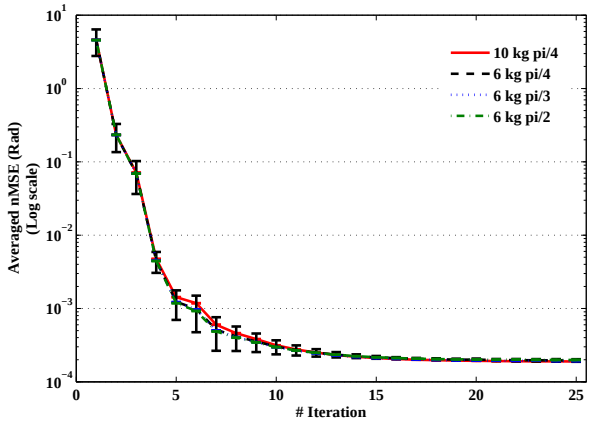


(a)

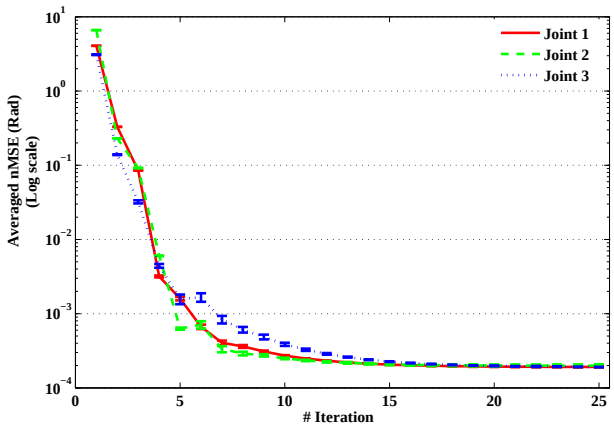


(b)

Figure 5.3: Control architecture tracking performance manipulating four objects (simulated as mass points) at the end of the last robot segment. Fig. 5.3a displays the nMSE averaged over three joints for four contexts. For the sake of clarity, error bars are plotted only for the 6 kg context and indicate the standard deviation between the joints. Fig. 5.3b shows the error in each joint averaged among the context experiments. Here, error bars indicate the standard deviation between the contexts.



(a)



(b)

Figure 5.4: Robustness of the RAFEL architecture under kinematics changes and with a different load added at the last robot segment. The lines in Fig. 5.4a represent the averaged nMSE performance among the joints during twenty-five iterations of the desired trajectory. For the sake of clarity, error bars, indicating the standard deviation between the joints, are plotted only for the 6 kg and $\pi/4$ radians context, that is representative of all the other contexts tested. Fig. 5.4b shows the averaged nMSE of the three joints over the contexts. Here, error bars indicate the standard deviation between the contexts.

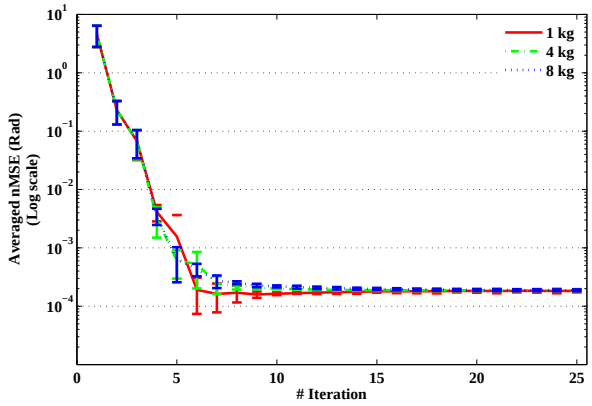


Figure 5.5: The RAFEL architecture still has a good performance under testing unseen dynamics contexts. This is expressed by the nMSE value plotted in the figure. Error bars represent the standard deviation between three joints.

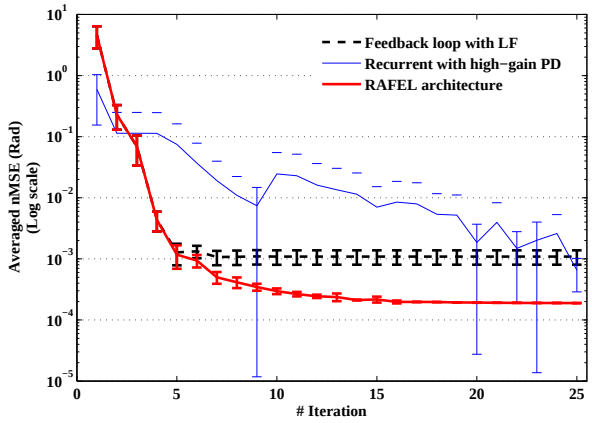
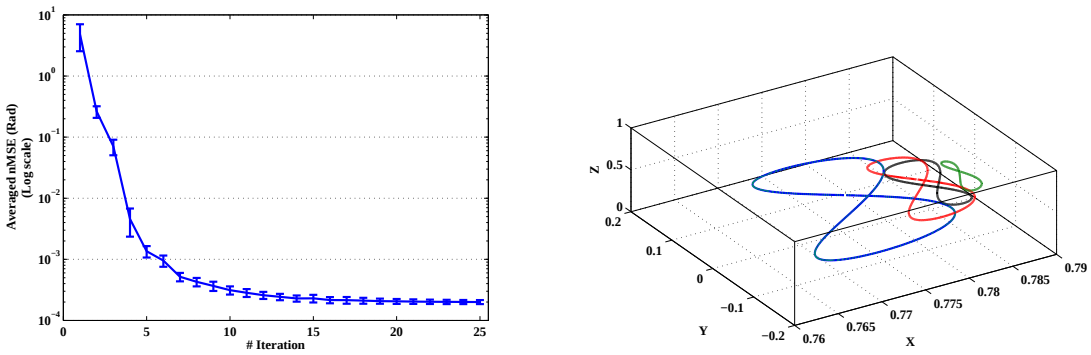


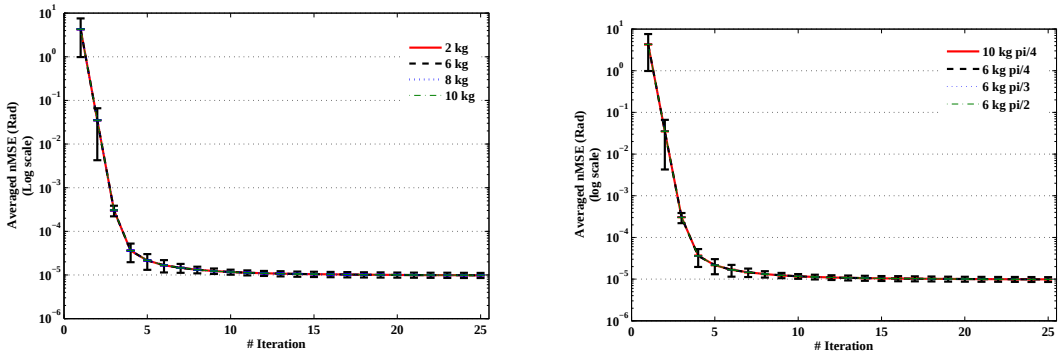
Figure 5.6: Comparison of three control architecture tracking performances manipulating 6 kg object in the last segment. Lines display the nMSE averaged over three joints, error bars indicate the standard deviation between the joints. The case of a PD with low gains is not plotted because it leads to a very bad performance (which lays out of range in this plot).

We also tested other trajectories obtained from Eq. (5.17) by changing the phase or the amplitude or summing two of them. In more details, we composed four trajectories with the following coefficients indicated in table 4.3. The performance results of tracking four precomputed trajectories under manipulation of 6 kg load in the last robot segment are shown in Fig. 5.7a and their eight-like figure-shape are plotted in Fig. 5.7b which correspond to the final iteration number 25.
The RAFEL architecture has also been tested with a higher number of DOFs and results in Fig. 5.8a indicates that the system's behavior becomes stable after few iterations of the desired trajectory (5.18). The LWPR creates 25 locally linear models (RFs) for each learned context. In order to highlight the important role of the cerebellum in driving all the joints to converge at a similar error value of nMSE we compared the RAFEL architecture performance with a similar one without the block C applying sensory corrections to the desired trajectory. Results are shown in the Fig. 5.9.



(a) **(b)**

Figure 5.7: 5.7a. The RAFEL architecture still has a good performance when performing the test stage with trajectories whose coefficients are defined in table 4.3 in Chapter 4. The robot arm manipulated 6 kg load at the end-effector-segment. The line indicates average nMSE value, averaged over the three joints first for each trajectory and then over the four trajectories. Error bars represent the standard deviation above and below the mean of the nMSE of the four trajectories. 5.7b. The eight-like figure-shapes obtained after 25 trials for the four precomputed trajectories (they are accurately approximating the desired trajectories).



(a) **(b)**

Figure 5.8: The RAFEL architecture still has a good performance with a robot arm of 7 DOFs when its dynamics are modified by manipulating different loads at the last end-effector as shown in Fig. 5.8a and under kinematics transformations as shown in Fig. 5.8b.

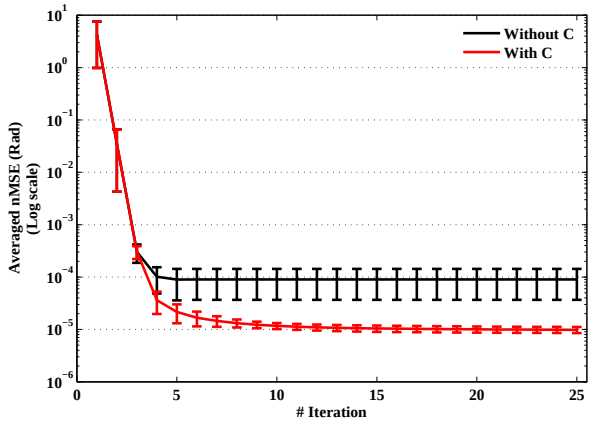


Figure 5.9: Lines represent the nMSE averaged over 7 joints after having being averaged over the four dynamic contexts (2, 6, 8, and 10 kg). The solid line shows the tracking error performance obtained by removing the cerebellar block in the RAFEL scheme. Comparing them, it becomes clear that the cerebellum drives the system to achieve a better performance in terms of nMSE with a small deviation among joints. Error bars represent the standard deviation above and below the mean of the nMSE of the seven joints.

5.5 Discussion

We implemented an architecture for predictive adaptive control of robotic arm movements which takes advantage of the connection between a machine learning algorithm (LWPR) and the adaptive cerebellar microcircuit. The cerebellum embedded in the recurrent control loop provides sensory corrections to the desired trajectory to compensate for deviations in the dynamics or kinematics of a robotic arm. We exploited the LWPR characteristics, as an efficient functional approximation at a low computational cost, to acquire and simulate the outcomes of robotic plant as a movement is repeated. In other words, the LWPR works as an internal forward dynamic model abstraction according to the process of formation of internal models in the cerebellum. Furthermore, the LWPR provides optimal input representation to the Purkinje layer as its RF kernels correspond to the input stage to the cerebellar structure (granular and molecular layers). The importance of an efficient and clean contribution to the Purkinje layer has been evaluated recently [59, 65] and other authors, such as Schweighofer et al. [17] (2001) and Porrill and Dean [8] (2007) hypothesized that the cerebellar learning is facilitated by optimizing the choice of the centers and the basis of RFs at the granular layer. Further work may be done in this sense to get a better optimization of the input stage and obtain more accurate signals to the PCs for improving the accuracy of the cerebellum corrective output. This is highly important for robot plants of high number of DOFs because of the high dimensions of the input space.

Due to the fact that using an analytical method is not always possible in order to obtain a sufficiently accurate dynamics model which is needed for compliant robot control, we proposed the LF controller. This controller adapts the error-correcting feedback over consecutive iterations of the same task using very low gains. The LF controller supplies the motor commands to the LWPR during the learning process.

The obtained results demonstrate that the RAFEL architecture leads the joint to converge fast to a small tracking error taking advantages of the LWPR's capability to retain and generalize dynamics contexts. The system is robust against both dynamics and kinematics transformations even in non-linear plants with high number of DOFs.

6

Conclusion

This chapter summarizes the contributions presented in this thesis, presents a discussion of future work, lists the publications arising from the work performed, and describes the synergies that had come out from the group-collaborations in the context of the SENSOPAC project.

In this dissertation, we presented two models (AFEL and RAFEL schemes) for the motor control and motor learning of robotic arm movements in which machine learning and biologically inspired approaches co-exist and complement each other. These hybrid schemes based on a machine learning engine (LWPR) and a bio-inspired component (cerebellar module) efficiently uses the LWPR component to optimize the input space representation abstraction and also efficiently uses the cerebellar-like structure to integrate different input-driven contributions for obtaining accurate corrective commands. Both architectures use an adaptive learning module to feed the LWPR component, making this approach useful even for robotic plants for which the analytical dynamics or kinematics are only roughly known.

The presented AFEL model is based on inverse model learning control and it uses constrained torques, which make the approach appropriate for compliant movements. However, the RAFEL model is based on forward model learning control and on recurrent sensory corrections to the desired trajectory. They both provide highly accurate movement capabilities (even in the presence of disturbances of the initial dynamics and kinematics of the "robot+object" plant), i.e., low errors for all joints with the same range of other approaches using control strategies based on high gains. Finally, they both achieve even better results in plants of high DOFs and all the joints converge faster to the minimum error.

6.1 Future Research

The next step towards implementing bio-inspired autonomous control systems is to provide the architectures developed in Chapter 4 and Chapter 5 with more real human like features. Firstly, we can modify the version of feedforward and recurrent schemes presented in this thesis including the transmission delays present in sensorimotor pathways in a biological system. Then, we will address the problem of compensating these time delays in order to stabilize the feedback controller which is sensitive to intrinsic delays. One of the potential uses of forward models is to act as an internal feedback to overcome the time delay [41]. Forward models provide an estimate of the outcome of a motor command which can be used for the feedback control. Miall et al. [71] (1993) proposed a particular form of control strategy to isolate the feedback delays from the

control loop, known as a Smith predictor that combines the advantages of feedforward control with those of feedback control [41, 43]. In other words, future work will address the exploitation of the learned forward models to provide an estimate of the outcome of a motor command avoiding biological time delays in the motor plants. Then, we could combine the recurrent control loop for compensating deviations in the trajectory and forward model predictions towards a more biological plausible system. Furthermore, we aim to exploit the forward model prediction, other future extensions include the combination of both internal models towards building an hybrid scheme. In fact, it seems that inverse and forward models might operate in combination [16].

Another potential extension to this work is to embed a spike-based cerebellar model in the feedforward and recurrent control loop and to exploit the connection between it and the machine learning engine (LWPR). With relation to this, we can take the advantage of the adaptive cerebellar spiking model built by Luque et al. [37] (2011). Furthermore, as we commented in Chapter 4, a potential role of the granular and molecular layers in biologically plausible cerebellar models is to provide accurate signals to the PCs for improving the learning of the current model. These signals seem to influence the DCN synapses capability in order to consolidate the learning [59, 63, 64]. So, next goal will be to study implications of a detailed computational model of the granular and molecular layers in order to facilitate the cerebellar learning and to improve the system performance.

Finally, future work will address the evaluation of the bio-inspired architectures mentioned above by dynamic and kinematic context switching experiments, in order to figure out and understand the mechanisms behind the motor learning and control of the CNS in manipulating different objects.

6.2 General Scientific Framework

This scientific work has been done and partially funded by the European Project SENSOPAC (IST-028056). This has represented an excellent collaborative framework with diverse research groups at other European Universities and research institutions. The

presented work represents a contribution of the University of Granada within this SENSOPAC consortium. The collaboration with the University of Edinburgh had been essential for the development of the idea of joining machine learning and biological processes and for the full understanding of the LWPR algorithm in terms of optimization of parameters and input-output function of regression. Then, this project had required a multidisciplinary approach and knowledge, from the biological point of view to the automatic learning control for a robotic arm. In fact, we implemented two biologically plausible control loop architectures in which cerebellar internal models are learned in order to manipulate tools with a robotic arm. The main effort had been accomplishing the hybrid connection between a machine learning engine and a bio-inspired module to learn different non-linear dynamics and control high nonlinear systems under various contexts.

6.3 Publication of Results

In the framework of the Sensopac project (IST-028056), the research group at the University of Granada has been involved in two main tasks: the development of the spiking neuron computation model (implemented within EDLUT) and the application of neuroscientist findings in order to design biologically inspired control systems capable of carrying out manipulating tasks. The work has been evaluated in the framework of international conferences and scientific journals (with impact factor on the JCR). In short, the scientific production can be summarized as: two journal papers published, two journal papers accepted for publication, and three international conferences (IWANN 2007, IPCAT 2007, and CogSys 2008).

Journals

1. R. R. Carrillo, E. Ros, **S. Tolu**, T. Nieus, E. D'Angelo, Event-driven simulation of cerebellar granule cells, Biosystems 94 (1-2) (2008) 10–17

2. N. R. Luque, J. A. Garrido, R. R. Carrillo, **S. Tolu**, E. Ros, Adaptive Cerebellar Spiking Model embedded in the control loop: Context switching and robustness against noise, International Journal of Neural Systems 21 (5) (2011) 385–401

3. **S. Tolu**, M. Vanegas, N. R. Luque, J. A. Garrido, E. Ros, Bio-inspired Adaptive Feedback Error Learning Architecture For Motor Control, under final revision at Biological Cybernetics.

4. **S. Tolu**, M. Vanegas, R. Agís, R. R. Carrillo, A. Cañas, Dynamics model abstraction scheme using radial basis functions, accepted for publication in Journal of Control Science and Engineering.

International Conferences

1. **S. Tolu**, E. Ros, R. Agís, Bio-inspired control model for object manipulation by humanoid robots, in: International Work Conference on Artificial Neural Networks (IWANN 2007), San Sebastian, Spain, LNCS, vol. 4507, 813–820, 2007

2. R. R. Carrillo, E. Ros, **S. Tolu**, T. Nieus, D. E., Event-driven simulation of cerebellar granule cells, in: Seventh International Workshop on Information Processing in Cells and Tissues (IPCAT 2007), Oxford, 2007

3. R. R. Carrillo, J. Garrido, E. Ros, **S. Tolu**, C. Boucheny, O. J. M. Coenen, A real-time spiking cerebellum model for learning robot control, in: International Conference On Cognitive Systems (COGSYS 2008), Karlsruhe, Germany, 2008

6.4 Main Contributions

We now summarize the main contributions of the presented work:

- Implementation of the bio-inspired **AFEL** architecture for motor control of robotic movements. A machine learning approach (LWPR) works as an internal inverse model abstraction kernel. The cerebellum embedded in the feedforward control loop drives the LWPR learning engine to learn an optimized internal model, and it contributes to deliver finer and more effective corrections. The performance obtained by the **AFEL** scheme in terms of error after learning is remarkable compared to other approaches.

- Implementation of the bio-inspired Recurrent AFEL (**RAFEL**) architecture, in which the forward internal model learns the dynamic model of the robot arm and provides optimal pre-processed signals to the cerebellum. The cerebellum

embedded in the recurrent control loop provides sensory corrections to the desired trajectory to compensate for deviations in the dynamics or kinematics of a robotic arm. The LWPR works as an internal forward dynamic model abstraction engine according to the process of formation of internal models in the cerebellum.

- The presented models take advantage of the connection between the accurate regression method LWPR and a basic cerebellar structure. Therefore, this approach integrates a machine learning approach (LWPR) with a bio-inspired scheme (cerebellar like structure) in a complementary manner. In fact, the cerebellar module takes full advantage of the LWPR for efficiently abstracting the high dimensional input space. LWPR provides optimal input representation to the Purkinje layer in terms of neural resources (it adapts its neural resources incrementally and according to the input data structure).

- In both architectures, LWPR works as a memory consolidation module. However, the cerebellar module with its learning rule is focused on short term adaptation. Thus, in terms of short/long term learning by analogy with biological systems, our cerebellar module, receiving inputs from the LWPR receptive fields, represents the MF-GrC/interneurons-PC pathway for short time learning while the LWPR adaptation kernels represent the MF-DCN adaptive pathway which is responsible of long term learning (memory consolidation).

- Study and development of low-gains control schemes through the LF controller, which adapts the error-correcting feedback over consecutive iterations of the same task. Furthermore, the LF controller accurately guides the LWPR during the learning process using very low gains. Results show that the global architecture has a compliant performance which is suitable for robotic systems in human environments.

- An evaluation of the combination of feedback and feedforward estimates indicates that we obtain considerable advantages for robust online control. Furthermore, we ensured accuracy and enhanced the speed of learning by optimizing the choice of the centers and transforming to an optimal basis of receptive fields by means of using LWPR.

- Due to the large number of DOFs and the nonlinearities of the 7-DOFs human arm, an internal model of the arm's dynamics is an extremely complex mapping, thus it requires a large number of encoding states. Despite this, we demonstrated that we achieved very good performances with a small number of states (or GCs) for a 7-DOf robot arm.

- Implementation of model abstraction schemes during object manipulation. The systems **AFEL** and **RAFEL** are robust against both dynamics and kinematics transformations.

7

Conclusiones en español

Este capítulo resume las contribuciones presentadas en esta tesis, presenta una discusión sobre el trabajo futuro, enumera las publicaciones que se obtienen del trabajo realizado y describe las sinergias resultantes de las colaboraciones entre grupos en el contexto del proyecto SENSOPAC.

En esta tesis hemos presentado dos modelos (los esquemas AFEL y RAFEL) para el control motor y el aprendizaje motor de los movimientos del brazo robótico en los que el control automático y los enfoques bio-inspirados coexisten y se complementan entre sí. Estos esquemas híbridos basados en un motor de aprendizaje automático (LWPR) y un componente bio-inspirado (módulo del cerebelo) utilizan eficientemente el componente LWPR para optimizar la abstracción de la representación del espacio de entrada y también utilizan eficientemente la pseudo estructura del cerebelo para integrar las distintas contribuciones de entradas para obtener órdenes correctivas precisas. Ambas arquitecturas utilizan un módulo de aprendizaje adaptativo para alimentar al componente LWPR, de modo que este enfoque es útil incluso para aquellas plantas robóticas cuyas dinámicas o cinemáticas analíticas son sólo parcialmente conocidas.

El modelo AFEL presentado se basa en el control del aprendizaje del modelo inverso y utiliza pares limitados que hacen que el enfoque sea adecuado para movimientos *dóciles*. Sin embargo, el modelo RAFEL se basa en el control del aprendizaje del modelo directo y en las correcciones sensoriales recurrentes a la trayectoria deseada. Ambos modelos proporcionan capacidades de movimiento de alta precisión (incluso en la presencia de alteraciones de las dinámicas y cinemáticas iniciales del conjunto "robot + objeto"), es decir, errores bajos para todas las articulaciones con el mismo rango de otros enfoques que utilizan estrategias de control basadas en altas ganancias. Finalmente, ambos logran resultados aún mejores en las plantas de muchos DOF y todas las articulaciones convergen más rápidamente al error mínimo.

7.1 Trabajo futuro

El siguiente paso hacia la implementación de sistemas de control autónomo bio-inspirados es proporcionar a las arquitecturas desarrolladas en los capítulos 4 y 5 más características reales humanas. En primer lugar, podemos modificar la versión de los sistemas hacia adelante y recurrente presentados en esta tesis incluyendo los retrasos de transmisión presentes en las vías sensorimotoras en un sistema biológico. Sucesivamente, vamos a abordar el problema de compensar estos retrasos de tiempo con el fin de estabilizar el controlador de retroalimentación que es sensible a los retrasos intrínsecos. Uno de

los usos potenciales de los modelos directos es actuar como una retroalimentación interna para superar el retardo de tiempo [41]. Los modelos directos proporcionan una estimación de la consecuencia debida a una orden motora que puede ser utilizada por el control de realimentación. Miall et al. [71] (1993) proponen una forma particular de estrategia de control para aislar los retrasos de retroalimentación desde el circuito de control, conocida como predictor de Smith, que combina las ventajas del control hacia adelante con las del control de retroalimentación [41, 43]. En otras palabras, el trabajo futuro abordará la explotación de los modelos directos para proporcionar una estimación de las consecuencias de una orden motora evitando los retrasos de tiempo biológicos de las entidades motoras. Sucesivamente, podemos combinar el lazo de control recurrente para compensar las desviaciones en la trayectoria y las predicciones de los modelos directos hacia un sistema biológico más plausible. Además de nuestro objetivo de explotación del modelo directo de predicción, otros trabajos futuros incluyen la combinación de ambos modelos internos en la construcción de un esquema híbrido. De hecho, parece que los modelos inversos y directos pueden trabajar en combinación [16].

Otro posible trabajo futuro sería integrar un modelo de cerebelo basado en impulsos (spikes) en el lazo de control hacia adelante y recurrente y explotar la conexión entre éste y el motor de aprendizaje de control automático (LWPR). En relación con esto, podemos aprovecharnos del modelo adaptativo cerebelar a impulsos desarrollado por Luque et al. [37] (2011).

Además, como hemos comentado en el capítulo 4, un papel potencial de las capas granular y molecular del cerebelo en los modelos biológicamente plausibles es proporcionar señales precisas a las PCs para mejorar el aprendizaje del modelo actual. Estas señales parecen influir en la capacidad de las sinapsis del DCN con el fin de consolidar el aprendizaje [59, 63, 64]. Por lo tanto, el siguiente objetivo será estudiar las implicaciones de un modelo detallado de cálculo de las capas granular y molecular con el fin de facilitar el aprendizaje del cerebelo y mejorar el rendimiento del sistema.

Finalmente, el trabajo futuro abordará la evaluación de las arquitecturas bio-inspiradas mencionadas anteriormente a través de experimentos de cambio de contextos dinámicos y cinemáticos, con el fin de descubrir y entender los mecanismos que subyacen al aprendizaje motor y al control del CNS en la manipulación de diferentes objetos.

7.2 Marco científico general

Este trabajo científico ha sido realizado y parcialmente financiado por el proyecto europeo SENSOPAC (IST-028056). Esto ha representado un excelente marco de colaboración con diversos grupos de investigación de otras universidades e instituciones de investigación europeas. El trabajo presentado constituye un aporte de la Universidad de Granada dentro de este consorcio SENSOPAC. La colaboración con la Universidad de Edimburgo ha sido fundamental para el desarrollo de la idea de unir el control automático con los procesos biológicos y para comprender plenamente el algoritmo LWPR en términos de optimización de los parámetros y de las entradas y salidas de la función de regresión. Este proyecto ha requerido un enfoque multidisciplinario y el conocimiento desde el punto de vista biológico hasta el control de aprendizaje automático de un brazo robótico. De hecho, hemos implementado dos arquitecturas de control biológicamente plausibles en las que los modelos internos del cerebelo se aprenden con el fin de manipular herramientas con un brazo robótico. El esfuerzo principal ha sido llevar a cabo la conexión híbrida entre un motor de aprendizaje automático y un módulo bio-inspirado para aprender diferentes dinámicas no lineales y controlar sistemas altamente no lineales bajo varios contextos.

7.3 Publicación de resultados

En el marco del proyecto SENSOPAC (IST-028056), el grupo de investigación de la Universidad de Granada ha participado en dos tareas principales: el desarrollo del entorno de computación de neuronas de spikes (implementado en EDLUT) y la aplicación de los resultados neurocientíficos con el fin de diseñar sistemas de control bio-inspirados capaces de llevar a cabo tareas de manipulación. El trabajo se ha evaluado en el marco de conferencias internacionales y revistas científicas (con factor de impacto en el JCR). En resumidas cuentas, la producción científica se puede enumerar como: dos artículos publicados en revistas, dos artículos aceptados para su publicación en revistas, y tres conferencias internacionales (IWANN 2007, IPCAT 2007, y CogSys 2008).

Revistas internacionales

1. R. R. Carrillo, E. Ros, **S. Tolu**, T. Nieus, E. D'Angelo, Event-driven simulation of cerebellar granule cells, Biosystems 94 (1-2) (2008) 10–17

2. N. R. Luque, J. A. Garrido, R. R. Carrillo, **S. Tolu**, E. Ros, Adaptive Cerebellar Spiking Model embedded in the control loop: Context switching and robustness against noise, International Journal of Neural Systems 21 (5) (2011) 385–401

3. **S. Tolu**, M. Vanegas, N. R. Luque, J. A. Garrido, E. Ros, Bio-inspired Adaptive Feedback Error Learning Architecture For Motor Control, revisión final en la revista Biological Cybernetics.

4. **S. Tolu**, M. Vanegas, R. Agís, R. R. Carrillo, A. Cañas, Dynamics model abstraction scheme using radial basis functions, aceptado para la publicación en la revista Control Science and Engineering.

Conferencias internacionales

1. **S. Tolu**, E. Ros, R. Agís, Bio-inspired control model for object manipulation by humanoid robots, in: International Work Conference on Artificial Neural Networks (IWANN 2007), San Sebastian, Spain, LNCS, vol. 4507, 813–820, 2007

2. R. R. Carrillo, E. Ros, **S. Tolu**, T. Nieus, D. E., Event-driven simulation of cerebellar granule cells, in: Seventh International Workshop on Information Processing in Cells and Tissues (IPCAT 2007), Oxford, 2007

3. R. R. Carrillo, J. Garrido, E. Ros, **S. Tolu**, C. Boucheny, O. J. M. Coenen, A real-time spiking cerebellum model for learning robot control, in: International Conference On Cognitive Systems (COGSYS 2008), Karlsruhe, Germany, 2008

7.4 Principales contribuciones

Ahora presentamos un resumen de las principales contribuciones del trabajo presentado:

- Implementación de la arquitectura bio-inspirada **AFEL** para el control motor de los movimientos robóticos. Un enfoque de aprendizaje automático (LWPR) funciona como un núcleo de abstracción del modelo interno inverso. El cerebelo incrustado en el lazo de control hacia adelante guía el motor de aprendizaje LWPR para aprender un modelo interno optimizado, que contribuye a ofrecer correcciones mejores y más efectivas. El rendimiento obtenido por el esquema **AFEL** en términos de error al final del aprendizaje es notable en comparación con otros enfoques.

- Aplicación de la arquitectura bio-inspirada AFEL recurrente (**RAFEL**), en la que el modelo interno directo aprende el modelo dinámico del brazo robótico y proporciona señales pre-procesadas óptimas en el cerebelo. El cerebelo incrustado en el lazo de control recurrente proporciona correcciones sensoriales a la trayectoria deseada para compensar las desviaciones en la dinámica o cinemática del brazo robótico. El LWPR funciona como un motor de abstracción del modelo dinámico interno de acuerdo con el proceso de formación de modelos internos en el cerebelo.

- Los modelos presentados se aprovechan de la conexión entre el método preciso de regresión LWPR y una estructura básica del cerebelo. Por lo tanto, este enfoque integra un enfoque de aprendizaje automático (LWPR) con un sistema bio-inspirado (la pseudo estructura del cerebelo) de manera complementaria. De hecho, el módulo del cerebelo aprovecha al máximo el LWPR de modo eficaz para abstraer el espacio de entrada de muchas dimensiones. El LWPR ofrece una representación óptima de entradas a la capa de Purkinje en términos de recursos neuronales (adapta sus recursos neuronales de manera incremental de acuerdo con la estructura de los datos de entrada).

- En ambas arquitecturas, el LWPR funciona como un módulo de consolidación de la memoria. Sin embargo, el módulo del cerebelo con su regla de aprendizaje se centra en la adaptación a corto plazo. Por lo tanto, en términos de aprendizaje a corto plazo / largo plazo, por analogía con los sistemas biológicos, nuestro módulo del cerebelo, que recibe aportaciones de los campos receptivos del LWPR, representa el camino MF-GrC/interneurons-PC para el aprendizaje a corto plazo, mientras que los núcleos adaptativos del LWPR representan el camino adaptativo MF-DCN, que es responsable del aprendizaje a largo plazo (consolidación de la memoria).

- Estudio y desarrollo de esquemas de control de baja ganancia a través del controlador LF, que adapta la realimentación correctiva de errores durante iteraciones consecutivas de la misma tarea. Además, el controlador LF guía de forma precisa el LWPR durante el proceso de aprendizaje utilizando ganancias muy bajas. Los resultados muestran que la arquitectura global tiene un rendimiento *dócil*, que es apropiado para sistemas robóticos en ambientes humanos.

- Una evaluación de la combinación de la realimentación y las estimaciones hacia adelante indica que obtenemos importantes ventajas para el control robusto en línea. Además, aseguramos precisión y aumentamos la velocidad de aprendizaje mediante la optimización de la elección de los centros y la transformación de una base óptima de los campos receptivos mediante el uso del LWPR.

- Debido al número elevado de DOFs y a las no linealidades de los 7-DOFs del brazo humano, un modelo interno de las dinámicas de la mano es un mapa muy complejo, por lo que requiere un gran número de estados de codificación. A pesar de ello, hemos demostrado que hemos conseguido muy buenos resultados con un pequeño número de estados (o GC) por un brazo robótico de 7 DOF.

- Implementación de esquemas de abstracción de modelos durante la manipulación de objetos. Los sistemas **AFEL** y **RAFEL** son robustos ante transformaciones tanto en la dinámica como en la cinemática.

Appendix A

LF Controller Details

Substituting the Laplace transform of Equation (4.6) in the Laplace transform of Equation (4.7), we obtain Equation (A.1):

$$\hat{D}_{i(k+1)}(s) = \hat{D}_{ik}(s) + P(s)H(s)[D_i(s) - \hat{D}_{ik}(s)], \tag{A.1}$$

where

$$H(s) = \frac{1}{s^2 + K_{vi}(s) + K_{pi}}. \tag{A.2}$$

Substituting the following Expression (A.3) in Equation (A.1)

$$G(s) = 1 - P(s)H(s), \tag{A.3}$$

we obtain Equation (A.4)

$$\hat{D}_{i(k+1)}(s) = G(s)\hat{D}_{ik}(s) + \hat{D}(s)[1 - G(s)]. \tag{A.4}$$

Starting with D_{i0}, after k iterations we will get Equation (A.5):

$$\hat{D}_{i(k)}(s) = D_i(s) + AG^k(s), \tag{A.5}$$

where A is a constant. The convergence of the learning algorithm depends on the factor $G^k(s)$ in Equation (A.5); The convergence will occur if $G^k(s)$ approaches zero. In this case, $\hat{D}_{i(k)}(s) \rightarrow D_i(s)$, and Expression (4.5) will be true with its right side equal to zero. The inverse Laplace transform of $G^k(s)$ is defined by Equation (A.6)

$$g_k(t) = L^{-1}[G^k(s)]. \tag{A.6}$$

If we choose an appropriate filter $P(s)$, (4.8), we would obtain

$$lim_{k\rightarrow\infty} |h_k(t)| = 0, \tag{A.7}$$

with which the convergence is possible to guarantee.

Appendix B

Architectures Performance Comparison

The next four figures, Figures B.1 to B.4, show a comparison between the feedforward and the recurrent architectures about how they respond under dynamics or kinematics transformations at the last joint robot for the 3-DOF and the 7-DOF arm configuration (see Chapters 4 and 5, results section, for further details about experiments). We can notice that they have a similar behaviour, although the feedforward system is faster than the other one reaching the minimum nMSE value. Anyway, both architectures achieve a very good performance with a low standard deviation.

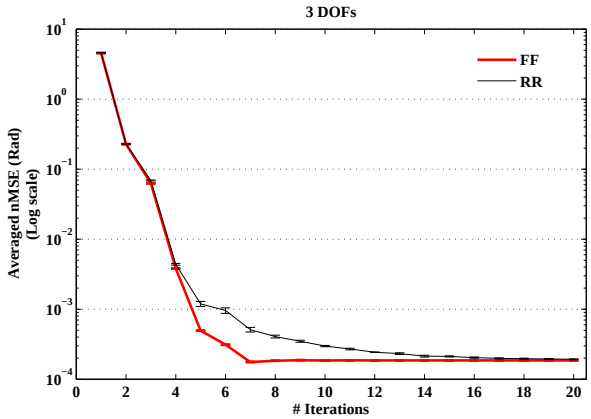


Figure B.1: Comparison between feedforward and recurrent architectures for the 3-DOF robot arm configuration. The nMSE is averaged among four dynamic contexts (2, 6, 8, and 10 kg) and the error bars represent the standard deviation among these contexts.

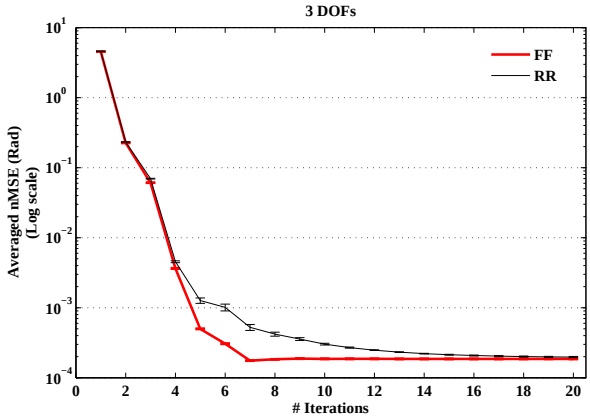


Figure B.2: Comparison between feedforward and recurrent architectures for the 3-DOF robot arm configuration. The nMSE is averaged among four kinematic contexts obtained shifting the orientation of the end-robot-segment by a factor $\lambda = [\pi/4, \pi/2]$ in radians (10 kg $\pi/4$, 6 kg $\pi/4$, 6 kg $\pi/3$, 6 kg $\pi/2$) and the error bars represent the standard deviation among these contexts.

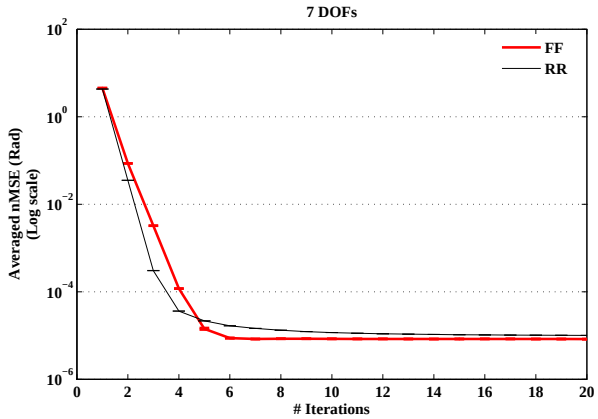


Figure B.3: Comparison between feedforward and recurrent architectures for the 7-DOF robot arm configuration. The nMSE is averaged among four dynamic contexts (2, 6, 8, and 10 kg) and the error bars represent the standard deviation among these contexts.

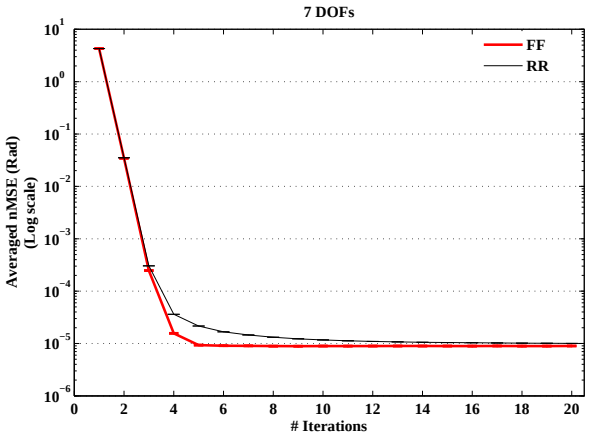


Figure B.4: Comparison between feedforward and recurrent architectures for the 7-DOF robot arm configuration. The nMSE is averaged among four kinematic contexts obtained shifting the orientation of the end-robot-segment by a factor $\lambda = [\pi/4, \pi/2]$ in radians (10 kg $\pi/4$, 6 kg $\pi/4$, 6 kg $\pi/3$, 6 kg $\pi/2$) and the error bars represent the standard deviation among these contexts.

Bibliography

[1] A. Albu-Schaffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimbock, G. Hirzinger, The DLR lightweight robot: design and control concepts for robots in human environments, Industrial Robot: An International Journal 34 (5) (2007) 376–385. xix, 38

[2] R. Shadmehr, S. P. Wise, The Computational Neurobiology of Reaching and Pointing : A Foundation for Motor Learning (Computational Neuroscience), The MIT Press, 2005. 3, 13

[3] M. Kawato, Internal models for motor control and trajectory planning, Current Opinion in Neurobiology 9 (6) (1999) 718–727. 3, 13, 26, 33

[4] D. Marr, A theory of cerebellar cortex, J. Physiol. 202 (1969) 437 – 470. 3, 14, 24, 25, 34

[5] J. S. Albus, A theory of cerebellar function, Mathematical Biosciences 10 (1-2) (1971) 25 – 61, ISSN 0025-5564. 3, 14, 24, 25, 34

[6] T. J. Sejnowski, Storing Covariance with Nonlinearly Interacting Neurons, Journal of Mathematical Biology 4 (1977) 303–321. 3, 14, 26, 42, 75

[7] M. Fujita, Adaptive filter model of the cerebellum, Biological Cybernetics 206 (3) (1982) 195–206, ISSN 0340-1200. 3, 14, 24, 43, 66

[8] J. Porrill, P. Dean, Recurrent Cerebellar Loops Simplify Adaptive Control of Redundant and Nonlinear Motor Systems, Neural Computation 19 (1) (2007) 170–193, ISSN 0899-7667. 3, 5, 6, 7, 8, 14, 16, 17, 18, 29, 32, 34, 36, 37, 40, 42, 61, 63, 66, 68, 75, 85

[9] A. Soylu, P. Causmaecker, P. Desmet, Context and Adaptivity in Pervasive Computing Environments: Links with Software Engineering and Ontological Engineering, Journal of Software 4 (9). 4, 14

[10] S. Vijayakumar, S. Schaal, Locally Weighted Projection Regression: Incremental Real Time Learning in High Dimensional Space, in: ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ISBN 1-55860-707-2, 1079–1086, 2000. 4, 15, 33, 67

[11] S. Vijayakumar, A. D'Souza, S. Schaal, Incremental Online Learning in High Dimensions, Neural Computation 17 (12) (2005) 2602–2634. 4, 15, 33, 34, 35, 58, 67, 69

[12] M. Kawato, Feedback-error-learning neural network for supervised motor learning, in: R. Eckmiller (Ed.), Advanced neural computers, Elsevier, North-Holland, 365–372, 1990. 5, 15, 25, 27, 66

[13] D. M. Wolpert, R. C. Miall, M. Kawato, Internal models in the cerebellum, Trends in Cognitive Sciences 2 (9) (1998) 338 – 347. 5, 15, 25, 26, 27, 32, 66, 74

[14] J. Porrill, P. Dean, J. Stone, Recurrent cerebellar architecture solves the motor-error problem, Proc Biol Sci 271 (1541) (2004) 789–96. 5, 15, 25, 29, 35

[15] P. Dean, J. Porrill, C. Ekerot, H. Jörntell, The cerebellar microcircuit as an adaptive filter: experimental and computational evidence, Nat Rev Neurosci 11 (1) (2010) 30–43. 5, 16, 25, 26, 27, 66

[16] M. Ito, Control of mental activities by internal models in the cerebellum., Nature reviews. Neuroscience 9 (4) (2008) 304–313. 6, 17, 26, 27, 28, 33, 37, 39, 72, 89, 97

[17] N. Schweighofer, K. Doya, F. Lay, Unsupervised learning of granule cell sparse codes enhances cerebellar adaptive control, Neuroscience 103 (2001) 35–50. 6, 17, 36, 37, 40, 61, 63, 85

[18] M. Kawato, Cerebellum: models, in: In Squire LR (Ed.), Encyclopedia of Neuroscience, 2, Elsevier Ltd., Oxford: Academic Press, 757–767, 2009. 7, 18

[19] A. Miyamura, H. Kimura, Stability of feedback error learning scheme, Systems and Control Letters 45 (4) (2002) 303 – 316. 7, 18, 24, 32

[20] P. Van der Smagt, Cerebellar Control of Robot Arms, Connection Science 10 (1998) 301–320. 7, 18, 32, 68

[21] H. Gomi, M. Kawato, Adaptive feedback control models of the vestibulocerebellum and spinocerebellum, Biological Cybernetics 68 (2) (1992) 105–114. 7, 18, 25, 27, 29, 32, 37, 66

[22] D. Nguyen-Tuong, J. Peters, Learning Robot Dynamics for Computed Torque Control Using Local Gaussian Processes Regression, in: Proceedings of the 2008 ECSIS Symposium on Learning and Adaptive Behaviors for Robotic Systems, IEEE Computer Society, Washington, DC, USA, 59–64, 2008. 7, 18, 32, 33, 34, 62, 68, 69

[23] M. I. Jordan, Computational Aspects of Motor Control and Motor Learning, in: Heuer, H. and Keele, S. (Ed.), Handbook of perception and action:Motor Skills, vol. 2, Academic Press, New York, 71–120, 1996. 7, 18, 32, 68

[24] M. I. Jordan, D. E. Rumelhart, Forward models: Supervised learning with a distal teacher, Cognitive Science 16 (1992) 307–354. 8, 18, 27, 28, 34, 40, 66

[25] CORDIS RTD-PROJECTS / European Communities, (SENSOPAC) SENSOrimotor structuring of perception and action for emerging cognition, http://cordis.europa.eu/fp7/ict/projects/home_en.html, Accessed 19 December 2011, 2011. 8, 19

[26] J. Garrido, Edlut oficial website, URL http://edlut.googlecode.com, Accessed 20 December 2011, 2009. 9, 20

[27] E. Ros, R. R. Carrillo, E. M. Ortigosa, B. Barbour, R. Agís, Event-Driven Simulation Scheme for Spiking Neural Networks Using Lookup Tables to Characterize Neuronal Dynamics, Neural Computation 18 (12) (2006) 2959–2993. 9, 20

[28] M. Kawato, K. Furukawa, R. Suzuki, A hierarchical neural network model for the control and learning of voluntary movement, Biological Cybernetics 57 (3) (1987) 169–185. 24, 25

[29] J. Nakanishi, S. Schaal, Feedback error learning and nonlinear adaptive control, Neural Networks 17 (10) (2004) 1453–1465, ISSN 0893-6080. 24, 27, 47, 49, 77

[30] M. Ito, Cerebellar circuitry as a neuronal machine., Prog Neurobiol 78 (3-5) (2006) 272–303. 24, 25, 26, 29

[31] M. Ito, Mechanisms of motor learning in the cerebellum, Brain Research 886 (1-2) (2000) 237–245. 25

[32] M. Ito, Historical Review of the Significance of the Cerebellum and the Role of Purkinje Cells in Motor Learning, Annals of the New York Academy of Sciences 978 (2002) 273Ű288. 25

[33] D. M. Wolpert, Computational approaches to motor control, Trends in Cognitive Sciences 1 (6) (1997) 209 – 216. 25, 74

[34] N. Ramnani, The primate cortico-cerebellar system: anatomy and function, Nature Reviews Neuroscience 7 (7) (2006) 511–522. 25

[35] N. R. Luque, J. A. Garrido, R. R. Carrillo, O. J. M. D. Coenen, E. Ros, Cerebellar-like Corrective Model Inference Engine for Manipulation Tasks, IEEE Transactions on System Man and Cybernetics B 41 (5) (2011) 1299 – 1312. 25, 66

[36] N. R. Luque, J. A. Garrido, R. R. Carrillo, O. J. M. D. Coenen, E. Ros, Cerebellar Input Configuration Toward Object Model Abstraction in Manipulation Tasks, IEEE Transactions on Neural Networks 22 (8) (2011) 1321–1328. 25, 66

[37] N. R. Luque, J. A. Garrido, R. R. Carrillo, **S. Tolu**, E. Ros, Adaptive Cerebellar Spiking Model embedded in the control loop: Context switching and robustness against noise, International Journal of Neural Systems 21 (5) (2011) 385–401. 25, 27, 66, 89, 97

[38] E. R. Kandel, J. Schwartz, T. Jessell, Principles of Neuroscience, McGraw-Hill Companies, New York, 4th edn., ISBN 0838577016, 2000. 26

[39] R. Apps, M. Garwicz, Anatomical and physiological foundations of cerebellar information processing, Nature Reviews Neuroscience 6 (4) (2005) 297–311. 26

[40] N. Schweighofer, M. A. Arbib, M. Kawato, Role of the cerebellum in reaching movements in humans. I. Distributed inverse dynamics control, European Journal of Neuroscience 10 (1998) 86–94. 26

[41] R. C. Miall, D. M. Wolpert, Forward models for physiological motor control, Neural Netw. 9 (1996) 1265–1279. 26, 28, 88, 89, 97

[42] S. Schaal, N. Schweighofer, Computational motor control in humans and robots, Current Opinion in Neurobiology 15 (6) (2005) 675–82. 27

[43] B. Mehta, S. Schaal, Forward models in visuomotor control, Journal of Neurophysiology (2002) 942–953. 27, 89, 97

[44] D. Mitrovic, S. Klanke, S. Vijayakumar, Adaptive Optimal Control for Redundantly Actuated Arms, in: Proceedings of the 10th international conference on Simulation of Adaptive Behavior: From Animals to Animats, Springer-Verlag, 93–102, 2008. 27, 70

[45] L. Lonini, L. Dipietro, L. Zollo, E. Guglielmelli, H. I. Krebs, An internal model for acquisition and retention of motor learning during arm reaching, Neural Computation 21 (7) (2009) 2009 – 2027, ISSN 0899-7667. 27, 33, 34, 62, 69

[46] R. J. Flanagan, P. Vetter, R. S. Johansson, D. M. Wolpert, Prediction Precedes Control in Motor Learning, Current Biology 13 (2) (2003) 146–150. 27

[47] S. Vaziri, J. Diedrichsen, R. Shadmehr, Why Does the Brain Predict Sensory Consequences of Oculomotor Commands? Optimal Integration of the Predicted and the Actual Sensory Feedback, J. Neurosci. 26 (16) (2006) 4188–4197. 27

[48] D. M. Wolpert, Z. Ghahramani, M. I. Jordan, An internal model for sensorimotor integration, Science 269 (1995) 1880–1882. 27

[49] H. Lalazar, E. Vaadia, Neural basis of sensorimotor learning: modifying internal models, Current Opinion in Neurobiology 18 (6) (2008) 573–581. 28

[50] A. Haith, S. Vijayakumar, Implications of different classes of sensorimotor disturbance for cerebellar-based motor learning models, Biological Cybernetics 100 (1) (2009) 81–95. 29, 37, 66

[51] P. Van der Smagt, F. Groen, S. K., Analysis and control of a rubbertuator arm, Biological Cybernetics 75 (5) (1996) 433–440. 32

[52] G. Hirzinger, J. Butterfaß, M. Fischer, M. Grebenstein, M. Hähnle, H. Liu, I. Schäfer, N. Sporer, A Mechatronics Approach to the Design of Light-Weight Arms and Multifingered Hands, in: ICRA, 46–54, 2000. 32, 38, 68

[53] German Aerospace Center, DLR Light-Weight Robot (LWR), http://www.dlr.de/rm/en/desktopdefault.aspx/tabid-3803/6175_read-8963/, Accessed 06 February 2012, 2011. 32, 38, 68

[54] S. Schaal, C. G. Atkeson, S. Vijayakumar, Scalable Techniques from Nonparametric Statistics for Real Time Robot Learning, Applied Intelligence 17 (1) (2002) 49 – 60, ISSN 0924-669X. 34, 69

[55] C. K. I. Williams, C. E. Rasmussen, Gaussian Processes for Regression, in: Advances in Neural Information Processing Systems 8, MIT press, 514–520, 1996. 34, 69

[56] A. J. Smola, B. Schölkopf, A tutorial on support vector regression, Statistics and Computing 14 (3) (2004) 199–222, ISSN 0960-3174. 34, 69

[57] C. G. Atkeson, J. G. Hale, F. Pollick, M. Riley, S. Kotosaka, S. Schaul, T. Shibata, G. Tevatia, A. Ude, S. Vijayakumar, E. Kawato, M. Kawato, Using humanoid robots to study human behavior, Intelligent Systems and their Applications, IEEE 15 (4) (2000) 46 – 56, ISSN 1094-7167. 34, 69

[58] T. Yamazaki, S. Tanaka, The cerebellum as a liquid state machine, Neural Networks 20 (3) (2007) 290–297. 36, 67, 68

[59] P. Wulff, M. Schonewille, M. Renzi, L. Viltono, M. Sassoè-Pognetto, A. Badura, Z. Gao, F. E. Hoebeek, S. van Dorp, W. Wisden, M. Farrant, C. I. De Zeeuw, Synaptic inhibition of Purkinje cells mediates consolidation of vestibulo-cerebellar motor learning., Nature neuroscience 12 (8) (2009) 1042–1049, ISSN 1546-1726. 36, 61, 68, 85, 89, 97

[60] T. Shibata, S. Schaal, Biomimetic gaze stabilization based on feedback-error-learning with nonparametric regression networks, Neural Networks 14 (2) (2001) 201–216. 37

[61] J. J. Craig, Introduction to Robotics: Mechanics and Control, Pearson/Prentice Hall, Upper Saddle River, New Jersey, 3rd edn., 2005. 38, 71

[62] P. I. Corke, A robotics toolbox for MATLAB, IEEE Robotics and Automation Magazine 3 (1) (1996) 24–32. 44, 78

[63] P. Attwell, S. Cooke, C. Yeo, Cerebellar function in consolidation of a motor memory, Neuron 34 (6) (2002) 1011–1020. 61, 89, 97

[64] E. Boyden, A. Katoh, J. Raymond, Cerebellum-dependent learning: the role of multiple plasticity mechanisms, Neuroscience 27 (1) (2004) 581–609. 61, 89, 97

[65] T. Honda, T. Yamazaki, S. Tanaka, T. Nishino, A Possible Mechanism for Controlling Timing Representation in the Cerebellar Cortex, in: International Symposium on Neural Networks, 67–76, 2010. 61, 85

[66] N. Masuda, S. Amari, A computational study of synaptic mechanisms of partial memory transfer in cerebellar vestibulo-ocular-reflex learning, Journal of Computational Neuroscience 24 (2) (2008) 137–156. 61

[67] S. Haddadin, A. Albu-Schäffer, G. Hirzinger, Safe Physical Human-Robot Interaction: Measurements, Analysis and New Insights., in: M. Kaneko, Y. Nakamura (Eds.), ISRR, vol. 66 of *Springer Tracts in Advanced Robotics*, Springer, 395–407, 2007. 62

[68] R. R. Carrillo, E. Ros, C. Boucheny, O. J. M. D. Coenen, A real-time spiking cerebellum model for learning robot control, Biosystems 94 (1-2) (2008) 18–27. 66

[69] P. R. Davidson, D. M. Wolpert, Widespread access to predictive models in the motor system: a short review, J Neural Eng 2 (3) (2005) 313–319. 68, 74

[70] O. Sigaud, C. Salaun, V. Padois, On-line regression algorithms for learning mechanical models of robots: A survey, Robotics and Autonomous Systems 59 (12) (2011) 1115–1129. 69

[71] R. C. Miall, D. J. Weir, D. M. Wolpert, J. F. Stein, Is the cerebellum a smith predictor?, Journal of Motor Behavior 25 (3) (1993) 203–216. 88, 97

[72] R. R. Carrillo, E. Ros, **S. Tolu**, T. Nieus, E. D'Angelo, Event-driven simulation of cerebellar granule cells, Biosystems 94 (1-2) (2008) 10–17.

[73] **S. Tolu**, M. Vanegas, N. R. Luque, J. A. Garrido, E. Ros, Bio-inspired Adaptive Feedback Error Learning Architecture For Motor Control, under final revision at Biological Cybernetics. .

[74] **S. Tolu**, M. Vanegas, R. Agís, R. R. Carrillo, A. Cañas, Dynamics model abstraction scheme using radial basis functions, accepted for publication in Journal of Control Science and Engineering. .

[75] **S. Tolu**, E. Ros, R. Agís, Bio-inspired control model for object manipulation by humanoid robots, in: International Work Conference on Artificial Neural Networks (IWANN 2007), San Sebastian, Spain, LNCS, vol. 4507, 813–820, 2007.

[76] R. R. Carrillo, E. Ros, **S. Tolu**, T. Nieus, D. E., Event-driven simulation of cerebellar granule cells, in: Seventh International Workshop on Information Processing in Cells and Tissues (IPCAT 2007), Oxford, 2007.

[77] R. R. Carrillo, J. Garrido, E. Ros, **S. Tolu**, C. Boucheny, O. J. M. Coenen, A real-time spiking cerebellum model for learning robot control, in: International Conference On Cognitive Systems (COGSYS 2008), Karlsruhe, Germany, 2008.

[78] **S. Tolu**, M. Vanegas, N. R. Luque, J. A. Garrido, E. Ros, Bio-inspired Adaptive Feedback Error Learning Architecture For Motor Control, revisión final en la revista Biological Cybernetics. .

[79] **S. Tolu**, M. Vanegas, R. Agís, R. R. Carrillo, A. Cañas, Dynamics model abstraction scheme using radial basis functions, aceptado para la publicación en la revista Control Science and Engineering. .