Adaptive Sensorimotor Peripersonal Space Representation and Motor Learning for a Humanoid Robot

THÈSE N^{O} 4289 (2009)

PRÉSENTÉE LE 11 FÉVRIER 2009

À LA FACULTÉ SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE D'ALGORITHMES ET SYSTÈMES D'APPRENTISSAGE
PROGRAMME DOCTORAL EN SYSTÈMES DE PRODUCTION ET ROBOTIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Micha HERSCH

acceptée sur proposition du jury:

Prof. M.-O. Hongler, président du jury
Prof. A. Billard, directrice de thèse
Prof. H. Bleuler, rapporteur
Prof. T. Flash, rapporteur
Prof. J. Santos-Victor, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2009

Abstract

This thesis presents possible computational mechanisms by which a humanoid robot can develop a coherent representation of the space within its reach (its peripersonal space), and use it to control its movements. Those mechanisms are inspired by current theories of peripersonal space representation and motor control in humans, targeting a cross-fertilization between robotics on one side, and cognitive science on the other side. This research addresses the issue of adaptivity the sensorimotor level, at the control level and at the level of simple task learning.
First, this work considers the concept of body schema and suggests a computational translation of this concept, appropriate for controlling a humanoid robot. This model of the body schema is adaptive and evolves as a result of the robot sensory experience. It suggests new avenues for understanding various psychophysical and neuropsychological phenomenons of human peripersonal space representation such as adaptation to distorted vision and tool use, fake limbs experiments, body-part centered receptive fields, and multimodal neurons.
Second, it is shown how the motor modality can be added to the body schema. The suggested controller is inspired by the dynamical system theory of motor control and allows the robot to simultaneously and robustly control its limbs in joint angles space and in end-effector location space. This amounts to controlling the robot in both proprioceptive and visual modalities. This multimodal control can benefit from the advantages offered by each modality and is better than traditional robotic controllers in several respects. It offers a simple and elegant solution to the singularity and joint limit avoidance problems and can be seen as a generalization of the Damped Least Square approach to robot control. The controller exhibits several properties of human reaching movements, such as quasi-straight hand paths and bell-shaped velocity profiles and non-equifinality.
In a third step, the motor modalities is endowed with a statistical learning mechanism, based on Gaussian Mixture Models, that enables the humanoid to learn motor primitives from demonstrations. The robot is thus able to learn simple manipulation tasks and generalize them to various context, in a way that is robust to perturbations occurring during task execution.
In addition to simulation results, the whole model has been implemented and validated on two humanoid robots, the Hoap3 and the iCub, enabling them to learn their arm and head geometries, perform reaching movements, adapt to unknown tools, and visual distortions, and learn simple manipulation tasks in a smooth, robust and adaptive way. Finally, this work hints at possible computational interpretations of the concepts of body schema, motor perception and motor primitives.

Key words Developmental robotics, body schema learning, manipulator control, programming by demonstration,

Résumé

S'inspirant des théories actuelles de la représentation de l'espace et du contrôle moteur chez les humains, ce travail se propose d'étudier comment un robot humanoïde peut se forger une représentation cohérente de son espace peripersonnel (l'espace immédiat) afin de pouvoir y contrôler ses movements. Cette approche pluridisciplinaire vise d'une part à découvrir des algorithmes plus robustes pour le contrôle de robots, et d'autre part à examiner des hypothèses sur la cognition humaine à l'aune de leur pertinence pour la résolution (dans un contexte robotique) de problèmes d'ordre cognitif. L'accent est mis sur la robustesse et la plasticité au niveau sensori-moteur, ainsi qu'aux niveaux du contrôle moteur et de l'apprentissage de tâches simples.

Dans un premier temps, ce travail s'attache au concept de schéma corporel et en propose une définition mathématique utile pour le contrôle d'un robot. Le schéma corporel ainsi défini est plastique et se construit en fonction des perceptions sensorielles du robot. Il suggère des explications à plusieurs phénomènes perceptifs observés chez les humains, tels que l'adaptation visuelle, l'insertion d'outils dans le schéma corporel, les membres illusoires, les champs récepteurs centrés sur les membres, et les neurones multimodaux.

Ensuite, l'aspect moteur est ajouté au schéma corporel permettant ainsi au robot d'atteindre des points de l'espace peripersonnel avec sa main. Le contrôleur proposé s'inspire de la théorie des systèmes dynamiques pour le contrôle moteur et permet au robot de contrôler son mouvement à la fois dans l'espace de travail et dans l'espace des articulations. Autrement dit, le contrôle se fait de façon simultanée dans l'espace visuel et dans l'espace proprioceptif. Ce contrôle multimodal permet de tirer profit des propriétés de chacun de ces deux espaces, et est donc plus avantageux que les contrôleurs traditionnels, notamment par rapport aux singularités et aux limites des articulations. De plus, les mouvements générés partagent plusieurs caractéristiques avec les mouvements humains, telles que des trajectoires presque rectilignes, un profil de vitesse en forme de cloche, et la "non-équifinalité" du mouvement.

Enfin, un module d'apprentissage statistique complète ce contrôleur afin de permettre au robot d'apprendre à effectuer des tâches simples de façon robuste. En se basant sur des exemples de trajectoires fournis par un opérateur humain, le robot est ainsi capable d'apprendre une tâche et de la réaliser malgré des conditions initiales différentes et des perturbations affectant la tâche alors même que celle-ci est en cours d'exécution.

Ces algorithmes ont été implémentés sur deux robots humanoïdes, le Hoap3 et le iCub, leur permettant d'apprendre leur géométrie, de contrôler leur bras, de s'adapter à l'utilisation d'outils et d'apprendre des tâches de manipulation simples telles que saisir un objet et le mettre dans une boîte.

Mots clés Robotique développementale, apprentissage du schéma corporel, contrôle de manipulateur robotique, programmation par imitation.

Acknowledgment

Working on this thesis has been a great pleasure, as well as a formative and challenging experience. For this, I would like to warmly thank my thesis director, Prof. Aude Billard for accepting me into her young lab at EPFL. Her high expectations and her trust have driven me throughout those years. The support and freedom she granted me allowed me to spread my wings in the realm of research and feel good in it. She has given me countless opportunities to meet and discuss with very interesting people, and made this thesis the fascinating and wonderful experience I had dreamed of.
My thanks also go to the members of my thesis committee Prof. Bleuler, Flash and Santos-Victor for accepting to read the present thesis and providing me with their feedback, and especially the latter two for coming from abroad to attend the thesis oral presentation.
I would also like to thank all the people from the RobotCub consortium, which funded this work. The RobotCub European FP6 project was a wonderful and stimulating introduction to transdiscplinary research, and it has been a real pleasure to discuss and work with the RobotCub partners. In particular, I am thankful to Francesco Nori and Lorenzo Natale at the Italian Institute of Technology (IIT) in Genova for helping me implement my work on the iCub robot.
I am also indebted to all my colleagues at LASA, especially Eric, Sylvain and Florent who were there before me and helped me find my way in the lab, and, along with Basilio, provided me with their software. It has been a pleasure spending my time with all of them and sharing an office with Biljana.

I would probably not have gone into research without the education I got from my parents. So I take this opportunity to also thank my dad for instilling scientific curiosity into my mind from very early on, and my mom for making apparent the narrowing effects of the scientific goggles.
Finally, I must of course thank my wife Déborah for agreeing to embark with me on this whole journey and for supporting all its unforeseeable consequences, and it is probably not over yet. And our son Jonas who did not ask for anything, but has been kind enough to diligently remind me of my priorities in a very persuasive way.

Contents

Foreword

In his play *The Dybbuk* [Ansky, 1914], S. Ansky tells the story of the dead kabbalist Khonon, whose soul takes possession of the body of his beloved Leah on her wedding day, and makes her oppose the wedding. Unlike Ansky's play, this thesis is not about love, justice and suffering. Rather, this thesis focuses on the practical aspects and investigates possible mechanisms by which such a soul could gain control over this newly inhabited body.
In less esoteric and more realistic terms, I will explore computational processes by which it is possible to gain control of a robot. I will investigate how a computer "mind" can learn how to use the sensory and motor signals provided by its robotic body in order to control it, and make meaningful movements. Like the dybbuk wandering between the worlds of the living and of the dead, we will dwell "between two worlds", namely the worlds of the roboticists and the cognitive scientists. Although I formulate the problem and provide experimental results in a robotics framework, I will keep a view on the human motor control as a source of inspiration. And conversely, I intend to present a model that can serve as a source of inspiration for the understanding of humans' ability to control their body.

Before going into the core of our topic, I shall formulate and justify some choices that were made concerning the presentation and the writing of this thesis. The first of those choices is related to the intended reader of this document. This report is first and foremost a thesis report, meant to be read by fellow researchers, primarily by the thesis jury, people who are well versed in this topic. However, my ambition is to make this report also intelligible and interesting to readers unfamiliar to the topic and to the mathematical tools used in this work. Before entering the core of the thesis describing its contribution, I briefly survey other works performed in the field, in order to enable the reader to situate my work within contemporary research and understand how it differs (or is similar to) work performed by other researchers. In order to keep the size of this thesis within reasonable bounds, those reviews are not extensive. Their aim is not to provide a comprehensive overview of current research in those topics, but rather to indicate the general research directions that have been pursued and the works that bear the strongest connections to this thesis. For the reader wishing to know more, I refer to more complete and well-written reviews recently published in the field.
In order to improve the readability of the text, and not to overload it with mathematical formulas, while keeping this work self-contained, an appendix introduces all the standard mathematical tools used in this thesis. In the body of the thesis, I will refer to this as I use the tools described in the appendix. The

reader familiar with all this material will not need going through it, but it can serve as a useful reminder to some.
At the end of each chapter I present a scientific discussion of the presented method and results. Those discussions are sometimes extended to provide a broader perspective and emphasize the general ideas. When going back to the work of our predecessor scientists (as done in e.g. [Holland, 2003]), it is indeed often the general ideas, hypotheses and assumptions rather than the technicalities of the work that are of interest.

The rest of this thesis is structured as follows. Chapter 1 presents the scientific approach (or paradigm [Kuhn, 1962]) to which this work belongs, and in which it must be understood. Chapter 2 describes with more precision the problems addressed in this thesis, and provides an overview of the solutions presented in previous work. Chapter 3 describes an adaptive model of the body schema, which can also be viewed as a model of peripersonal space representation. Chapter 4 shows how the motor modality can be added to this model and Chapter 5 further extends it by endowing it with the ability to learn motor primitives from demonstrations. In robotics terms, Chapter 3 is about multimodal fusion, Chapter 4 deals with control and Chapter 5 addresses programming by demonstration. A general discussion is then presented in Chapter 6.

Chapter 1

Scientific Approach

> Triomphe de la pensée calculante, éclipse de la pensée méditative.
>
> Alain Finkielkraut

1.1 Introduction

Despite its pretension to full objectivity, science, when purposing to describe the world, is never this "view from nowhere" [Nagel, 1986], which could claim absolute truth. Rather, scientific theories are offspring of non-scientific views of the world, which emerge as a result of the cultural, social, political, and technological development of society [Kuhn, 1962]. These views are mostly tacitly accepted, along with the corresponding hypotheses and assumptions. Making these assumptions explicit is a necessary step for a full understanding of scientific theories, and for a possible questioning of those assumptions, which is essential to any scientific progress.
This is why it seems important, before reporting on the actual contribution of this thesis, to include this chapter which presents an attempt to describe the general "philosophical" and scientific context in which this thesis has blossomed. This is particularly relevant for this thesis due to the nature of its topic, which is intimately related to concepts such as cognition, intelligence, mind and brain. Those concepts have recently been the objects of heated and passionate debates among philosophers [Copeland, 1993, Dreyfus, 1992, Searle, 1980, Varela et al., 1991]. This chapter should help the reader understand this work, evaluate its relevance and formulate his critics.
In the following sections, I describe the scientific approach (or paradigm) adopted in the thesis, which has been called *embodied cognition*. I briefly retrace the evolution that has led to this paradigm and what are its main aims, assumptions, focuses, challenges and pitfalls. I then describe in more details the scope and objective of this thesis and how they fit in the embodied cognition research trend.

1.2 Symbolic versus embodied cognition

The mind has been the object of human attention and curiosity from very early on. The philosophical substrate on which artificial intelligence developed can be traced back to the revolution of the Enlightenment in the 17^{th} century. Starting from Descartes, a process of rationalization of the scientific knowledge took place, which marked the end of the Aristotelian physics. This revolution, which led, among others, to the Newtonian mechanics, did not leave the common conception of thought unscathed. A new research program was set out, aiming at understanding the rules of cognition. For example, in its "Investigation of the laws of thoughts" (1854), Boole explains his attempt "to investigate the fundamental laws of these operations of the mind by which reasoning is performed, to give expression to them in the symbolical language of calculus" [Boole, 1854, p.3]. This program was continued by Frege's "Begriffschrift, eine der arithmetischen nachgebildete Formelsprache des reines Denkens" and then by Hilbert's program of rationalization and axiomatization of logic and mathematics. According to the latter, the axiomatization of mathematics was not only intended to give it sounder a basis, but also "to describe the activity of our understanding, to make a protocol of the rules according to which our thinking actually proceeds" [Hilbert, 1928] Following the development of the Turing machine, and its subsequent materialization which gave birth to the computer, it became possible to actually implement the putative "rules of thoughts" in a man-made device. This aroused tremendous expectations, and in the sixties A.I. fathers seriously announced that within two decades, the computer would be able to do anything a human can do [Crevier, 1993].

However this enthusiasm was met with stubborn skepticism by some philosophers, most prominently by Hubert Dreyfus [Dreyfus, 1992] and John Searle [Searle, 1980]. Coming from the analytical philosophy school, Searle argued that computers, however sophisticated, are merely syntaxic devices, and as such will never be able to cope with meaning and semantics. According to him, intelligence and understanding cannot be reduced to symbol manipulation. This sets a gap between human and computer intelligence, which cannot be bridged as long as computers remain mere computers.
Grounding his criticism in a phenomenological tradition, Dreyfus argued that a mind is inconceivable without a body. Indeed, the body mediates all our experiences, and experiences ground intelligence and skill development. According to him, A.I. advocates were living in the same Cartesian state of mind, that 19^{th} and 20^{th} century science and philosophy had proven illusory.

In parallel to this evolution, another scientific tradition developed, which also tried to endow artifacts with some sort of intelligence. This tradition, which would eventually lead to the discipline coined cybernetics, took an opposite stance. Instead of focusing on high-level cognition and abstract symbol manipulation, its main topic of investigation was the design of electro-mechanical devices that would show "intelligent" behavior. Its favorite application was autonomous navigation. Starting in the beginning of the 20^{th} century, engineers created devices that exhibited tracking behavior such as Hammond's and Miessner's "electric dog" (1912) [Cordeschi, 2002] or Grey's tortoise in the late forties [Holland, 2003] up to the Webb's recent cockroach [Webb, 2002] and Ijspeert's

salamander [Ijspeert et al., 2007].
While "traditional" A.I. focused on high level cognition, emphasizing symbol manipulation, reasoning and inference, the cybernetics school had animal behavior in mind, emphasized the interaction of the agent with its environment, the integration of what we now call hardware and software, and the coupling of sensors and actuators.
During the heyday of A.I. in the sixties and seventies, the cybernetics school fell in oblivion. It did not seem so interesting nor rewarding to try to emulate a tortoise behavior, when one could make a computer do such advanced things as playing chess or proving theorems. However, in the beginning of the nineties, when the limits of symbolic A.I. became obvious to all, scientists found a renewed interest in low-level robot intelligence, especially under the influence of Rodney Brook, who called it "Cambrian intelligence" [Brooks, 1999] and gave a new momentum to this approach, giving it his own flavor behavior-based robotics. The encounter between the high level cognition of traditional A.I. with the low-level cognition of the post-cybernetics school, along with recent discoveries in cognitive science, lead to the appearance of the developmental robotics approach and the building of developing humanoid robots such as Cog [Brooks et al., 1999] at the MIT and the European Robotcub project [Sandini et al., 2004], as part of which this thesis was written.

1.3 Developmental robotics

Starting in the second half of the nineties, the embodied cognition approach began to be applied to more human-inspired cognition. In other words, researchers tried to extend the usage of embodied cognition to tasks more sophisticated than wheeled-robot navigation and tackle problems such as manipulation and communication. This attempt led to the appearance of *developmental robotics* [Weng, 1998, Weng et al., 2001] which tries to reproduce aspects of the human cognitive development in a robotic system. The aim of this new research field is two-pronged [Lungarella et al., 2003]. First it aims at building robots that "develop", i.e., that are constantly learning and can thus improve over time and be more autonomous and smarter than traditional robots. Its second objective is to provide new insights into current theories of cognition and development, by adopting the synthetic method [Cordeschi, 2006], i.e., understanding it by building it. The robot is then no more considered as an end, but rather as a tool, a platform on which cognitive theories can be tested and evaluated.
In the beginning, most research done in this field followed a developmental road-map and first tackled tasks performed by neonates such as gazing, pointing and reaching [Lungarella and Metta, 2003]. Robotic head typically learned to perform visual tracking [Andry et al., 2001], and arm-eye systems learned to point and reach at observed locations (see Section 2.4). Inspired by results from developmental psychology concerning neonate and infant imitation [Meltzoff and Moore, 1977, Nadel, 1986, Butterworth, 1999], attempts were made to further endow robots with imitative abilities [Billard, 2000, Andry. et al., 2002, Lopes and Santos-Victor, 2007]. Imitation was also investigated in terms of communication skills such as joint attention [Kozima and Yano, 2001], and simple language acquisition through imitation games was studied and implemented [Steels, 1998]. As a further step toward manipulation, object affordances were

learned as a result of physical interactions of the robot with these objects [Fitzpatrick et al., 2003, Montesano et al., 2008].
The cognitive theories underlying all those works are relatively disparate, but they reveal a strong influence of the Gibson's ecological theory of perception [Gibson, 1979], Varela's enactive approach to cognition [Varela et al., 1991], and the theory of motor perception [Fadiga et al., 2000] (see [Vernon et al., 2007] and the 198 references therein for more details). Those theories strongly emphasize the codependency of perception and action.

1.4 Potential difficulties

To complete the description of the scientific approach adopted in this thesis, I mention here the potential difficulties and pitfalls that this approach entails. It seems important to keep them in mind in order to be aware of its inherent limitations.
The first and probably the main difficulty lies in the "schizophrenic" posture adopted by the developmental roboticist. By trying to be at the same time a naturalist and an engineer, he may perform poorly in both jobs. On one hand, since he wants to emulate biology, he restricts himself as an engineer and may thus propose suboptimal solutions to his engineering problem. On the other hand, because he has an engineering problem in mind, which is never exactly the one encountered by nature, [1] he will be biased as a naturalist and will suggest explanations that will fit his own engineering setup. This is a major pitfall in the approach adapted here, and it seems hardly avoidable. Indeed, this tension between the naturalist's and the roboticist's postures runs through this whole report, as it ran through the whole unfolding and preparation of this work. In the following chapters, I try to draw the reader's attention where this tension is most present and where a compromise had to be made between the roboticist and the naturalist.
This pitfall should, however, not be sufficient ground to dismiss the developmental robotics approach. On the contrary, this tension is likely to provide new points of views and ways of thinking, that are complementary to traditional naturalist and engineering approaches.

Another point that should be kept in mind is the level of description *de facto* adopted by the developmental robotics approach. Because its methodology is to use robots controlled by computers, it implicitly assumes that the computational level is an adequate level of abstraction for the explanation of cognitive processes. Although it seems a reasonable assumption, this belief is clearly grounded in the fact that computers are the devices that achieve what we see as most similar to human cognition. However, the same way it now seems ridiculous to model cognition with wheels and bolts, as done in the times of automata, computers may turn out in the future to be the wrong tools. But since this can hardly be known before exploring this hypothesis, it should not deter us from our endeavor.

There are of course many differences between biological and artificial cognitive systems. Maybe the most significant one is related to their material

[1] assuming it is not meaningless to speak of nature encountering problems

substrate. While biological systems are made of self-organized organic material, robots are made of inert matter, controlled by a centralized computing unit. While this separation between inert hardware and clever software may evoke the Platonician and Cartesian mind-body dualism, it may not reflect what is actually happening in nature. Although it may be tempting to apply this dualism in biological system, this difference undoubtedly limits the explanatory power of a robot.

1.5 Objectives

Within a developmental robotics perspective, the aim of this thesis is to enable a humanoid robot to take control of its limbs, move them to given positions and learn simple goal-directed gestures. More precisely, the aim is for a robot having little knowledge about its body, to learn the relationship between its visual and proprioceptive modalities, and use it to reach visual targets. In a way, this amounts to building a coherent representation of "itself in space" from the regularities in the sensorimotor modalities.
This objective is addressed in three steps. Chapter 3 focuses on the relationship between proprioceptive and visual modalities, in robotics terms, on the direct kinematics function. As we shall see, this is tightly linked to the geometrical properties of the robot body and to what is known in the cognitive science literature as the body schema. Then in Chapter 4, the motor aspect is investigated. In other words, we shall see how this relationship can be used to actually control the limbs of the robot, enabling it to move its hand to desired locations, the same way humans easily do it. This amounts to adding a motor modality to the body schema. The suggested controller implements principles putatively similar to the ones controlling human reaching movements. In a third step, Chapter 5 extends this controller to enable the robot to learn movements somewhat more complex than plain point-to-point reaching. Learning is performed on the basis of movements demonstrated by humans.

Throughout this thesis, a particular attention has been paid to the issue of adaptivity. Adaptivity is a fundamental property of human and animal motor control, which in this respect largely outperforms robot control. The mechanisms underlying such adaptivity are still mostly unknown, although under growing scrutiny and their implementation is a major challenge of contemporary robotics. In this work, adaptivity is addressed at the level of the body schema (or sensorimotor relationship) by having a body schema that adapts to changes in the body shape (for example when using a tool) and poor sensor calibration. Adaptivity is also addressed at the level of control, by suggesting a controller that can adapt to changes in the environment and perturbations occurring during movement execution. At the level of movement learning, the same system can learn various tasks and adapt its movement to different initial configurations and to perturbations.

True to the developmental robotics approach, this thesis not only aims at designing an adaptive robot controller, it also aims at providing insights on the computational principles possibly underlying human control of movements and

perception of peripersonal space. Understanding those seemingly basic human capabilities may well be the prerequisite for an understanding of any further, more elaborate, human cognitive ability. This is why, in each chapter, the biological inspiration underpinning this work is presented in the beginning, and the relevance of this work for the understanding of human cognition and control is briefly discussed at the end.

The topics of motor control and peripersonal space representation are very broad, and it seems hardly possible to cover them rightly in the scope of a doctoral dissertation. In this thesis, only specific aspects are addressed, while others are left totally untouched. In particular, this thesis only considers the upper-body part of the humanoid and all the interesting questions related to locomotion and balance are ignored. Moreover, the only modalities available on the robot are the visual and proprioceptive/motor modalities. This is why touch, audition and inertial sensing are not explicitly integrated in our model, although this could be done at a later stage.

1.6 Historical perspective

Despite developmental robotics being a relatively recent approach, as it is less than two decades old, it is grounded in a philosophical tradition which is much older [Madinier, 1938]. Already in the 18^{th} century, the French philosopher Condillac suggested something somewhat similar to the developmental robotics approach in [de Condillac, 1754]. There, he imagined a statue which would initially have no senses, and to which one would progressively add a new sensory modality, smell, audition, taste, vision and touch. Condillac tried to imagine how such a statue would react and how it would come to develop some kind of intelligence. Although this was done in a very different context, this thought experiment also aimed at figuring out how knowledge emerges out of sensations, a program that has not lost its actuality. So the idea of taking a synthetic approach to this problem is far from new, and the problem itself has been one of the most recurring question in Western philosophy. Indeed, the controversy between symbolic and embodied artificial cognition is but a declension of the opposition between Plato's innate ideas and Aristotle's view of the soul as *tabula rasa*.

1.7 Summary

In this chapter, I have laid down the general scientific framework in which this thesis is to be read and understood. I have described the developmental robotics approach to artificial intelligence, and shown that this approach emerged as the synthesis of two antithetical conceptions of cognition, the abstract, symbolic and disembodied view of traditional A.I. on one hand, and the low-level sensorimotor view of cybernetics on the other hand. Consequently developmental robotics stresses the importance of the physical interaction between the agent and its environment, the tight coupling between the motor, perceptive and cognitive abilities of the agent, the learning of sensorimotor contingencies as a necessary step to abstract and higher level cognition.

We have also seen that developmental robotics aims at studying cognition as a computational phenomenon. It therefore looks at the two sides of the coin: one side is cognition in natural systems, where an analytic approach is mostly followed, and the other side is a cognition in artificial systems, where a synthetic approach is taken. In this thesis, this second approach is clearly dominant.
In particular, the objective of this thesis is to look for computational mechanisms and principles that are liable to shed light on possible ways for a robot to build a representation of its peripersonal space, combine information coming from different modalities, and control simple arm motions. The solutions suggested in the following chapters are inspired by the current knowledge of how this is performed in human beings. The insights they provide can in turn be useful for understanding human cognitive mechanisms.
Finally, we have highlighted the difficulties and pitfalls that are intrinsically linked to our methodology and the developmental robotics methodology in general. Those are linked to the discrepancy between the artificial and biological systems hardware, and even more generally to this dualism between hardware and software, which is the legacy of a philosophical and scientific tradition going back to the Greeks.

Chapter 2

Aspects of robot motor learning

On le questionne
Et tous les problèmes sont posés
Soudain le fou rire le prend
Et il efface tout

Jacques Prévert

2.1 Introduction

This work is not the first attempt to endow a robot with the ability to learn to control its movements. Similar attempts were made before using different methods, and are briefly presented here. The aim of this chapter is not to provide an exhaustive account of all this work. Rather it is to more precisely introduce the challenges addressed in this thesis and to give the reader an idea of the general flavor of previous methods. Ideally, it should enable the reader to have just enough information to understand how the work presented in this thesis differs or is similar to previous approaches. For the reader interested in more details, references to more complete reviews are also provided.
This chapter first addresses the issue of motor control learning, tackled in the Chapters 3 and 4 of this thesis. Then Section 2.5 briefly reviews goal-directed task learning, which is addressed in Chapter 5.

2.2 The manipulator control problem

The basic question here is the following:

> *How can a robot know what commands to send to its motors to move its end-effector to a particular location?*

In order to clarify this question, one must first specify what types of commands need to be sent to a motor. Standard electrical motors (the ones considered throughout this work), can be commanded either in torque, in velocity or in

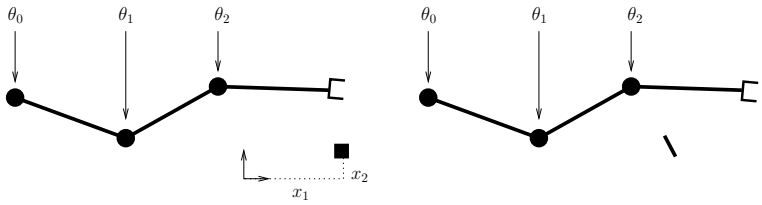


Figure 2.1: The control problem. What angles θ_0, θ_1 and θ_2 should be sent to the robot to enable it to reach a given position (the square on the left) or a given position and orientation (the stick on the right)?

position. Controlling the motor in torque implies that the resulting motion of the manipulator will depend on the specified torque, but also on other forces such as the gravitational force, the Coriolis force and the inertia of manipulator. Thus, the position, velocity, weight distribution and geometry of the manipulator play a role in the relationship between the motor command and its observed effect. On the other hand, when commanded in position or in velocity, the motor control board usually takes care of balancing the effects of the external forces and brings the limb to the desired position or velocity (typically using a PID control). This makes the control problem easier to handle. In this thesis, torque control is not considered, first to simplify the problem and second because the robots available for the experiments can only be controlled in position.
This basic question is illustrated in Fig. 2.1. If a command is sent to each of the motors, this will usually result in a displacement of the end-effector. The question is then, how to learn what commands are appropriate for bringing the end-effector to a desired location. If everything is known about the robot geometry there is no need for learning, and the problem can be solved analytically using the tools described in the next section.

2.3 Position control techniques

2.3.1 Setting and notations

We consider a robotic arm (or manipulator) with n degrees of freedom (DoF) manipulating objects in a m-dimensional space. The arm configuration and the end-effector location are given by vectors $\theta \in \mathbb{R}^n$ and $\mathbf{x} \in \mathbb{R}^m$ respectively. Those vectors may be indexed by the time t, indicating that at that moment in time the manipulator is in configuration θ_t and the end-effector in location $\mathbf{x}_t$. The fact that a particular arm configuration θ corresponds to a particular end-effector location $\mathbf{x}$ is expressed by the following relationship.

$$\mathbf{x} = \mathbf{K}(\theta), \tag{2.1}$$

where $\mathbf{K}$ is the kinematic function. This function is defined by the geometry of the robotic arm, and is assumed to be known. In our case, it is learned, as described in Chapter 3. Due to the redundancy of the manipulator, different values of θ may yield the same value $\mathbf{K}(\theta)$.
Note that the same formulation can be applied for $\mathbf{x}$ describing not solely the

end-effector location, but also its orientation. We now turn to the main problem of position control, namely inverse kinematics.

2.3.2 Local inverse kinematics

Problem description

Local inverse kinematics is often used to track a whole trajectory in Cartesian space. For reaching a particular point in space with the robot, a classical method is first to design a trajectory for the end-effector (for example a straight line) and iteratively find the joint velocities that will bring the end-effector from one point of the trajectory to the next. So the local inverse kinematics problem can be stated as follows. The manipulator is in a configuration θ, and one would like to know how to modify its joint angles so that its end-effector moves by a given (small amount) in a desired direction. Put differently, the aim is to find the joint angle speed $\dot{\theta}$ corresponding to a desired end-effector speed $\dot{\mathbf{x}}$. Expressed mathematically, one is looking for $\dot{\theta}$ such that

$$\mathbf{K}(\theta + \tau\dot{\theta}) = \mathbf{K}(\theta) + \tau\dot{\mathbf{x}}, \tag{2.2}$$

where τ is a time constant which tends to zero. Deriving with respect to time τ yields

$$\mathbf{J}(\theta)\dot{\theta} = \dot{\mathbf{x}} \quad \text{with} \quad \mathbf{J}(\theta) = \frac{\partial}{d\theta}\mathbf{K} \mid_{\theta} . \tag{2.3}$$

The matrix $\mathbf{J}(\theta)$ is called the Jacobian of the kinematic function $\mathbf{K}$. The above equation defines a plane which is tangent to $\mathbf{K}$ on θ, as is illustrated on Fig. 2.2. So the local inverse kinematics problem is simply to solve (2.3) with respect to $\dot{\theta}$. If the matrix $\mathbf{J}(\theta)$ is square and invertible, the solution is straightforward:

$$\dot{\theta} = \mathbf{J}(\theta)^{-1}\dot{\mathbf{x}}. \tag{2.4}$$

In this case, $n = m$ and (2.3) has a unique solution. However, if $\text{rank}(\mathbf{J}(\theta)) < n$, there is no solution to (2.3) and if $\text{rank}(\mathbf{J}(\theta)) > n$, there is an infinite number of solutions. In the latter case, all those solutions belong to an affine space, called the *solution space*, and the equation

$$\mathbf{J}(\theta)\dot{\theta} = \mathbf{0} \tag{2.5}$$

defines a vector space, called the *null space*. Adding any vector $\dot{\theta}_n$ of the null space to a vector $\dot{\theta}^d$ of the solution space yields a vector belonging to the solution space, as can be seen from the following:

$$\mathbf{J} \cdot (\dot{\theta}_n + \dot{\theta}^d) = \mathbf{J}\dot{\theta}_n + \mathbf{J}\dot{\theta}^d = \mathbf{0} + \dot{\mathbf{x}} = \dot{\mathbf{x}}. \tag{2.6}$$

In the above equation, and in the forthcoming ones, the dependency of $\mathbf{J}$ on θ is dropped in order to lighten the notation. One also defines the *orthogonal space* as the vector space orthogonal to the null space. A geometric representation of the null space, the solution space and the orthogonal space is provided in Fig. 2.3. The problem is to find, among all $\dot{\theta}$ belonging to the solution space, the best one depending on our needs. In the next sections, we briefly describe the most popular solutions to this problem (see [Sciavicco et al., 2000] for more details).

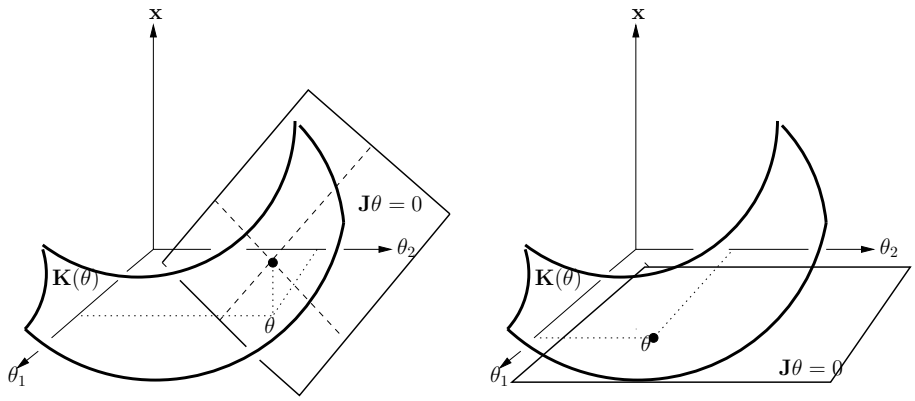


Figure 2.2: The kinematic function and a tangent plane. Left: the tangent plane is not horizontal. Right: the tangent is horizontal, i.e., orthogonal to $\mathbf{x}$. This is a singular configuration as there is no way (in the tangent plane) θ can be changed to change $\mathbf{x}$.

Known solutions

Least Norm (pseudo-inverse) method The first solution to this problem, suggested in the late sixties by [Whitney, 1969], is simply to take the smallest of the $\dot{\theta}^d$. This is in a way a "lazy" approach as one moves as little as possible to attain the desired end-effector velocity. In other words, the resulting position will be the one closest to the actual position θ, while lying on the solution space. This solution is obtained by solving

$$\min_{\dot{\theta}} \ \|\dot{\theta}\|^2 \quad \text{u.c.} \quad \mathbf{J}\dot{\theta} = \dot{\mathbf{x}}. \tag{2.7}$$

This is easily solved using the Lagrange multipliers technique, yielding

$$\dot{\theta} = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}\dot{\mathbf{x}} \doteq \mathbf{J}^\dagger\dot{\mathbf{x}}, \tag{2.8}$$

where $\mathbf{J}^\dagger$ is called the Moore-Penrose pseudo-inverse and satisfies $\mathbf{J}\mathbf{J}^\dagger = \mathbf{I}$, $\mathbf{I}$ being the identity matrix. As can be seen in Fig. 2.3, the resulting position is simply the orthogonal projection of θ on the solution space.

However, this solution is not always applicable, as it can happen that the matrix $\mathbf{J}\mathbf{J}^T$ is not invertible, i.e. singular. Those cases correspond to particular values of θ, i.e. to particular configurations of the manipulator, called the *singular configurations*. Those so-called *singularities* occur when the tangent plane is orthogonal to a subspace of the x space and thus there is a direction in which it is not possible to go, as illustrated in Fig 2.2, right.

Another issue that can arise with this method is the joint limits issue. This method contains no way to avoid jumping into the joint limits and is therefore likely to bring the joint angles outside of their allowed ranges. This must of course be avoided as it can damage the robot.

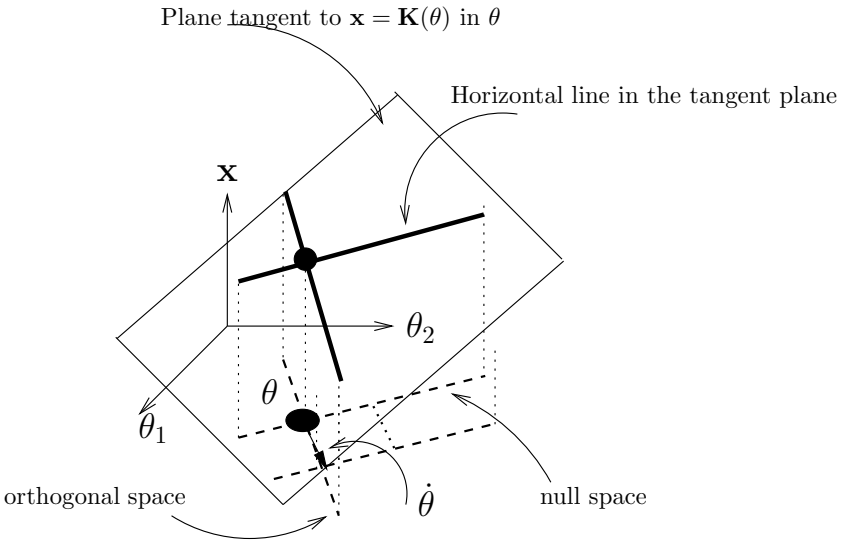


Figure 2.3: The Least Norm method illustrated for $n = 2$, $m = 1$.

Weighted Least Norm method The Least Norm method has been extended to the Weighted Least Norm (WLN) method by its inventor [Whitney, 1972] in order to have the possibility to give more weights to some joints than to others. This can be the case if one wants to move particular joints as little as possible, while giving more freedom to other joints to move. This can be expressed by the following constrained minimization problem

$$\min_{\dot{\theta}} \ \dot{\theta}^T \mathbf{W}_\theta \dot{\theta} \quad \text{u.c.} \quad \mathbf{J}\dot{\theta} = \dot{\mathbf{x}}, \tag{2.9}$$

where $\mathbf{W}_\theta$ is a square diagonal matrix of size n. Numbers on the diagonal indicate the "cost" of moving the corresponding joint. This problem is also solved using the Lagrange multipliers technique, yielding

$$\dot{\theta} = \mathbf{W}_\theta \mathbf{J}^T (\mathbf{J}\mathbf{W}_\theta \mathbf{J}^T)^{-1} \dot{\mathbf{x}}. \tag{2.10}$$

This is quite equivalent to the Least Norm method in a "stretched" joint angle space, where the stretching factors along the joint angle dimensions are given by the square root of the corresponding elements of $\mathbf{W}_\theta$. As such it is as vulnerable as the Least Norm method to singularities. However, this can be used to avoid joint limits by giving more weights to joints that are getting close to the limit of their allowed range [Chan and Dubey, 1995].

Gradient Projection method This method has been suggested in the seventies by Liégeois [Liégeois, 1977] to address the joint limit avoidance problem. The idea is to take advantage of the null space of the Jacobian to optimize a cost function which will bring the joints to the center of their range. This can be expressed by the following optimization problem:

$$\min_{\dot{\theta}} \ \dot{\theta}^T \dot{\theta} + \lambda H(\theta + \tau\dot{\theta}) \quad \text{u.c.} \quad \mathbf{J}\dot{\theta} = \dot{\mathbf{x}}, \tag{2.11}$$

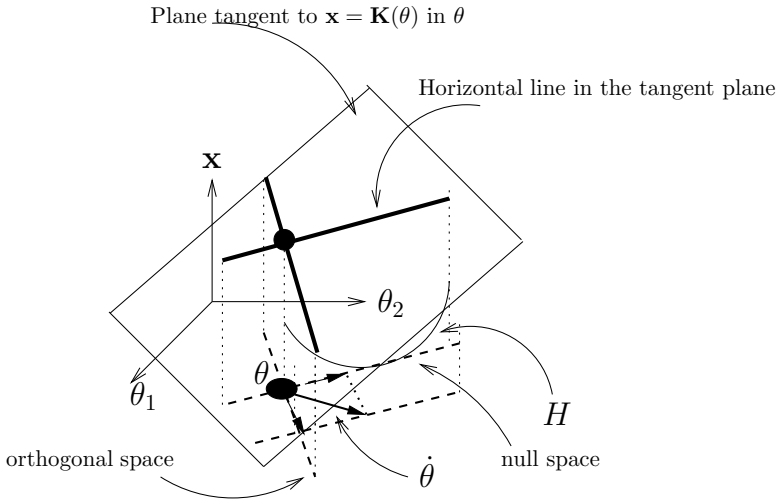


Figure 2.4: The Gradient Projection method illustrated for $n = 2$, $m = 1$.

where H is the function to optimize and $\lambda \in \mathbb{R}$ weighs the influence of the optimization function with respect to the joint velocity. The solution to this problem is given by:

$$\dot{\theta} = \mathbf{J}^{\dagger}\dot{\mathbf{x}} + \lambda(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})\nabla H, \tag{2.12}$$

where ∇ is the gradient operator. As can illustrated in Fig. 2.4, the first term of this last sum corresponds to the Least Norm solution and the second term is the projection of the gradient of H in the null space of $\mathbf{J}$ so that it does not influence $\mathbf{J}\dot{\theta}$. If H is used for joint limit avoidance, a typical expression for it is

$$H(\theta) = \frac{1}{n}\sum \left(\frac{\theta_i - \bar{\theta}_i}{\theta_i^M - \bar{\theta}_i}\right)^2 \quad \text{with} \quad \bar{\theta}_i = \frac{1}{2}(\theta_i^m + \theta_i^M), \tag{2.13}$$

where θ_i^m, $\bar{\theta}_i$ and θ_i^M are respectively the lower boundary, mid-range and upper boundary of joint i.

Damped Least Squares method This method has been proposed independently and quite simultaneously by [Wampler, 1986] and [Nakamura and Hanafusa, 1986] to deal with singularities. As mentioned above, singularities occur when the robot is incapable to move in a certain direction due to the configuration of its joints. In this case, the constraint given in 2.7 cannot be met and the problem has no solution. Thus, an approximate solution is found by solving the following optimization problem:

$$\min_{\dot{\theta}} \|\mathbf{J}\dot{\theta} - \dot{\mathbf{x}}\|^2 + \lambda\|\dot{\theta}^2\| \quad \text{with} \quad \lambda > 0. \tag{2.14}$$

The solution is given by:

$$\dot{\theta} = (\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I_n})^{-1}\mathbf{J}^T\dot{\mathbf{x}} \tag{2.15}$$

A weighted expression of the Damped Least Squares (DLS) also exists, given by

$$\min_{\dot{\theta}} (\mathbf{J}\dot{\theta} - \dot{\mathbf{x}})^T \mathbf{W}_{\mathbf{x}} (\mathbf{J}\dot{\theta} - \dot{\mathbf{x}}) + \dot{\theta}^T \mathbf{W}_\theta \dot{\theta} \quad \Rightarrow \quad \dot{\theta} = (\mathbf{J}^T \mathbf{W}_{\mathbf{x}} \mathbf{J} + \mathbf{W}_\theta)^{-1} \mathbf{J}^T \mathbf{W}_{\mathbf{x}} \dot{\mathbf{x}}, \tag{2.16}$$

where $\mathbf{W}_{\mathbf{x}} \in \mathbb{R}^{m \times m}$ and $\mathbf{W}_\theta \in \mathbb{R}^{n \times n}$ are diagonal positive definite matrices. Various variations on this method were suggested and can be found in [Chiaverini et al., 1994].

Extended Jacobian method This method, suggested in [Baillieul, 1985] tries to avoid singularities by imposing additional constraints in task space. The idea is simply to extend the kinematic function $\mathbf{K}$ with an objective function $\mathbf{G}(\theta)$ of dimension $n - m$, which we want to optimize. This defines a n-dimensional kinematic function $\mathbf{K}_e = [\mathbf{K}^T \ \mathbf{G}^T]^T$ with a squared Jacobian matrix, which can then be inverted:

$$\dot{\theta} = \mathbf{J}_e^{-1} \begin{pmatrix} \dot{\mathbf{x}} \\ 0 \end{pmatrix} \quad \text{with} \quad \mathbf{J}_e = \begin{pmatrix} \frac{\partial}{d\theta}\mathbf{K} \\ \frac{\partial}{d\theta}\mathbf{G} \end{pmatrix}. \tag{2.17}$$

As an objective function to maximize, it was suggested to use the manipulability $\sqrt{\mathbf{J}^T\mathbf{J}}$.

2.3.3 Global inverse kinematics

Problem description

The global inverse kinematics problem consists simply in finding the inverse of $\mathbf{K}$. More precisely, given a target $\mathbf{x}_T$, the aim is to find the set $\mathbf{\Theta_T}$ of the manipulator's joint configurations that bring the end-effector to $\mathbf{x}_T$:

$$\mathbf{\Theta_T} = \{\theta | \mathbf{K}(\theta) = \mathbf{x}_T\}. \tag{2.18}$$

In the case of redundant manipulators, this set is usually a manifold of dimension $n-m$, which can be disjoint in case of joint limits constraints (see [Burdick, 1989] for more details on the structure of this manifold). Usually one does not need the entire manifold, but just one point belonging to it, so one settles for a single value $\theta_{\mathbf{T}} \in \mathbf{\Theta_T}$.

Known solutions

Closed-form solutions For a number of robots closed-form solutions to the global inverse kinematics problem can be found using geometrical considerations and solving trigonometric equations [Küçük and Bingül, 2004]. More relevant to us, a closed-form solution for the positioning of a 7-DoF anthropomorphic arm has been provided by [Tolani et al., 2000].

But of course, depending on the geometry of the manipulator, closed-form solutions can often not be derived analytically. In those cases, iterative methods can be used. Those methods start from an initial guess of the solution and iteratively update this guess to get closer to the real solution, hoping that one will eventually find a guess that is precise enough for our purpose. One

of the most common iterative method is the Cyclic Coordinate Descent (CCD) method, which is used in following chapters.

Cyclic Coordinate Descent method This method was originally proposed by [Wang and Chen, 1991] to solve the global inverse kinematics problem for a serial manipulator with rotative and translational joints. In the present description, I restrict myself to rotative joints, as translational joints are not used in this work. Let us first consider the case where $\mathbf{x}$ denotes the end-effector position only (and not the orientation). In this case the CCD method is fairly straightforward. Starting from an initial configuration, each joint angle is successively updated so as to minimize the distance between the end-effector position and the target, as depicted in Fig. 2.5. One starts from the distal joint, updates it and gets a new configuration. One then proceeds to the next joint, and down till the proximal joint, when one starts with the distal joint again. This goes on until the distance to the target reaches a given precision, or if no further update is possible, meaning that the algorithm failed to find a solution.
As mentioned above, the formula for updating the joint is obtained by minimizing the distance between the end-effector position and the target, which is equivalent to minimizing the angle between the end-effector and the target position vectors expressed in a FoR centered on joint i:

$$\Delta\theta_i = \max_{\theta_i} (\mathbf{x} - \mathbf{x}^i)^T (\mathbf{x}_T - \mathbf{x}^i), \tag{2.19}$$

where $\mathbf{x}^i$ is the positions of joint i, as illustrated in Fig 2.5. The solution is given by

$$\Delta\theta_i = \text{atan} \frac{(\mathbf{a}_i \times (\mathbf{x} - \mathbf{x}^i))^T (\mathbf{x}_T - \mathbf{x}^i)}{\mathbf{a}_i^T (\mathbf{x} - \mathbf{x}^i) \mathbf{a}_i^T (\mathbf{x}_T - \mathbf{x}^i)}, \tag{2.20}$$

where $\mathbf{a}_i$ is the rotation axis of joint i, making sure that it is a maximum (by adding π otherwise). When considering the end-effector orientation as well as position, the algorithm remains the same, only the update rule changes. It is obtained by minimizing a distance function that takes position and orientation into account. The update rule presented here differs slightly from the standard one. Using the representation described in Appendix A.2.1, we denote the target orientation by a 3-dimensional vector $\mathbf{b}_T$ and the end-effector orientation by a vector $\mathbf{b}$.

$$\Delta\theta_i = \min_{\theta_i} k_1 (\mathbf{x} - \mathbf{x}^i)^T (\mathbf{x}_T - \mathbf{x}^i) + k_2 \Big|\Big| (-\mathbf{b}_T) * \mathbf{b} \Big|\Big|^2, \tag{2.21}$$

where $*$ denotes the composition operator defined by (A.21) and $k_1, k_2 \in \mathbb{R}^+$ are two scalars accounting for the different units of position and orientation vectors. The solution, computed in Appendix B.2 is the given by

$$\Delta\theta_i = \text{atan} \frac{k_1 (\mathbf{a}_i \times (\mathbf{x} - \mathbf{x}^i))^T (\mathbf{x}_T - \mathbf{x}^i) - 2 k_2 \alpha_i \tilde{\mathbf{b}}_i^T \mathbf{a}_i}{k_1 \mathbf{a}_i^T (\mathbf{x} - \mathbf{x}^i) \mathbf{a}_i^T (\mathbf{x}_T - \mathbf{x}^i) + k_2 \big(1 - 2\tilde{\mathbf{b}}_i^2 + (\tilde{\mathbf{b}}_i \times \mathbf{a}_i)^2\big)}, \tag{2.22}$$

where $\tilde{\mathbf{b}}_i$ is the rotation given by the composition $\mathbf{b}_{i+1} * \cdots * \mathbf{b}_n * (-\mathbf{b}_T) * \mathbf{b}_1 * \cdots * \mathbf{b}_{i-1}$. The main drawback of the CCD method is that it is difficult to deal with joint limits. It can indeed happen that joint limits perturb the algorithm

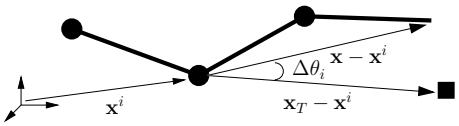


Figure 2.5: The CCD method. The square shows the position of the target

and prevent it from finding a solution although it exists. A second limitation of this method is that its convergence is rather slow at the end of the algorithm. This is why it was originally suggested in conjunction with another method, which takes over as CCD gets close enough to the target.

2.4 Previous approaches to learning manipulator control

2.4.1 The bio-inspired approach

Since controlling a robot can be done without learning, until recently learning a robot controller only raised marginal interest among scholars, and only among connectionists. Maybe the earliest attempt to tackle this problem is the Cerebellar Model Articulation Controller (CMAC) [Albus, 1975]. This work, which prefigures the developmental robotics approach, explicitly takes the brain as a source of inspiration to control a robotic manipulator. In those days (and to a somewhat lesser extend in our days), taking inspiration from neurobiology meant using artificial neural networks. The basic ideas in the CMAC is to learn an association between an input signal containing the joint position and desired end-effector location and an output signal containing the appropriate command. This association is learned in a supervised manner, i.e., someone has to give the system many examples of matched input and output signals. Using what will then be refered as receptive fields, the system in then able to perform some sort of interpolation between the example points and learn a control function. This approach has serious shortcomings, as it does not specify how the training examples are to be generated, nor handles the case of redundancy in the manipulator. However it first laid the motivational foundations for learning a robot controller, and argued for a bio-inspired approach to manipulator control. It thus exerted a widespread influence among connectionists and many extensions to the CMAC have been suggested (see [Jiang and Wang, 2003] and references therein).

During the revival of the connectionist approach, in the late eighties, some roboticists took up on Albuse's work and suggested different ways to learn to control a robotic arm with artificial neural networks. Among the most influential are the attempts by [Kawato et al., 1987], [Kuperstein, 1988] and [Mel, 1990]. The first one introduces idea of learning a model of the inverse dynamics, which computes the torque appropriate for generating a desired end-effector trajectory. This inverse dynamics model is combined with a simple position feedback controller, which enables the tracking of the trajectory even if the inverse dynamics model is not accurate. The data for training the model is acquired by

repeatedly (and slowly) tracking the same trajectory. In [Kuperstein, 1988], the idea of visuo-motor coordination is introduced, i.e., the target is not given in an objective "task space", but rather in a visual space. Control is achieved by learning the correlation between visual input on one hand, and eyes and arm postural sensory input on the other hand. Training data is provided by a random exploration of the workspace. A similar approach is followed in [Mel, 1990], but the system also learns the Jacobian as a mapping between joint configuration, joint velocity and visual end-effector velocity.
In the following years, countless variations on those models were suggested, which are reviewed by [Katić and Vukobratović, 1994]. Later work include [Metta et al., 1999], where an articulated head-eye system is used to fixate the hand using visual servoing. A mapping is learned between the head and eye proprioceptive information and the adequate combination predefined torque fields. Fixating a target then generates the torque field that brings the end-effector to the target. Other authors suggested to learn the inverse kinematics using other function approximators such as Locally Weighted Regression [D'Souza et al., 2001], Self-Organizing Maps [Gaskett and Cheng, 2003], Radial Basis Functions [Meng and Lee, 2007] or Bayesian Networks [Sturm et al., 2008].

Extending the control problem to the tactile modalities, in [Fuke et al., 2007] a simulated humanoid robot learns to reach to felt but not seen spots of its face. Using a single tactile sensor on the hand, in [Nabeshima et al., 2006] a robot learns how to control a tool.

2.4.2 The self-calibration approach

Another approach for learning to control an arm, is the self-calibration approach. Unlike the approaches described above, self-calibration usually assumes some parametrized structure of the manipulator. The self-calibration approach aims at finding the appropriate parameters for a given structure. Standard tool, such as those described in Section 2.3 are then used to control the robot. Self-calibration differs from standard calibration in that it does not require the active intervention of a human.
Self-calibration of a kinematic chain for a hand-eye system was first suggested in [Bennett et al., 1991]. There, the chain is modeled using the DH parameters, and a pinhole camera model. A number of positions are recorded along with the corresponding positions on the camera images. A gradient descent is then performed on the DH parameters, using a least squares cost function. The initial guess of the parameters is assumed to be close enough from the actual parameters, to ensure convergence. Other method have been suggested, most of them requiring first the visit of a number of configurations and then the optimization of the kinematic parameters using some iterative non-linear optimization method [Wampler et al., 1995, Iurascu and Park, 2003, Gatla et al., 2007].
In [Bennett et al., 1992], the Jacobian in a given configuration is directly estimated by measuring the effect of different sets of torques or velocities on the end-effector. The kinematic parameters of the chain are then retrieved from the Jacobian.
In all those previously cited works, calibration is considered as a process preliminary to the actual use of the robot, and requires first a sampling procedure, followed by an optimization procedure.

2.5 Learning simple movements

If learning how to move one's hand from one point to another is an essential aspect of motor learning, it is by far not the only one. Very often, it is useful to learn a whole gesture, i.e., also the characteristics of a movement trajectories for a particular task. In humans as well as in robots, this can be done by *exploration* and *imitation*. Exploration consists in trying out different possibilities, and finding the best possible solution for the execution of a given task. Imitation consists in trying to copy in some way the solution shown by some demonstrator who knows how to accomplish the task. The main difficulty in exploration is that the search space is usually high-dimensional and very large, so that it is often very hard to find an adequate solution in it. While imitation reduces the search space by focusing on the demonstrations, it brings another challenge, the so-called *correspondence problem* [Nehaniv and Dautenhahn, 2002]. Indeed, when imitating someone, one has to match his actions to our owns. This matching process can become very tricky if the imitator and the demonstrator have different bodies. For example, a child imitating an adult holding in an object with one hand, may have to use both hands if the object is heavy.
Although in humans and other animals, exploration and imitation are intertwined processes, in robotics people have usually used either imitation or exploration separately, but not together (apart from a few exceptions such as [Guenter et al., 2007]).

In the following, I will briefly present how the problem of learning and generalization of simple gestures through imitation has been addressed. By simple, I mean that I do not consider the sequencing aspects of movements. In order to avoid the correspondence problem, it is assumed that we have a set of examples adapted to the robot geometry. In the following, I distinguish between two kinds of movements. The first kind, called gestures, have a very loose interaction with the environment and are not directed to a precise target. Those gestures include drawing letters, martial arts movement, communicative gestures like waving. The second kind of movements, goal-directed movement, have a precise target given by the environment. Those movements include reach-to-grasp, reach-to-press or transporting an object to a given position.

2.5.1 Learning gestures

The most trivial way to learn a gesture is simply to sample the trajectory and then interpolate between sample points during replay. And indeed, this method has been predominant in robotics [Lozano-Pérez, 1983]. However, this method contains some drawbacks, as its inability to generalize the gesture to similar initial conditions. Moreover, human demonstrations may not be perfect, due to wiggles in the motions or poor timing. In order to improve the motion, [Delson and West, 1994] suggested to let the human perform several demonstrations, and use the variability in the demonstration as indication of how constrained the movement is. During reproduction the average of the demonstrations was used and optimized within the allowed variability, thus obtaining a movement that was better (in their case shorter) than the demonstrated ones. Other, more sophisticated averaging techniques have been used. First it became clear that, although those movements involve many DoFs, a

given movement lies within a lower-dimensional subspace or manifold. Methods for reducing the dimensionality of the movement included the use of motion primitives [Inamura et al., 2001, Ilg et al., 2004], projection on principal or independent components [Calinon and Billard, 2005], or the building of a lower-dimensional manifold (called Spatio-temporal Isomap) based on local distances across visited configurations [Jenkins et al., 2007]. Different averaging techniques were then suggested, using neural networks [Maurer et al., 2005, Lopes and Santos-Victor, 2005], Gaussian Mixture Models [Calinon et al., 2007], Hidden Markov Models [Inamura et al., 2004, Aleotti and Caselli, 2006] or some other Bayesian Networks [Grimes et al., 2007].
All those examples model the gesture as a simple function of time. In contrast, [Wang et al., 2006] models a walking motion as a dynamical system acting on a latent space (see Appendix A.3 for an introduction to dynamical systems). This means that at each time step, position is not given by a function of time, but rather as a function of the preceding state.

2.5.2 Learning goal-directed movements

Goal-directed movements involving some interaction with specific elements of the environment are more difficult to learn and to generalize, as the demonstrations and the reproduction are contingent to the actual state of the environment. This is why learning those kind of movements has often been reduced to learning how to sequence a set of built-in controllers to execute a particular task [Dillmann, 2004],[Ogawara et al., 2003]. Learning then occurs at a more abstract symbolic level, while the built-in controllers handle the low-level interaction with the environment.
The challenge in learning goal-directed movements is to reproduce desirable features of the demonstrations, while still reaching the target at the end of the movement, (bearing in mind that this target may change from time to time). If one sticks too much to the demonstrations, one may miss the target. If one tries to reach the target, the resulting trajectory may be to different from the demonstrations, resulting in task failure. To ensure task success, the robot should learn what the features of the movement are important for the successful accomplishment of the task. This is quite challenging, especially in the absence of any knowledge concerning the geometrical and dynamical properties of the environment. This explains why rather few works addressing this problem have been published. It was often thought to be easier to have a complete model of the environment and accomplish the task using some planning technique [Latombe, 1991].

There are, however a couple of works tackling this problem. In [Campbell et al., 2006] a robot learns the appropriate reaching trajectories in a reach and grasp task. Starting from the same position, a robot arm is shown through multiple teleoperation trials how to reach and grasp a vertical wrench located on 9 positions defining a volume in the work space. After learning, the robot is able to autonomously reach and grasp the wrench anywhere in the volume. This result is achieved by a locally linear interpolation of the demonstrated trajectories, assuming a nonlinear mapping between target position and trajectories.
In [Ijspeert et al., 2002] a dynamical system approach is taken (see Appendix A.3). Each joint angle trajectory is generated by linear dynamical system having

the target joint angle as attractor. Using a single demonstration, a modulation is learned, that will make the velocity profile of the generated trajectories similar to the demonstration. This results in trajectories that are very similar to the demonstration if the starting and ending points are similar. However, it remains unknown how the system generalizes to different settings for the accomplishment of a given task.

2.6 Summary

In this chapter, I have provided a brief overview of positional motor control and learning. After describing the manipulator control problem and its major component, the inverse kinematics problem, I have presented the classical engineering solutions to this problem. Those solutions all rely on *a priori* knowledge about the robot geometry. I then gave the essence of bio-inspired approaches trying to learn the inverse kinematics problem, without relying on this *a priori*. Those methods mostly work by exploring the configuration space of the robot and learning the relationship between commands sent to the motors and their effect on the end-effector. Those methods are limited to small numbers of degrees of freedom, lest the configuration space becomes too big. I also mentioned the intermediate self-calibration approach, where it is assumed that one has some *a priori* knowledge of the manipulator geometry, but that this knowledge is possibly inaccurate. Gathering many configurations and corresponding end-effector positions, it is possible to reduce the imprecision of this *a priori* knowledge. Those methods always assume a configuration phase prior to the actual use of the robot. (Alternative methods to those problems are presented in Chapters 3 and 4.)
I then explained how motor learning does not reduce to learning to reach a target or to follow a trajectory, but also learning appropriate trajectories for a given task. Restricting myself in a Programming by Demonstration framework, I gave a taste of how gestures are usually learned by some kind of (more or less sophisticated) averaging over demonstrated trajectories. I then explained that this method cannot be applied to goal-directed movements as one must ensure that the target is reached. I then gave a flavor of the two ways of addressing this problem available in the robotics literature. An other method, combining aspects of some of the presented methods will be proposed in Chapter 5.

Chapter 3

A model of the body schema

> Quelle est, à ma taille sans cesse en movement, sans cesse différente,
> la taille du monde.
>
> Paul Eluard

3.1 Introduction

In order to start our journey, let us first put ourselves in the shoes of a little homunculus that sits in our brain. We assume that this homunculus receives a variety of perceptual inputs, coming from different modalities. The question we are asking is then: how can the homunculus build a coherent representation of the "body in space" out of all those perceptions, in order to be able to know how to move in this space, for example for grasping an observed object. Assuming the homunculus lives in the digital age, it receives a flow of numbers describing its proprioceptively felt limb positions, the position of its hands and feet seen through the eyes, maybe also the tactile information that he is touching an object and so on. How can it "make sense" or "organize" all this information to form a coherent view of its potential for action? This is the first question we are addressing and to do so, we turn to the concept of *body schema*.

From a robotics perspective, the work presented here differs from previous approaches (see Section 2.4) in that the learning is performed entirely online and it can deal with a high number of degrees of freedom. Moreover, the model does not focus solely on determining the position of the end-effector, but also yields the position of each segment and can compute the associated Jacobians. It thus provides additional information, that can be very useful, e.g. for obstacle avoidance or for computing iterative local inverse kinematics.

The rest of this chapter is organized as follows. Section 3.2 presents the concept of body schema and related experimental evidence. A model of the body schema is then described in Section 3.3, followed by some experimental results in Section 3.4 and a short discussion in Section 3.5. Part of this work has been published in [Hersch et al., 2008b].

3.2 The body schema

3.2.1 A disputed concept

The concept of body schema is usually attributed to [Head and Holmes, 1912] who used this concept to describe a preconscious postural model of the body. Despite a great deal of confusion over this concept [Wallon, 1954, Gallagher, 2005b], it has survived for almost a century and it is still debated [Reed, 2002, Sheets-Johnstone, 2005, Paillard, 2005, Gallagher, 2005a]. For the purpose of this thesis, we rely on the definition given by [Paillard, 2005, p.99]:

> [T]he plurality of sensorimotor action spaces depending on the acting body segments and the involved sensory modality have to be coordinated in a unified amodal dynamic structure of space, anchored in a geotropically oriented postural frame, which constitutes the *body schema*.

For this work, the main and most useful points of this definition are the following

- The body schema performs the fusion (or coordination) of modalities.
- The body schema structures the representation of space.
- The body schema serves action, it is a pragmatic representation.

Less relevant for robotics, the body schema in this acceptation is also considered to be preconscious [Rossetti, 1998, Paillard, 2005].
To illustrate the function of the body schema, the case of Ian Waterman (often referred as I.W.) is often invoked (e.g. [Gallagher, 2005b]). At age 19, I.W. lost his sense of proprioception and of touch for its entire body below the neck. Thus he has no postural information about its body, he can only retrieve its limb positions by actually visually looking for them. This caused him to totally lose the ability to control its posture and its limbs. This ability was then slowly and partially regained during rehabilitation by adopting alternative conscious strategies for controlling its limbs, based on its intact modalities. However, I.W. must always be very careful while moving around in unknown or cluttered environments, not to hurt himself by hitting things, furniture or walls. The case of I.W. shows that proprioception is essential, not only for knowing one's own posture but also for controlling one's movements, keeping one's balance and have a feeling of oneself in space.

3.2.2 Relevant evidence

Although it has not necessarily be linked to the body schema, we review here some evidence based on psychophysical and neurophysiological studies that suggest that, in primates, multi-sensory information is integrated through a hierarchy of frames of reference that reflects the body structure. This hierarchy allows a mapping across the visual, proprioceptive, motor and tactile modalities and is highly adaptive. A more comprehensive review has been written by [Holmes and Spence, 2004].
A strong interaction between visual, tactile and proprioceptive sensory information has been demonstrated in several ways. The existence of bimodal visuotactile neurons in the monkey and psychophysical experiments involving cross-

modal extinctions have put in evidence the existence of a visuo-tactile representation [Làdavas, 2002], while the discovery of body-part centered visual fields [Rizzolatti et al., 1997] shows that there is a strong interaction between proprioception and vision.
Apparently each sensorimotor modality receives and provides information represented in different FoRs. For instance, proprioception, touch and motor commands are coded in a FoR centered on the specific body part they represent and control, [Aflalo and Graziano, 2007, Prud'homme and Kalaska, 1994, Tillery et al., 1996], whereas visual information is perceived in an eye-centered or retinotopic manner, [Jellema et al., 2004]. In order to form a coherent representation of one's own body, the information coming from various modalities needs to be somehow integrated. It has been suggested that this integration is made through a series of transformation across intermediary FoRs, located in between those provided by the sensory receptors [Burnod et al., 1999, Pouget et al., 2002]. Indeed, neurons coding position in FoR centered on body parts have been reported by [Graziano and Gross, 1993]

The adaptivity of those transformations is particularly evident in psychophysical experiments involving prism adaptation [Welch, 1986]. It has long been known, that when subjected to a visual shift or distortion caused by a prism, human subjects first tend to reach, expectedly, to the seen position, rather than to the actual position of the reaching target. After a while, however, they can correct for the visual distortion and accurately reach to the target. For this to occur, visual and proprioceptive feedback of the hand is necessary. When the visual distortion is removed, the subjects show so-called after-effects, i.e., they still reach for the virtual target, as if the visual distortion was still active. This occurs although they are aware that it is not the case. This adaptability has been demonstrated for visual shifts (rotations), reflections, and stretches but could not be shown for more complicated deformations which do not preserve the space topology [Bedford, 1999]. Furthermore, people could also adapt to transformations expressed in intrinsic (joint angle) coordinates [Imamizu et al., 1998].
Another kind of experiment emphasizing the adaptivity of the body schema, involves the use of tools. It has been shown that after some practice with a tool, the monkey integrates this tool into his body schema [Maravita and Iriki, 2004, Berti and Frassinetti, 2000]. The somatosensory receptive field of given neurons where observed to be expanded by the tool, after some practice.
Finally, the "fake limb" experiments also highlight the adaptive and tight connection between different sensory modalities and the feeling of one's own body [Botvinick, 2004]. In those experiments, a subject sees a fake limb being touched synchronously with his real unseen arm and feels that the fake arm is his.

3.2.3 Working hypotheses

The evidence summarized above argues in favor of the existence of a comprehensive framework, which allows to combine information across visual, tactile and proprioceptuo-motor modalities, and to perform the appropriate FoR transformations required for the integration of this information. Those transformations are highly adaptive, and are constantly learned as a result of sensory experience. In particular, these transformations can be adapted to accommodate distorted

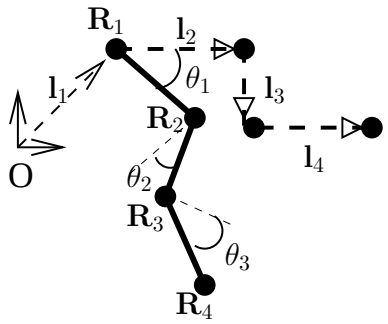


Figure 3.1: The parametrization of a kinematic chain. The dashed line represents the kinematic chain in the zero position (when all angles are equal to zero). The solid line represent the same chain with different rotation angles. O refers to the origin.

visual input, to include tools as an extension of the body. The adaptivity, of this framework, coexists with some innateness of its structure, as is suggested by the existence of aplasic phantom limbs, i.e., limbs felt by people born without them [Brugger et al., 2000].

3.3 A body schema for humanoid robots

3.3.1 Kinematic chains

Considering a serial manipulator with n rotative joints, it is possible to compute how a position given in the end-effector frame of reference (FoR) can be expressed in the manipulator base FoR. In other words, we can compute the FoR transformation from a FoR centered on the distal segment to the FoR centered on the proximal segment. This is done by considering the rotation and translation corresponding to each joint and segment, as is commonly done for computing kinematic functions, e.g. the Denavit-Hartenberg kinematic chain parametrization. This transformation can be seen as a series of successive rotations and translations, where the rotation angles and axes are given respectively by the manipulator joint angles and motor rotation axes, and the translations are given by the vector difference between the joints. Thus, it is possible to transform a vector $\mathbf{v}_n$ from a frame of reference centered on the end-effector to a vector $\mathbf{v}_0$ in a frame of reference centered on the other side of the chain by a transformation $\mathcal{T}$ described by the following equation:

$$\mathbf{v}_0 = \mathcal{T}(\mathbf{v}_n) = \mathbf{T}_1 \circ \mathbf{R}_1 \circ \mathbf{T}_2 \circ \mathbf{R}_2 \circ \cdots \circ \mathbf{T}_n \circ \mathbf{R}_n(\mathbf{v}_n) = \mathbf{l}_1 + \mathbf{R}_1(\mathbf{l}_2 + \mathbf{R}_2(\ldots(\mathbf{l}_n + \mathbf{R}_n(\mathbf{v}_n))\ldots)) \tag{3.1}$$

where $\mathbf{T}_i$ and $\mathbf{R}_i$ represents respectively the translation and rotation corresponding to segment i and joint i, and $\mathbf{l}_i$ denotes the vector representing the link proximal to joint i at the zero position, and $\mathbf{R}_i$ is the rotation caused by the joint i. Fig. 3.1 illustrates how the segments are numbered and how this FoR is computed. Note that, similarly to the Denavit-Hartenberg parametrization of kinematic chains, $\mathbf{l}_i$ can be zero if joints $i-1$ and i have intersecting rotation axes.

3.3.2 Static adaptation

Single segment adaptation

We consider the following problem regarding a single joint manipulator (see Fig. 3.2). We assume that we have an initial guess of the unit rotation axis $\mathbf{a}$ and the joint position $\mathbf{l}$. Now, given a vector $\mathbf{v}$ in a FoR centered on the distal segment, its actual transform $\mathbf{v}'$ centered on the proximal segment and the rotation angle θ, how is it possible to adapt $\mathbf{a}$ and $\mathbf{l}$ so that they account better for the actual transformation induced by the manipulator?
In order to do so, we perform a simple gradient descent on the squared distance between the actual and simulated transform vector:

$$\Delta\mathbf{l} = -\epsilon\frac{\partial}{\partial\mathbf{l}}\frac{1}{2}\Big|\Big|\mathbf{v}' - \big(\mathbf{l} + \mathbf{R}^{\theta}_{\mathbf{a}}(\mathbf{v})\big)\Big|\Big|^2 \tag{3.2}$$

$$\Delta\mathbf{a} = -\epsilon\frac{\partial}{\partial\mathbf{a}}\frac{1}{2}\Big|\Big|\mathbf{v}' - \big(\mathbf{l} + \mathbf{R}^{\theta}_{\mathbf{a}}(\mathbf{v})\big)\Big|\Big|^2, \tag{3.3}$$

where $\mathbf{R}^{\theta}_{\mathbf{a}}$ is the rotation of angle θ around axis $\mathbf{a}$ and the learning step ϵ is a small positive scalar. The derivative with respect to $\mathbf{l}$ in (3.2) is straightforward to compute. We have

$$\Delta\mathbf{l} = \epsilon\Big(\mathbf{v}' - \big(\mathbf{l} + \mathbf{R}^{\theta}_{\mathbf{a}}(\mathbf{v})\big)\Big). \tag{3.4}$$

In order to compute the derivative with respect to $\mathbf{a}$ in (3.3), we make use of the Rodrigues formula [Bauchau and Trainelli, 2003]

$$\mathbf{R}^{\theta}_{\mathbf{a}}(\mathbf{v}) = \cos(\theta)\mathbf{v} + \sin(\theta)\mathbf{a}\times\mathbf{v} + (1-\cos(\theta))\mathbf{a}^T\mathbf{v}\mathbf{a}. \tag{3.5}$$

Hence,

$$\bar{\mathbf{R}}^{\theta}_{\mathbf{a}} \doteq \frac{\partial}{\partial\mathbf{a}}\mathbf{R}^{\theta}_{\mathbf{a}}(\mathbf{v}) = \sin(\theta)\mathbf{v}\uparrow + (1-\cos(\theta))(\mathbf{a}\mathbf{v}^T + (\mathbf{a}^T\mathbf{v})\mathbf{I}), \tag{3.6}$$

where $\mathbf{I}$ is the 3×3 identity matrix and the unary operator $\uparrow$ is defined as

$$\mathbf{v}\uparrow \doteq \frac{\partial}{d\mathbf{a}}(\mathbf{a}\times\mathbf{v}) = \begin{pmatrix} 0 & v_3 & -v_2 \\ -v_3 & 0 & v_1 \\ v_2 & -v_1 & 0 \end{pmatrix}, \quad \text{with} \quad \mathbf{v} = [v_1\ v_2\ v_3]^T. \tag{3.7}$$

Thus

$$\Delta\mathbf{a} = \epsilon\Big(\mathbf{v}' - \big(\mathbf{l} + \mathbf{R}^{\theta}_{\mathbf{a}}(\mathbf{v})\big)\Big)^T\Big(\sin(\theta)\mathbf{v}\uparrow + \big(1-\cos(\theta)\big)\big(\mathbf{a}\mathbf{v}^T + (\mathbf{a}^T\mathbf{v})\mathbf{I}\big)\Big). \tag{3.8}$$

Since $\mathbf{a}$ must be of unit norm, it is normalized to 1 after being updated. This solves our problem. Using (3.4) and (3.8), it is possible to adapt the representation of the joint position and orientation online, as examples of positions in the distal FoR and the corresponding position in the proximal FoR are provided. This algorithm is proven to always converges to the correct translation and rotation axis when provided with enough different values of $\mathbf{v}$ and $\mathbf{v}'$, (see Section 3.3.4).

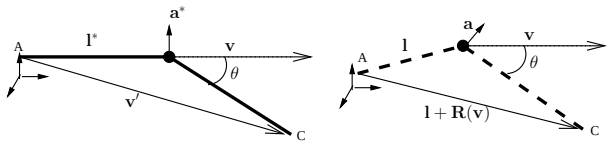


Figure 3.2: The learning problem for a single segment. The real rigid transformation is shown on the left and is parametrized by unknown vectors $\mathbf{l}^*$ and $\mathbf{a}^*$ and known angle θ. The current guess of this rigid transformation appears on the right and is parametrized by $\mathbf{l}$, $\mathbf{a}$ and θ. Knowing a vector $\mathbf{v}$ and its real transform $\mathbf{v}'$ and the rotation angle θ, we try to update our guess of $\mathbf{l}$ and $\mathbf{a}$. The letters A and C indicate respectively the origin and the end-effector of the manipulator.

Multi-segment adaptation

We can now apply the same principle to multi-segment manipulators. Starting from (3.1) it is possible to compute

$$\Delta\mathbf{l}_i = -\epsilon\frac{\partial}{d\mathbf{l}_i}\frac{1}{2}\|\mathbf{v}'_n - \mathcal{T}(\mathbf{v}_n)\|^2 \tag{3.9}$$

$$\Delta\mathbf{a}_i = -\epsilon\frac{\partial}{d\mathbf{a}_i}\frac{1}{2}\|\mathbf{v}'_n - \mathcal{T}(\mathbf{v}_n)\|^2, \tag{3.10}$$

where $\mathbf{a}_i$ is the rotation axis of $\mathbf{R}_i$ and $\mathbf{v}'_n$ is the actual (or observed) transform of $\mathbf{v}_n$. If $\mathbf{R}_i$ is the rotation matrix corresponding to joint i (i.e., of axis $\mathbf{a}_i$ and angle θ_i), we have

$$\frac{\partial}{d\mathbf{l}_i}\mathcal{T}(\mathbf{v}_n) = \prod_{j=1}^{i-1}\mathbf{R}_j \tag{3.11}$$

$$\frac{\partial}{d\mathbf{a}_i}\mathcal{T}(\mathbf{v}_n) = \Big(\prod_{j=1}^{i-1}\mathbf{R}_j\Big)\frac{\partial}{d\mathbf{a}_i}\Big(\mathbf{R}_i\big(\mathbf{T}_{i+1}\circ\mathbf{R}_{i+1}\cdots\circ\mathbf{T}_n\circ\mathbf{R}_n(\mathbf{v}_n)\big)\Big) \tag{3.12}$$

where the derivative on the right-hand side of the last equation is obtained by applying (3.6). All the rotation axes and translation vectors can thus be simultaneously updated using

$$\Delta\mathbf{l}_i = \epsilon\Big(\mathbf{v}'_n - \mathcal{T}(\mathbf{v}_n)\Big)^T\prod_{j=1}^{i-1}\mathbf{R}_j \tag{3.13}$$

$$\Delta\mathbf{a}_i = \epsilon\Big(\mathbf{v}'_n - \mathcal{T}(\mathbf{v}_n)\Big)^T\Big(\Big(\prod_{j=1}^{i-1}\mathbf{R}_j\Big)\frac{\partial}{d\mathbf{a}_i}\Big(\mathbf{R}_i\big(\mathbf{T}_{i+1}\circ\mathbf{R}_{i+1}\cdots\circ\mathbf{T}_n\circ\mathbf{R}_n(\mathbf{v}_n)\big)\Big)\Big) \tag{3.14}$$

3.3.3 Rotation-based adaptation

In the previous section, the parameters of a kinematic chain were estimated using a set of static postures. In this section, the same parameters are estimated using the Jacobian of the transformation, i.e. by considering the effect of joint rotations on the end-effector. As we shall see, this method is complementary to the one described above, as it enables to update the kinematic chain parameters

based on a different set of measurements.

The problem we address is illustrated in Fig. 3.3. We try to update estimates $\mathbf{a}_i$ and $\mathbf{l}_i$ of the rotation axes and joint positions respectively, knowing the joint angles θ_i, small joints displacements $\tau\dot{\theta}_i$, and the resulting rotation of the end-effector $\mathbf{R}^*$.
For doing so we use the quaternion parametrization of rotations, or more precisely the Rodrigues parametrization (see Appendix A.2.1). Let $\mathbf{b}' \in \mathcal{B}_1$ be the vector describing the observed rotation of the end-effector, where $\mathcal{B}_1$ is the 3D ball of radius 1 centered on the origin. We can perform a gradient descent on the squared difference between the predicted rotation $\mathbf{b}$ and the observed rotation $\mathbf{b}'$ of the end-effector:

$$\Delta\mathbf{a}_i = -\epsilon\frac{\partial}{d\mathbf{a}_i}\frac{1}{2}\|\mathbf{b}' - \mathbf{b}\|^2. \tag{3.15}$$

The predicted rotation $\mathbf{b}$ can be computed as

$$\mathbf{b} = \sum_{k=1}^{n}\frac{\partial}{d\theta_k}(\prod_{j=1}^{n}\mathbf{R}_j)\tau\dot{\theta}_k = \tau\sum_{k=1}^{n}(\prod_{j=1}^{k-1}\mathbf{R}_j)\mathbf{R}'_k(\prod_{j=k+1}^{n}\mathbf{R}_j)\dot{\theta}_k, \tag{3.16}$$

where $\mathbf{R}'_k$ is the derivative of $\mathbf{R}_k$ with respect to θ_k. If all rotations are represented by quaternions and all products in the equation above are quaternion products (see Appendix A.2.1), then $\mathbf{R}'_k$ is given by the following quaternion:

$$\mathbf{R}'_k = \frac{1}{2}[\cos(\frac{\theta_k}{2})\mathbf{a}_k^T \quad -\sin(\frac{\theta_k}{2})]^T.^1 \tag{3.17}$$

Deriving $\mathbf{b}$ with respect to $\mathbf{a}_i$ yields

$$\begin{aligned}\frac{\partial}{d\mathbf{a}_i}\mathbf{b} &= \tau\sum_{k=1}^{i-1}\dot{\theta}_k(\prod_{j=1}^{k-1}\mathbf{R}_j)\mathbf{R}'_k(\prod_{j=k+1}^{i-1}\mathbf{R}_j)\bar{\mathbf{R}}_i(\prod_{j=i+1}^{n}\mathbf{R}_j)\\ &+ \tau\dot{\theta}_i(\prod_{j=1}^{i-1}\mathbf{R}_j)\bar{\mathbf{R}}'_i(\prod_{j=i+1}^{n}\mathbf{R}_j)\\ &+ \tau\sum_{k=i+1}^{n}\dot{\theta}_k\prod_{j=1}^{i-1}\mathbf{R}_j\bar{\mathbf{R}}_i(\prod_{j=i+1}^{k-1}\mathbf{R}_j\mathbf{R}'_k\prod_{j=k+1}^{n}\mathbf{R}_j),\end{aligned} \tag{3.18}$$

where $\bar{\mathbf{R}}_k$ and $\bar{\mathbf{R}}'_k$ can be expressed as the "quaternion matrix" described respectively by (A.28) and (A.29) in Appendix A.2.2. [2] In that case, only the first three rows of the global resulting "quaternion matrix" are considered, because the Rodrigues parametrization corresponds to the first three elements of

[1] If rotations are represented with matrices and the product is the matrix product, then the matrix $\mathbf{R}'_k$ is obtained by deriving (3.5):

$$\mathbf{R}'_k = \sin(\theta_k)(\mathbf{a}_k\mathbf{a}_k^T - \mathbf{I}) - \cos(\theta_k)\mathbf{a}_k\uparrow .$$

[2] If matrix representation is used we have

$$\bar{\mathbf{R}}'_i(\mathbf{v}) \doteq \frac{\partial}{d\mathbf{a}_i}\frac{\partial}{d\theta_i}\mathbf{R}_i(\mathbf{v}) = -\cos(\theta)\mathbf{v}\uparrow - \sin(\theta)(\mathbf{a}\mathbf{v}^T + (\mathbf{a}^T\mathbf{v})\mathbf{I})$$

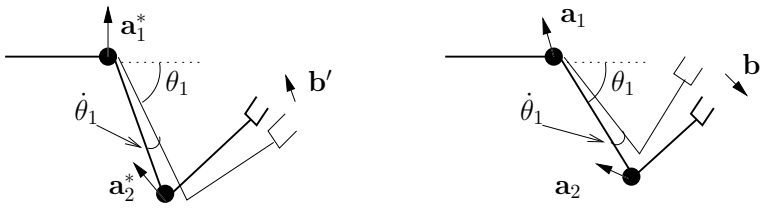


Figure 3.3: Rotation-based adaptation. With the "true" kinematic chain (left) parametrized by $\mathbf{a}_1^*$ and $\mathbf{a}_2^*$, moving from a configuration θ (thick line) to a configuration $\theta + \tau\dot{\theta}$ (thin line), results in a rotation $\mathbf{b}'$ of the end-effector. With the estimated kinematic chain (right) parametrized by $\mathbf{a}_1$ and $\mathbf{a}_2$, this results in a rotation $\mathbf{b}$ of the end-effector. The problem is to how update $\mathbf{a}_1$ and $\mathbf{a}_2$, by knowing θ, $\tau\dot{\theta}$ $\mathbf{b}$ and $\mathbf{b}'$ so that $\mathbf{b}$ gets closer to $\mathbf{b}'$. (θ_2 and $\tau\dot{\theta}_2$ are not shown.)

the quaternion parametrization. Thus the updating of the rotation axes $\mathbf{a}_i$ is performed by

$$\Delta\mathbf{a}_i = -\epsilon(\mathbf{b}' - \mathbf{b})^T \frac{\partial}{d\mathbf{a}_i}\mathbf{b}, \tag{3.19}$$

where $\mathbf{b}$ and $\frac{\partial}{d\mathbf{a}_i}\mathbf{b}$ are given respectively by (3.16) and (3.18)

This learning method considers only the Jacobian of the orientation component of $\mathbf{x}$. Hence it does not depend on the translation vectors. A similar method can be derived for the Jacobian of the position component of $\mathbf{x}$. Since it was found to have no real advantage over the static adaptation method, it is not presented here.

3.3.4 Convergence

Static adaptation

For the static adaptation algorithm, in the single-joint case, there is a formal convergence result. It is given by the following theorem.

Proposition 1 *Assuming that we run the algorithm on a set of configurations given by* $\{\mathbf{v}, \mathcal{T}^*(\mathbf{v}), \theta_j\}_{j=1}^J$*, where the* θ_j *follow a symmetric probability density function (pdf) centered on* 0*, such that* $\mathrm{var}(\cos\theta_j) < 2\mathrm{var}(sin\theta_j)$*, the algorithm described by iteratively applying (3.2) and (3.3) converges to a correct estimate of* $\mathbf{a}$ *and* $\mathbf{l}$*.*

The proof is given in Appendix B.1.

Multi-joint case The convergence for the multi-segment case cannot be proven. In order to have an idea of the convergence properties, simulations were performed. In a single simulation run, the rotation axes $\mathbf{a}_i^*$ and $\mathbf{a}_i$ of two kinematic chains were randomly generated. The $\mathbf{l}_i$ were initialized with small random values and the algorithm was run in order to see whether the $\mathbf{a}_i$ converge to the $\mathbf{a}_i^*$. Convergence is considered to be attained if the distance between the real limb position and the modeled limb position remains smaller than a threshold (around 1% of chain length) over 500 different configurations. Ten thousands

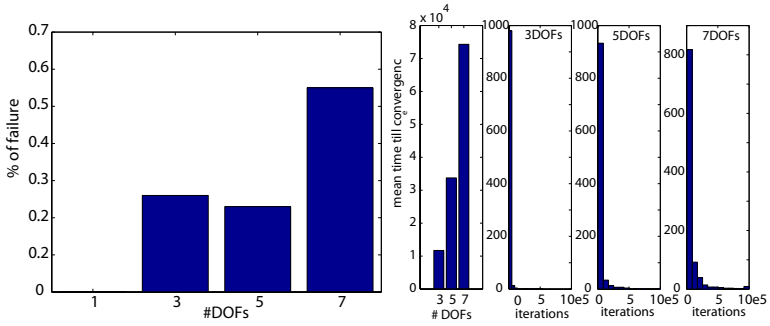


Figure 3.4: Left: the percentage of trials that did not converge after a million iterations. Right: the time needed for convergence depending on the number of DoFs. The bars on the left show the mean number of iterations until convergence, and the three histograms on the right show the distribution of convergence time. The distribution have quite a long tail, indicating that in some cases it takes much longer than average to converge.

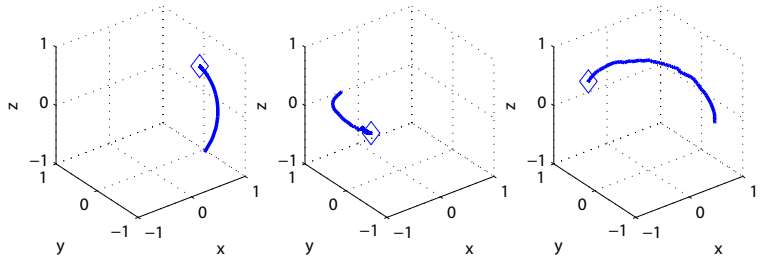


Figure 3.5: The evolution of the rotation axes for a 3-DOF kinematic chain using the static update. The real axes are indicated by the three diamonds, and each graph show the evolution of one estimated axis on a sphere of radius one

of those runs were performed for 1,3,5 and 7 DoFs kinematic chains. The results can be seen in Figure 3.4, left. Expectedly, if there is only one joint, the algorithm always converges. When there are more DoFs, the algorithm fails to converge after a million iterations in less than 1% of the cases. The time it takes for convergence is plotted in Fig. 3.4, right. Figure 3.5 gives an example of the evolution of the estimate of the rotation axes for a kinematic chain containing 3 DoFs.

Rotation-based adaptation

Proposition 2 *The algorithm described by (3.19) on a single joint and converges to a correct estimate the rotation axis.*

The proof is given in Appendix B.1.2.

Multi-joint case In order to investigate the convergence properties of the rotation-based adaptation, simulations were again performed, using kinematic

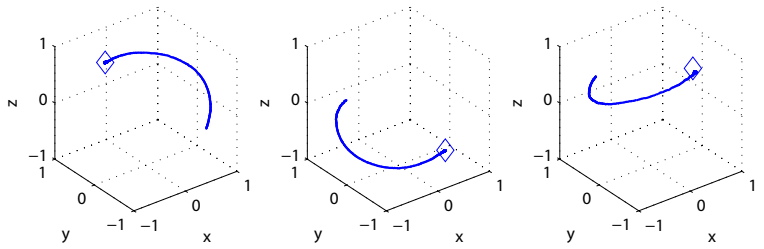


Figure 3.6: The evolution of the rotation axes for a 3-DOF kinematic chain using the rotation-based update. The real axes are indicated by the three diamonds, and each graph show the evolution of one estimated axis on a sphere of radius one.

chains of one, three, five and seven degrees of freedom. To estimate the convergence rate, a series of runs were performed, whereby two kinematic chains with random axes where generated. A joint angle configuration θ was also randomly generated, as well a a small joint angle displacement $\dot{\theta}$. The first kinematic chain was updated according to (3.19), where the value for $\mathbf{b}'$ was given by the second kinematic chain. Convergence was achieved if after 10 millions runs all rotation axes of the first chain would be in a small neighborhood of the rotation axes of the second chain. An example where convergence is achieved is displayed on Fig 3.6. The number of such runs was 10 thousands for one and three DoF chains and one thousand for five and seven DoF chains. Two neighborhood sizes were used, 10 and 20 degrees. The results, are shown in Fig. 3.7, left. One sees that for one and three DoF chains, the algorithm always converges. For five and seven DoF chains, this is not the case. When considering a 10 degrees neighborhood, the algorithm fails to converge in about 4% of the cases for five DoF chains and 20% for seven DoF chains. Those numbers reduce to 0.3% and 4% respectively when considering a 20 degrees neighborhood. It has been observed in the cases where the algorithm fails to converge, most rotation axes end up in a neighborhood of their "true" value, with the exception of a couple of them being further away, but never more than 45°. Further studies would be necessary to investigate the shape of the basin of attraction of the algorithm, but those simulations seem to indicate that the algorithm usually converges to a region close to the true values.

3.3.5 Adaptive body schema

Kinematic Tree

The humanoid body schema can be represented as a tree of rigid transformations reflecting the limbs structure, as shown in Fig. 3.8. We thus have a kinematic tree with adaptive joint positions and orientations. Note that the structure of this tree (i.e., the number of joints and the ordering) is given and remains fixed. Out of this tree, kinematic chains can easily be extracted as paths in the tree. It is possible to compute the FoR transformation from a FoR attached to any joint of the kinematic tree to a FoR attached to any other joint. This is done by first finding the path joining the two corresponding nodes. To each edge along this path, there corresponds a FoR transformation. Depending on the direction

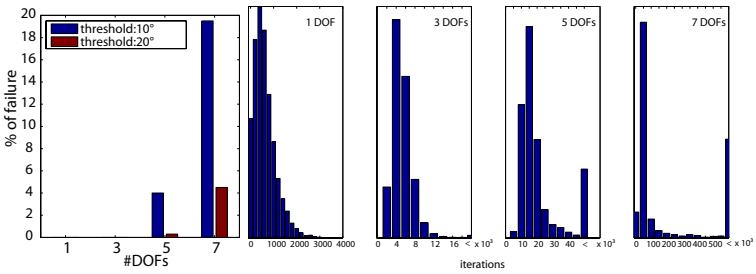


Figure 3.7: The convergence for the rotation-based adaptation. First left: the percentage of convergence after 10 million iterations, for two difference convergence thresholds, 10 and 20 degrees. Four right: The distribution of convergence time for one, three, five and seven DoF chains.

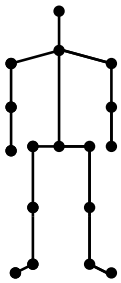


Figure 3.8: A kinematic tree representing a humanoid. Nodes represent rotations and edges represent translations.

an edge is taken, the transformation or its inverse is considered. An example is provided in Fig. 3.9, showing how a kinematic chain is extracted from the kinematic tree.

Static body schema learning

We assume that the robot is endowed with a calibrated stereo-vision system that can track the 3D position of its end-effectors. This position is provided in a head-centered frame of reference. Within the kinematic tree, the path going from the head to the end-effector corresponds to a kinematic chain that transforms positions and orientations from a frame of reference centered on the end-effector to a visual or head-centered frame of reference. Using (3.14) and (3.13), it is possible to update all the rigid transformations along this chain. As input the $\mathbf{v}'_n$ are given by the stereo-vision system and $\mathbf{v}_n$ is the position of the end-effector in its own frame of reference. This is illustrated in Fig 3.10, left. This figure also illustrates how the same algorithm could be used with tactile sensors for tactile body schema learning. Instead of using vision to close the kinematic chain, tactile sensors indicate how to close the kinematic chain. Of course, the same algorithm can be used to learn kinematic chains which are already closed like parallel robots, although this is somewhat less biologicaly relevant.

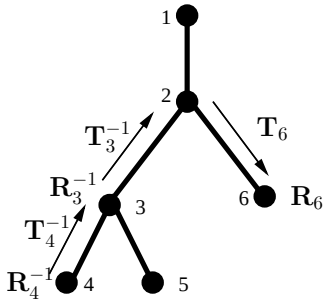


Figure 3.9: The generating of the kinematic chain for the FoR transformation relating the FoR attached to two nodes in the tree. When taking an edge up (to the root), one takes the inverse transform, and when taking an edge down (to the leaf) the actual transform is taken. In this example, the transformation from a FoR centered on joint 6 to a FoR centered on joint 4 is given by $\mathbf{R}_4^{-1} \circ \mathbf{T}_4^{-1} \circ \mathbf{R}_3^{-1} \circ \mathbf{T}_3^{-1} \circ \mathbf{T}_6 \circ \mathbf{R}_6$.

Head kinematics learning

The rotation-based adaptation described in Section 3.3.3 can be used to learn the rotation axes of the DoFs of the head. As illustrated on Fig 3.10, right, when robot moves its head and eyes, it induces a visual flow as if the environment was moving. It is then possible to analyze the rotational component of this visual flow, to know what is the rotation resulting from the head and eye movements. If this analysis is done at a sufficient rate, it can be taken as the Jacobian of the orientation component of the kinematic function. It is thus possible to use (3.19) to update the estimate of the kinematic chain of the head, assuming a fixed torso. If another part of the body is anchored to the environment (for example the buttock), then it is the kinematic chain from this anchor point to the eyes that can be learned.

Subjective body schema

Traditionally, the body schema has been considered as an objective account of the body characteristics, such as the arrangement of the limbs, their lengths or the positioning and effect of the joints. To this "objective" body schema, it is possible to oppose a "subjective" view of the body schema, which would be dependent on the perceptual abilities of the robot. In this view, which is adopted in this thesis, the body schema only indirectly deals with physical properties of the body. It primarily deals with the frame of reference transformations associated to the sensory signals. For example, given a proprioceptive input corresponding to a particular posture, the body schema can predict the corresponding visual perception. This depends not only on the physical properties of the body, but also on the properties of the sensory system. Moreover, it can be that a precise account of the physical properties of the body is not necessary to the "subjective body schema", depending on the sensory system. For example, in our case, if the robot can track only end-effector positions, many different body geometries will yield the same "subjective" body schema. In the simple example depicted in Fig. 3.11, such a robot will not be able to differentiate between the two "objective" body schemata. They will both correspond to the

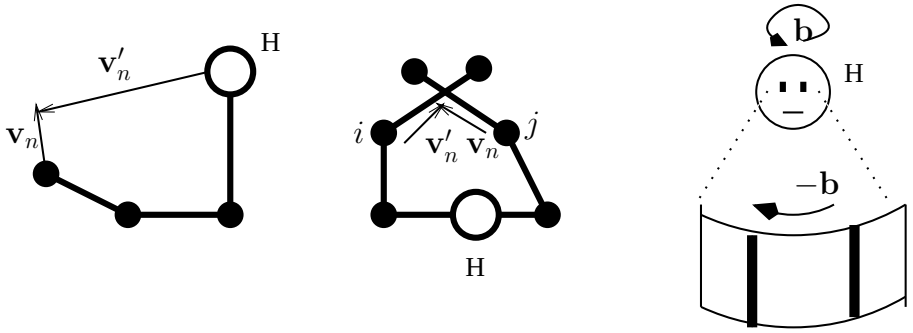


Figure 3.10: Humanoid body schema adaptation. The big circle marked H is the head of the humanoid. Left: visual learning, side-view of the robot. The position of the limb in its own frame of reference $\mathbf{v}_n$ is transformed into a visual head-centered frame of reference $\mathbf{v}'_n$. Center: tactile learning, top-view. The position of touch sensors in the frame of reference of their limb $\mathbf{v}_n$ and $\mathbf{v}'_n$ transform into one another. Right: head kinematics learning, front view. When the robot turns its head of an amount $\mathbf{b}$, it sees the world turning in the opposite direction ($-\mathbf{b}$) of its eyes.

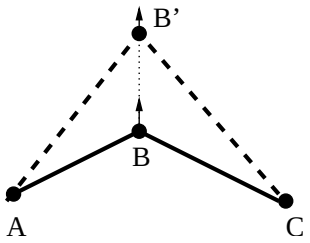


Figure 3.11: A simple example of two different geometries (solid and dashed) yielding the same FoR transformation from a A-centered to a C-centered FoR.

same "subjective" body schema. As the corresponding kinematic function and Jacobian remain the same in both cases, using one or the other body schema for controlling its movements will produce the same end-effector trajectories.

3.4 Experiments

3.4.1 Simulations

In order to validate the algorithm described above, we first tested it in simulation on a 24 DoFs humanoid robot. The simulated humanoid (or avatar) has the shape of the Fujitsu Hoap3 robot and comprises 24 DoFs. A schematics of the robot is drawn in Fig 3.12. When learning the body schema, the avatar configuration space was randomly sampled with a uniform distribution. The joint angles and corresponding visual position were fed into the algorithm. The body schema was initialized with random joint orientations and small random body segments.

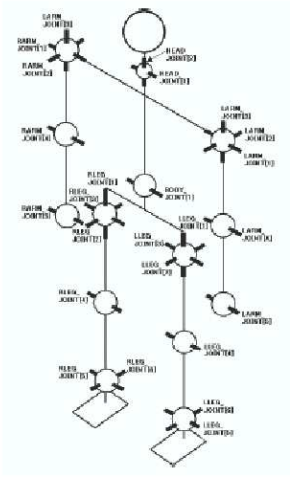


Figure 3.12: The structure of the Hoap3 robot. Spheres show the joints, with the rotation axis shown as dark lines going through them. The hand rotations were not used, and an additional head roll joint was modeled. This makes 24 DoFs. This picture is taken from the Hoap documentation provided by Fujitsu.

In a first validation step, we used the static adaptation algorithm described above to learn the body schema of the Hoap3 robot. In this first experiment, only the two hands and feet were tracked. This means that four kinematic chains where concurrently used: head - right hand, head - left hand, head - right foot, head - left foot. At each time step a joint angle configuration was randomly chosen, and the corresponding position of the hands and feet in a head-centered FoR where computed for the Hoap3 robot. Along with the joint angle values, those four positions where fed as $\mathbf{v}'_n$ (see (3.13), (3.14)) for adapting the corresponding kinematic chains. The result can be seen in Fig. 3.13 which plots the error of the kinematic function, i.e. $\|\mathbf{v}'_n - \mathcal{T}(\mathbf{v}_n)\|$ at each iteration. The after many iterations, the system converges to the appropriate subjective body schema as the error converges to zero. Note however, that the objective body schemata differ, as can be seen in Fig. 3.14.

In this experiment, a minimal amount of information is provided by the vision system as it always only tracks end-effectors, like hands and feet. But it is also possible for the vision system to track non-terminal body parts, like elbows and knees, as well. This, of course, is expected to make the system converge to the right geometry and also to considerably speed-up the learning process, as much more information is available to the system. Indeed, tracking non-terminal body parts amounts to having shorter kinematic chains which significantly reduces the dimension of the problem.

So, in a second experiment, the vision system alternatively tracks terminal (hand and feet) and non-terminal (knees,elbows, shoulders and waist) body segments. In this case, convergence is faster and the robot geometry is correctly retrieved as there is no ambiguity on the joint locations (see Fig 3.14).

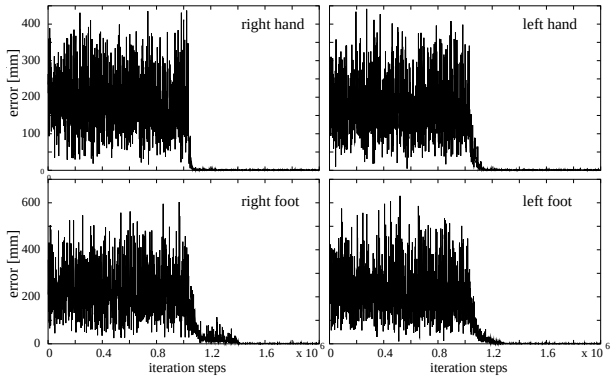


Figure 3.13: The convergence of the learning algorithm. On the y-axis the error on the computation of limb position, in a head-centered frame of reference.

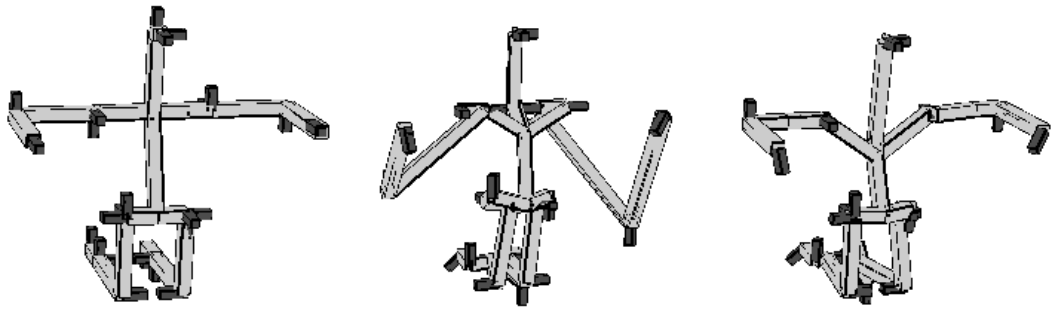


Figure 3.14: Left: the Hoap3 "real" body schema, when standing on its knees. Middle: The objective body schema that was learned by the system when looking only at its hands and feet. It is not exactly the same, but the sensory relationships of both schemata (i.e., the subjective body schemata) are the same, when considering only vision of the hands and feet. Right: The body schema learned when also looking at elbows, knees, shoulders and the waist. The darker sticks indicate the rotation axis of each joint. There are three DoFs at the shoulders and the hips.

3.4.2 Real robot experiments

Tool use adaptation with the Hoap3

In order to evaluate the practicality of our approach, I conducted an experiment on a real robotic setting. In this experiment, the robot carries an unknown tool. By looking at the tip of the tool the robot can integrate this tool into his body schema, thus enabling it to manipulate it adequately. This setting is shown on Fig. 3.15, left. I used the Hoap3 robot, which is endowed with a stereo-vision system.
In the results presented here, the robot is initialized with its "real" body schema. The robot holds a 340 mm long stick in the hand with a color blob at its tip. I then make the robot passively move its arm, with the tip of the stick remaining within the field of view of the cameras. The stereo-vision system tracks the tip of the stick and the robot joint angles are read from the motor encoders. Those two set of values (position of the tip of stick and joint angles value) are then continuously fed into the learning algorithm, like in the simulation experiments. The vector $\mathbf{v}_n$ in (3.13) and (3.14) is the position of the end-effector in its own FoR, in other words zero. Two cases (using the same data) are tested. In the first case only the terminal limb is adaptive, i.e. the position and orientation of the non-terminal joints are kept constant. In the second case the whole arm is adaptive, as in the simulations presented above. The results are displayed in Fig. 3.16. This figure shows the distance between the position of the end-effector as seen by the stereo-vision on one hand, and as computed by the body schema on the other hand. In other words, the y-axis plots $\|\mathbf{v}'_n - \mathcal{T}(\mathbf{v}_n)\|$, where $\mathbf{v}'_n$ is given by the stereo-vision and $\mathbf{v}_n$ is zero. One can see that in both cases the system starts with an error of about 40cm, which corresponds approximately to the length of the stick, and reduces this error to about 5cm. This means that the stick has been incorporated in the body schema, although imprecision in the stereo-vision system keeps this error of about 5cm. In the plot, one can notice a few peaks of large errors. Those are outliers of the stereo-vision system, and correspond to a bad tracking of the color blob. Of course, the resulting error values are inevitable, but it is interesting to notice that the system is quite robust to such outliers, as the next error values are again in a reasonable range. It takes about 2000 steps to reduce this error if only the terminal limb transformation is adaptive and 1000 steps if all the limb transformation are adaptive. This takes two to three minutes as updates are performed at a rate of approximately 10 Herz.

Learning to reach and track with the iCub

I also used this algorithm to implement real world reaching with the iCub humanoid robot as illustrated in Fig 3.15, right. The iCub is a 54 DoFs humanoid robot built as part of the RobotCub project [Sandini et al., 2004], of which this thesis is part. This robot has 7 DoFs in each arm (not including the hand) and 6 DoFs in the head (3 in the neck and 3 for the eyes). Being "home-made", it has not yet gone through the full benchmarking and calibration process, so that its encoders reference values are quite imprecise. Moreover, the calibration of the stereo-vision system is done by holding a chessboard, which determines the frame of reference used by the stereo-vision system. This is done in such a way that this frame of reference remains more or less parallel to the frame of

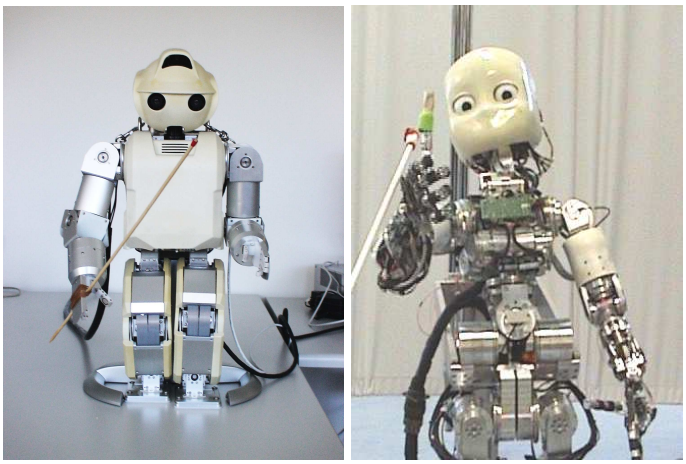


Figure 3.15: The setup of the real robot experiments. Left: the Hoap3 robot holds a stick and visually tracks its tip and records its arm joint angl values, while its arm is passively moved. Right: The iCub robot reaches for a target while tracking its thumb.

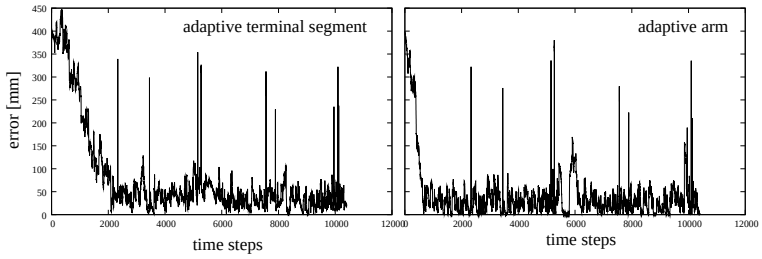


Figure 3.16: Incorporating the tool in the Hoap3 body schema. Those graphs show the evolution of the distance between the end-effector position seen by the hand on one hand, and computed using proprioception and the body schema on the other hand. On the left, only the terminal limb is adaptive, on the right all the limbs are adaptive. In both cases the body schema adaptation reduces the error from approximately 40 cm to 5 cm. Vertical bars are errors due to vision outliers.

reference attached to the robot, but it is also quite imprecise. This really jeopardizes the possibility to perform accurate reaching movements with the robot. Bearing this in mind I did the three following experiments.

Autonomous self-calibration In this experiment, I initialized the robot body schema with the theoretical, uncalibrated values. A color marker was fixed on the robot right thumb and was tracked by the head and eyes of the robot. The robot performed reaching movements, using the algorithm described in Chapter 4. In the beginning, reaching movement were very imprecise. However, after a couple of minutes, the system learned the correct body schema and could perform accurate reaching movements. Fig. 3.17, left shows the distance between the expected and seen positions of the thumb. In the beginning this gap is of about 10 centimeters, meaning the the robot could not be more precise than that. After a couple of minutes, the robot adapts its body schema and can precisely predict the position of its hand. And indeed, it can perform accurate reaching movements. There are four peaks indicating large errors on this plot. The last three are of very short duration and can be attributed to a bad tracking of the end-effector (outliers of the stereo-vision system). The first one, at step 300 is gradual and persists for a while, and can thus not be attributed to a poor visual tracking. It shows that at this stage the body schema is inadequate. It is due to the fact that first, the body schema adapts locally, the learned parameters are only good for a given region of space. When the robot moves to a different region (at step 300), those parameters are no longer adequate, thus causing a gap between the expected and the observed end-effector position. After a while though, the system learns parameters that are globally valid and the error remains low.
On the right of Fig 3.17, one sees the evolution of all rotation axes of the arm. After some adaptation, they remain rather stable, meaning that the robot has found an adequate body schema.

Having the initial body schema not too far from the real one allows the robot to be able to look at its hand. If the initial body schema is too different from the real body schema, the robot gaze very seldom encounters its hand, so that there is almost no data to feed the algorithm with. This problem can be handled as illustrated by the next experiment.

Learning to track with the eyes In this experiment, I used the rotation-based learning algorithm to learn the kinematic chain of the head, as described in Section 3.3.5. Three joint rotation axes of the head were initialized with the wrong values, inverting tilt and pan rotation axes. The robot tried to track a color blob using an adapted version of the reaching algorithm described in Chapter 4. While moving its head and eyes, the optical flow induced by those movements was analyzed and the corresponding rotation was fed as input of (3.19). In the beginning, the robot could of course not properly track the target, having the wrong body schema of the head. As pan and tilt axes were inverted the robot would, for example turn its head left when it should raise its eyes. However after several minutes, the robot would correct for the wrong rotation axes and could track the moving target. The results are displayed in Figs. 3.18 and 3.19. One sees that for the neck tilt, the rotation axes was initialized along

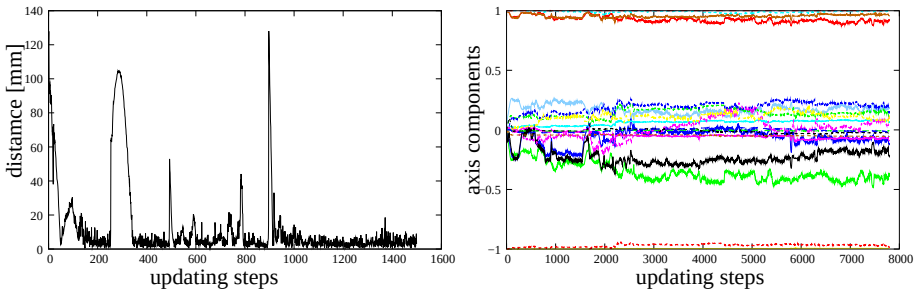


Figure 3.17: Left: the evolution of the distance between the observed and the expected finger position. In the beginning, the distance is about ten centimeters but it quickly reduced to less than a centimeter. Right: the evolution of all rotation axes of the arm. The x,y,z coordinates of all axes are shown. Initially, all rotation axes are oriented along the x, y, or z axes (the projections on those axes is either 0, 1 or -1). After adaptation, the rotation axes stabilize on different values.

the y axis instead of the x axis. After a while this rotation axis align itself with the correct x axis. Similar evolutions can be seen for the neck pan joint and the eyes tilt joint.

Full learning The two experiments described above where united into a single experiment. Starting with bad rotation axes of the head, the robot first learned to visually track the target. When its tracking ability became sufficient, it could then track the target and its hand. It then started to perform reaching movements and update its body schema using the static adaptation. The distance between the expected and seen hand position is plotted in Fig. 3.20. Again, the robot learned its body schema and could accurately reach the target. So the robot could perform accurate reaching movements, even if it was initialized with a "wrong" body schema.

3.5 Discussion

3.5.1 Robotics

Our results show that the suggested framework for body schema learning is effective. Indeed, simulation results show that it can learn the structure of a 24 DoFs robot by tracking only the end-effectors. Furthermore, real robot experiments show that the method is applicable in a real setting. There is quite a big difference in the precision obtained with each robot. After learning the Hoap robot could predict its end-effector position with a precision of about five centimeters, whereas the iCub could do it with a precision of one centimeter. This is explained by the implementation. For the iCub, visual and proprioceptive input were provided by the same computer, an on-board PC104. Thus, time stamps could be used to make sure that visual and proprioceptive data were temporally matched. This could not be done for the Hoap, so that there was probably some delay between the visual and proprioceptive data, which resulted in a lower precision.

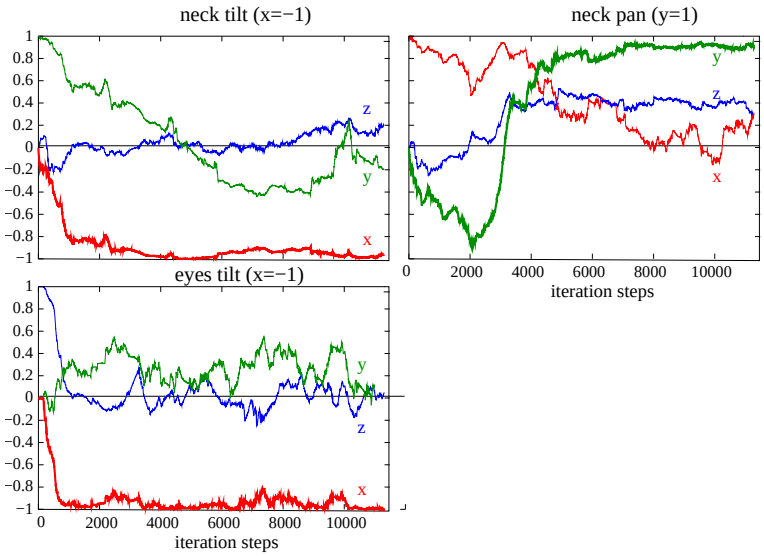


Figure 3.18: The evolution of the rotation axes of the head. The rotation exes of the head were initialized with the wrong values, pan and tilt rotation axes being inverted. After a while the rotation axes evolve close to their true value indicated in the title (see bold lines). The axes were updated at a rate of 10 Hz, meaning that the 10000 iteration steps correspond to 1000 seconds, i.e 16 minutes. The same data is represented differently in Fig. 3.19.

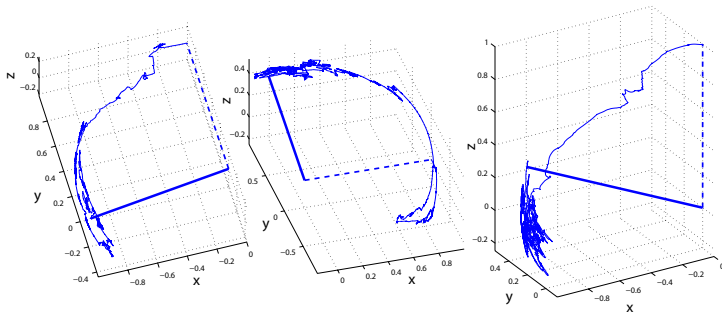


Figure 3.19: The evolution of the rotation axes of the head. This is the same data as Fig. 3.18, plotted as trajectories on a ball of radius 1. The dashed line shows the initial configuration and the solid is the final one. Note that the exact "real" value for those axes is unknown, and can therefore not be plotted. But the axis reach to the right region, that is $-x$ for the first and third axes, and y for the second one.

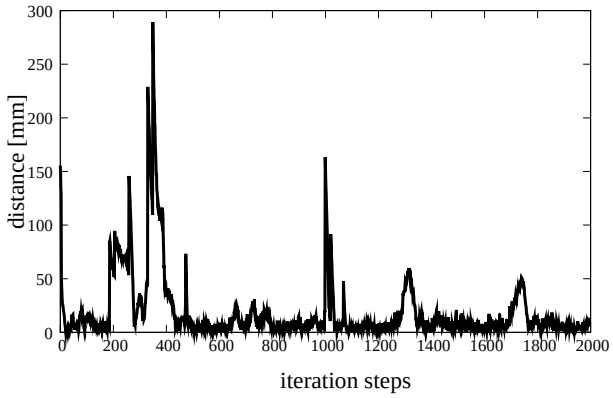


Figure 3.20: The evolution of the distance between the observed and the expected finger position. Again, after learning, the gap between expected and seen finger position reduces from 15 to 1 centimeter.

Given the fast developments of actuator and sensor technologies, it may be possible in a near future to use this algorithm with all DoFs of real humanoid robot. For practical purposes, as it may be cumbersome to gather millions of points in the training set, it may be more efficient to apply this algorithm to a smaller set of data, well distributed in the joint ranges and perform the iterations on this more restricted set. Another possibility to speed up the learning process would be to have an active exploration strategy. Such a strategy would for example preferentially explore configurations of the workspace that are not well predicted. Or one could start by "freezing" some degrees of freedom while learning the others, as suggested by [Berthouze and Lungarella, 2004]. This makes the learning problem easier and the algorithm would thus converge faster. Taken to its extreme, such a strategy could be used to learn one degree of freedom at a time and thus ensure the convergence of the algorithm. But although this could be useful from a roboticist's perspective, this would amount to a pure calibration phase, which is not very natural.
The model presented here differs from earlier work on this topic mainly in that the knowledge of the kinematic structure is given in advance. In other words, the robot not only knows the number of DoFs, but also how they are arranged (serially or in parallel). Moreover, the model takes explicitly advantage of the fact that those are rotative joints, which was usually not done in other works. Thus the effectiveness of the learning algorithm relies on this a priori knowledge. However, I believe that it is reasonable to assume such an a priori knowledge, as the kinematic structure of the humanoids usually do not change over time. Similarly the kinematic structure of humans is fixed and does not evolve. Limbs grow, but new joints do not appear in a lifetime.

3.5.2 Biological relevance

As stated above, the aim of this model is to endow the robot with the ability to learn its sensorimotor contingencies but also to provide some insight on biological systems. Of course, the body schema in its definition given in Section 3.2.1 includes much more than what has been modeled in this work. However, this model could be a good starting point, and however limited it raises interesting questions about the human body schema.

First it hints at possible computational explanations for psychophysical observations, such as adaptation to distorted vision, or the integration of tools in the body schema. (Even if the analogy between tool and limb control is not new and can already be found in Aristotles [Aristotles, 1978]).

In the proposed model, the tree structure, i.e., the number of limbs and the numbers of DoFs in each of them is given and is not the outcome of a learning process. It is not yet known for sure if this is also the case for the human body schema, but the existence of aplasic phantom limbs [Brugger et al., 2000] seems to indicate that this is indeed the case. Aplasic phantom limbs are limbs felt by subjects, although they were born without them, due to some malformation. Apparently, persons with a congenitally missing limbs have, at some level, a representation of that limb. This seems to indicate that some representation of the body structures exists independently from the sensory experience (but see [Price, 2006] for alternative explanations). It could be something analog to the tree structure described in this chapter.

If the body schema can be modeled as a tree of FoR transformations, another question that this model raised is the nature of those transformations. Whereas the robot motors can faithfully be modeled as rotation, this is not necessarily true for human joints. Elbow and knee joints effect are similar to a rotation around a fixed axis, but other joints such as the shoulder are more complex. And if one considers the spine, it is unlikely that each vertebra is modeled as a rotation. However, any limb motion is rigid and can thus be modeled by a rotation and a translation, and it seems reasonable to assume that something similar to the rotation angle around a fixed axis can be retrieved from the proprioceptive signal.

Experiments have shown that body schema adaptation can also happen at a local level, just for a particular region of space [Ghahramani et al., 1996, Malfait et al., 2008] or for particular velocities [Kitazawa et al., 1997]. These effects cannot be explained by the present model, as a modification of the rotation axis or translation vector would affect space perception in the whole peripersonal space. Additional (or different) mechanisms should be invoked to explain those effects.

Importantly, the body schema adaptation presented here is performed entirely online, that is without a distinct learning phase. The robot is constantly learning and adapting as a a result of sensory input. Although this may seem trivial to the naturalist, this is not the case for the roboticist, which often use batch learning. Batch learning consists in first gathering all the necessary data (training phase) and then fit a model with it (learning phase), which is usually much easier. But batch learning is inappropriate to model biological systems that absorb data as a continuous flow and where learning is done gradually and continuously.

3.5.3 Cognition

This chapter has shown that it is possible for a humanoid robot to learn the relationship between its proprioceptive and visual sensory input, in robotics term, the kinematic function. However, there is a (maybe subtle) difference between the robot body schema presented here and the traditional kinematic function. In standard robotic (e.g. [Sciavicco et al., 2000]) the kinematic function transforms a set of angles from the configuration space into coordinates of a task space, which is the pre-given, objective "existing" Cartesian space "out there". The relevance and mere existence of the so-called objective space has been put into question by [Varela et al., 1991], who consider this perception of space to be the result of our sensory, motor, and cognitive abilities with no *a priori* existence. Similarly, in the model of the body schema presented here, the "kinematic function" does not transform joint angles into a pre-existing objective space, but rather into a visual space, i.e. into a subjective sensory space. The fact that this visual space is a Cartesian space homomorphic to the putative objective space, is of course an *a priori* built into the system. It was motivated primarily by practical reasons. This could be avoided by integrating the stereo-vision process into to body schema, which would not only include translations and rotations but also projections. Work in this direction has been done in [Bennett et al., 1991], and a study of the emergence of a 3D space representation has been presented in [Philipona et al., 2004].

Because this model of the body schema consists in a hierarchy of coordinate system transformations, it can also be seen as a model of peripersonal space representation. Indeed, as mentioned in Section 3.2.1 the body schema *structures* the perception of space and is thus intrinsically linked to space representation. Although perception and action have traditionally been considered separately, a body of evidence from neurophysiology [Rizzolatti et al., 1997], psychology [Proffitt, 2006], along with current theories such as the theories of motor perception [Fadiga et al., 2000, Galantucci et al., 2006] and of sensorimotor account of consciousness [O'Regan and Noë, 2001] suggest that those two concepts should be unified in a single framework. Indeed perception and action are densely intertwined processes [Higuchi et al., 2006], that cannot be easily isolated one from another. So far, the motor modality has not yet been considered. This shortcoming is dealt with in the next chapter.

3.6 Summary

In this chapter, I have reviewed psychophysical and neurobiological evidence showing that humans are able to develop an adaptive representation of their peripersonal space. This representation is very tightly linked to the so-called body schema, so tightly indeed, that it is possible to argue, as is done here, that they are indistinguishable. The evidence reviewed covers the phenomenas of prism adaptation, multi-modal neurons, fake limbs and tool inclusion in the body schema.
I have then presented a novel computational model of how such an adaptive peripersonal space representation can develop. This model integrates visual, and proprioceptive modalities, and could include touch as well. It differs from other

existing models of visuo-motor mappings in that it explicitly models the limb structure as a set of rigid transformations assembled in a tree structure. The various modalities can then be merged using the hierarchy of frame of reference transformations defined by this tree. Simulation and real robot experiments show the effectivity of the algorithm to learn the robot body schema even if it comprises a high number of degrees of freedom, and its ability to incorporate tools in the body schema.

Chapter 4

Control of reaching movements

Je m'envole,
je m'envole, à la façon du temps,
sur une trajectoire unique
irréversible.

Lubomir Levtchev

4.1 Introduction

In the preceding chapter, we have seen how it is possible for a humanoid robot to develop a representation of its peripersonal space and, equivalently, of its body schema. We have seen that the relationship between the proprioceptive and visual information can be represented in a compact manner using a tree of rigid-body transformations mirroring the limb structure of the body. In this chapter, we address the *control* problem. In other words, we investigate how this peripersonal space representation can be used to perform movements. More precisely, and in a robotic setting, we look for appropriate motor commands that will bring a particular body part to a given location and, possibly, with a given orientation. This problem is probably the most classical problem in robotics, and many solutions have been suggested, using various frameworks (see [Sciavicco et al., 2000] for an overview, and Section 2.3 for a brief summary of standard position control methods). Drawing our inspiration from the study of human movement control, we present, in the following sections, a new controller for reaching motions.
The rest of this chapter is structured as follows. Section 4.2 emphasizes the differences between robotic and human actuation systems, which put the developmental roboticist in a somewhat uncomfortable posture. Current theories on human control of reaching motions are then summarized in Section 4.3. Inspired by those theories, a new controller for reaching motions is presented in Section 4.4, and its implementation is described in 4.5. The experimental results are then provided in Section 4.6 followed by a discussion in Section 4.7. Part of this work was presented in [Hersch and Billard, 2006] and [Hersch and Billard, 2008].

4.2 Of robot and human actuation

In the preceding chapter, the peripersonal space representation I suggested and described could be applied for robotic experiments but could also be seen as a model of human peripersonal space representation. As mentioned in the discussion (section 3.5) the model assumes that all joints are rotative joints, which is almost the case for humans. So it is reasonable to make the hypothesis that there is no such a big fundamental difference between the kinematic structure of a robot and that of a human. This analogy, however becomes quite shaky when it comes to control. Indeed the physical mechanisms by which limbs are moved are very different. For the robots considered here, limbs are moved by motors which convert an electric current into a magnetic field, thus producing a torque on the rotor attached to the limb and rotating it. Therefore each motor, and thus each limb, can be controlled independently in torque. Using a motor position sensor and applying a feedback control loop (such as a PID) each limb can also be independently controlled in position.
For humans, the actuation, i.e. the limb displacement, is performed by the muscle-tendon complex, which is rigidly connected to the skeletal structure [van Ingen Schenau and van Soest, 1996]. What apparently complexifies the matter is that a single limb may be attached to many muscles and a single muscle may span several limbs. So the contracting of a particular muscle may affect several limbs, even those not spanned my the muscle, through the influence of joint reaction forces. So joint are most probably not controlled independently. To make things worse (for scientist to understand), the force exerted by a muscle does not only depend on the neural activation but also on the muscle length, velocity and contraction history. And some of this force is absorbed by the tendon elasticity. So what exactly is controlled by the neural signals remains largely unknown, but it is most probably not directly the position of the joint or the force exerted by it, as in a standard robotic actuation system. A more detailed account of the human motor system can be found in [Latash, 1998].

4.3 Human control of reaching movements

Here, we briefly review the main current theories on the control of reaching movements in humans. Better and more exhaustive overviews can be found in [Desmurget et al., 1998b, Shadmehr and Wise, 2005].
The modern and systematic studies of human movements can be traced back to the work of Woodworth [Woodworth, 1899], at the end of the 19^{th} century. Those studies are characterize by the quest for some kind of invariants in the reaching movements performed by humans. The first invariants investigated by Woodworth were the relationships between movement speed and accuracy. In the fifties, [Fitts, 1954] proposed the so-called Fitts' law establishing relationships between the movement velocity and its precision, and the movement amplitude and its velocity. In the eighties, a lot of work was devoted to finding invariants in the reaching trajectory. It was argued in [Atkeson and Hollerbach, 1985] that the bell-shaped velocity profile was invariant when normalized for speed and distance. Other kinds of invariants were suggested, such as the relationship between hand trajectory curvature and speed, the so-called 2/3 power law [Lacquaniti et al., 1983]. It was believed that identifying the invariants of the

movement would help discover the principles and rules underlying the human control of motion.

For example, observing that the hand paths of reaching movements are quasi-straight, [Morasso, 1981] suggested that movement were planned in Cartesian space. In the following years, analogous arguments were used to hypothesize a control in joint angle space, as movements were found to also be quasi-straight in that space [Soechting and Lacquaniti, 1981, Lacquaniti et al., 1986]. This hypothesis was supported by other evidence including the predictivity of final posture [Desmurget et al., 1995], variability of unconstrained reaching trajectories performed in the dark [Magescas and Prablanc, 2006], Further studies argued that both joint angle and Cartesian space frames of reference were used, either concurrently [Cruse and Brüwer, 1987, Carrozzo and Lacquaniti, 1994], or depending on the task setting [Desmurget et al., 1997]. According to the latter, free movements are controlled in joint space, while constrained movements (such as 2D movements) were controlled in Cartesian space.

To further inquire about the frames of reference used for planning reaching movements, experiments involving reaching to remembered targets were performed. By varying the initial conditions (arm, head or hand position), it was investigated whether the target position was stored in a FoR centered on the eyes, the head, the torso, the hand, or the environment. Results indicated that targets were memorized in FoR centered on the arm and the eye [Lemay and Stelmach, 2005, Beurze et al., 2006]. Other results indicated that an eye-centered FoR is used [Batista et al., 1999, Vetter et al., 1999] More generally, it was argued that the nervous system could adapt the FoR depending on the task demand [Adamovich et al., 1998, Ghafouri et al., 2002].

Other kinds of invariants where investigated, such as optimality criteria governing the control of reaching movements. The underlying hypothesis is that trajectory planning is performed by optimizing some given functional of the movement. Various criteria were suggested [Engelbrecht, 2001], such as minimizing the total jerk (i.e. the path integral of the third time derivative of the hand trajectory) [Hogan, 1984, Flash and Hogan, 1985, Hogan and Flash, 1987], minimum torque change [Nakano et al., 1999], minimum energy [Alexander, 1997].

It was also observed that subjects could adapt the control of their movements when the dynamics of the environment was modified, for example by an external force field [Shadmehr and Mussa-Ivaldi, 1994]. When subject's arm is submitted to such an artificial force field, the reaching trajectories are first importantly disturbed. After some practice, however, the subject can again reach a target with quasi-straight trajectories, as if there was no force field. When the force field is removed, one can observe so called after-effects, i.e. the reaching trajectories are perturbed again, but this time in the other direction. Those after effects disappear after some practice in the natural setting. To explain those observations, it was suggested that the nervous system learns a forward model of the dynamics the the arm in its environment [Wolpert et al., 1995, Gandolfo et al., 1996]. According to the latter, the first a trajectory is planned and the appropriate commands for realizing this trajectory are computed using the forward model and sent to the muscles.

However, the distinction between trajectory planning and execution advocated by engineers [Hollerbach, 1982], never gained a full consensus among scholars. Many of them, in particular those coming from a biological background, objected it. According to them, movements are control by a dynamical system acting on a small number of variables. According to this dynamical system approach [Kelso, 1995], trajectory are not planned. Rather, they are implicitly specified by a dynamical system, and unfold as time goes by. The exact nature of this dynamical system and of the variables involved is still unclear. One of the earliest and most influential model is the equilibrium point hypothesis, dating back to the mid-fifties [Bizzi et al., 1984, Feldman and Levin, 1995, Feldman and Latash, 2005, Feldman, 2006]. This model is based on a study of muscle properties and suggests that the dynamics are provided by the biomechanical properties of the arm. According to this theory, the signals sent to the muscles are thresholds for muscle activation, which indirectly specify an arm configuration were agonist, antagonist and external forces are balanced, the equilibrium point. This point can be seen as an attractor for the arm, due to the spring properties of the muscles.
Another hypothesis fitting into the dynamical system approach is the stochastic optimal feedback control theory [Todorov and Jordan, 2002]. According to this view, the control is operative only along the dimensions relevant to the particular task at hand. Perturbations that do not interfere with the goal, will thus not be corrected. This is the *minimal intervention principle* [Todorov, 2004], also related to the concept of an *uncontrolled manifold* [Scholz and Schöner, 1999]. The controller is thus optimal in some (still to be defined) sense, with respect to the task. This theory aims not at reproducing precise trajectories, but rather the variability among trajectories for a particular task. Any performed trajectory is dependent, among others, on noise at the sensory level and can hence not be precisely modeled. Due to the random aspect of the noise, trajectories are best seen as realizations of a random variable. This view is reminiscent of the interpretation of quantum mechanics given by [Prigogine, 1996]. Other dynamical systems were also suggested, such as the Vector Integration To Endpoint (VITE) [Bullock and Grossberg, 1988] described below.

To sum up, the main question concerning the human control of reaching movements is what is being actually controlled and in what frame of reference. This problem is still unsolved [Admiraal et al., 2004]. The evidence accumulated during several decades of research seems to indicate that those questions may be somewhat too simplistic and that many frames of reference can be (possibly simultaneously) used to control the arm. It all depends on the task at hand. It seems reasonable to hypothesize that humans can adapt the control of their movements and choose the most appropriate frame of reference. Similarly, they can adapt to different dynamics of the environment, and they can adapt the movement trajectories, if it does not interfere with the task. This is compatible with the dynamical system approach to motor control. Below, I present the VITE dynamical system, as it will be used in the following of this chapter.

The VITE model

The Vector Integration To Endpoint (VITE) model [Bullock and Grossberg, 1988] describes the neural signals commanding a pair of agonist-antagonist muscles in a point-to-point reaching movement. This model was extended for various purposes like writing movements [Bullock et al., 1993, Paine et al., 2006], via-points motions [Bullock et al., 1999] and grasping [Ulloa and Bullock, 2003, Vilaplana and Coronado, 2006]. Here we only present the simplest and original version of VITE. In this model, the target limb position $\mathbf{T}$ is assumed to be known. The actual limb position is given by a signal $\mathbf{P}(\mathbf{t})$. The model hypothesizes the existence of a "difference vector population" of neurons with activity $\mathbf{V}$ and a scalar "go signal" $G(t)$, which gates the execution of the movement. The VITE model for a single muscle is then described by the following equations:

$$\dot{\mathbf{V}} = \alpha(-\mathbf{V} + \mathbf{T} - \mathbf{P}) \quad (4.1)$$

$$\dot{\mathbf{P}} = G \cdot [\mathbf{V}]^+, \quad (4.2)$$

where α is a positive constant and $[\cdot]^+$ indicates the positive value function (i.e 0 if the argument is negative). The go signal $G(t)$ is usually taken to be an exponential function. From a modeling perspective, its function is mainly to smooth the acceleration at movement onset, thus enhancing the similarity with the velocity profiles of human reaching motions.
Applying this model on a pair of muscles, agonist and antagonist, and taking a step-like go function yields

$$\ddot{\mathbf{r}} = \alpha(-\dot{\mathbf{r}} + \beta(\mathbf{r_T} - \mathbf{r})). \quad (4.3)$$

In this equation, the vector $\mathbf{r}$ represents the limb position under the influence of both agonist and antagonist muscles, $\mathbf{r_T}$ the target position and $\alpha, \beta \in [0\ 1]$ are constant scalars. This is the equation of the spring-and-damper system described in Appendix A.3. It is straightforward to show that this dynamical system has a stable attractor on $\mathbf{r_T}$ if the latter is assumed to be fixed. So, from any starting point, the system will be brought smoothly to the target and remain there. Moreover, for $\alpha > 4\beta$, there is no trajectory overshoot. A critically damped solution is obtained if $\alpha = 4\beta$. Trajectories for various values of α and β are illustrated in Fig. 4.1. The evolution of the position $\mathbf{r}$ given by this equation is illustrated in Fig. 4.1 for different parameters α and β.

4.4 A model of reaching movement control

4.4.1 Overview

The idea driving the design of the robot arm controller described here is to unite a VITE-like dynamical system control with a multi-referential control. This is achieved by considering two VITE-like sub-controllers active in different spaces. The first sub-controller is active in the joint angle space (or arm configuration space) and the second sub-controller is active in the end-effector location space (or Cartesian space). Each sub-controller is specified by the VITE-like dynamical system described in section 4.3. Those two dynamical systems concurrently act on two distinct sets of variables, robot joint angles for the first one, and robot end-effector location for the second one.

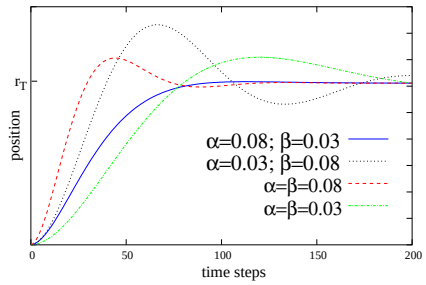


Figure 4.1: The dynamics of the VITE model for various values of the parameters. Throughout this document the values $\alpha = 0.08$ and $\beta = 0.03$ were used. Refer to (4.3) for variable definitions.

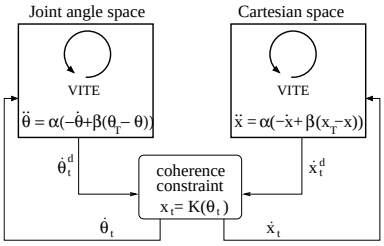


Figure 4.2: A schematic representation of the system. A VITE-like dynamical system is applied to the joint angle variables θ and to the end-effector variables $\mathbf{x}$. Coherence constraints are enforced between the two sets of variables.

However, those sets of variables are not independent, as the end-effector location is uniquely determined by the robot arm configuration. Thus coherence constraints are enforced so that the joint angles specified by the first dynamical system correspond to the end-effector location specified by the second dynamical system. In other words, there are two redundant representations of the movement, each one with its own dynamics. Those two dynamics are coupled by coherence constraints (described in section 4.4.2) that ensure that no contradiction between the two representations occur. The structure of this controller is shown in Fig. 4.2.
This coupling can then be modulated in order to adapt to the relative influence of the sub-controllers, thus allowing a smooth transition from one controller to the other. This amounts to adapting the control strategy to the situation at hand. In section 4.4.5 this feature is exploited in order to obtain a very simple and effective joint limit avoidance mechanism.
In summary, our controller can be viewed as two linear dynamical systems coupled through non-linear coherence constraints. As such, the resulting global dynamical system is non-linear.

4.4.2 Coherence Enforcement

It is clear that if one formally applies the dynamical system described in (4.3) to both the robot joint angles and the end-effector location, the result will bear

no sense. This is due to the fact that both sets of variables (joint angles and end-effector location) will be brought to values where they do not correspond to each other. In order to avoid this, coherence constraints are enforced, which will force the joint angles and the end-effector location to be consistent with each other at all times. This is achieved by finding the position which is closest to the positions given by each of the dynamical systems, while remaining coherent. In other words, at every time step the two dynamical systems specify velocities $\dot{\mathbf{x}}_t^d$ and $\dot{\theta}_t^d$ that bring the manipulator to a position $(\mathbf{x}_t^d, \theta_t^d)$ which is not coherent. Coherence is then enforced by finding the coherent velocities $(\dot{\mathbf{x}}_t, \dot{\theta}_t)$ that will bring the system closest to $(\mathbf{x}_t^d, \theta_t^d)$. Those velocities can be found by solving the following constrained optimization problem:

$$\begin{aligned} \min_{\dot{\theta}_t, \dot{\mathbf{x}}_t} \quad & \frac{1}{2}((\dot{\theta}_t - \dot{\theta}_t^d)^T \mathbf{W}_\theta (\dot{\theta}_t - \dot{\theta}_t^d) + \\ & (\dot{\mathbf{x}}_t - \dot{\mathbf{x}}_t^d)^T \mathbf{W}_\mathbf{x} (\dot{\mathbf{x}}_t - \dot{\mathbf{x}}_t^d)) \qquad (4.4) \\ \text{u.c.} \quad & \dot{\mathbf{x}}_t = \mathbf{J_t} \dot{\theta}_t, \qquad (4.5) \end{aligned}$$

where $\mathbf{J_t}$ is the Jacobian of $\mathbf{K}$ at θ_t. It is assumed that the $\dot{\theta}_t^d$ and $\dot{\mathbf{x}}_t^d$ are small enough so that the constraint (4.5) is equivalent to $\mathbf{x}_{t+1} = \mathbf{K}(\theta_{t+1})$. Here the semi-definite positive diagonal matrices $\mathbf{W}_\theta \in \mathbb{R}^{n \times n}$ and $\mathbf{W}_\mathbf{x} \in \mathbb{R}^{m \times m}$ serve as coefficient indicating the respective weight that one should give to the desired joint angles or end-effector location, and account for the unit difference between $\mathbf{x}$ and θ. Because it is a positive quadratic optimization problem subject to linear constraints, this problem has a single minimum. It can be found using Lagrange multipliers. The quantity to be minimized can be expressed as

$$\begin{aligned} L(\dot{\theta}_t^d, \dot{\theta}_t, \dot{\mathbf{x}}_t^d, \dot{\mathbf{x}}_t) &= \frac{1}{2}\big((\dot{\theta}_t - \dot{\theta}_t^d)^T \mathbf{W}_\theta (\dot{\theta}_t - \dot{\theta}_t^d) + \\ & \quad (\dot{\mathbf{x}}_t - \dot{\mathbf{x}}_t^d)^T \mathbf{W}_\mathbf{x} (\dot{\mathbf{x}}_t - \dot{\mathbf{x}}_t^d)\big) - \lambda^T (\dot{\mathbf{x}}_t - \mathbf{J_t} \dot{\theta}_t), \qquad (4.6) \end{aligned}$$

where λ is the vector of Lagrange multipliers. After differentiating with respect to $\dot{\theta}_t$ and $\dot{\mathbf{x}}_t$ and setting to zero, one gets

$$\begin{aligned} \frac{\partial L}{\partial \dot{\theta}_t} &= \mathbf{W}_\theta (\dot{\theta}_t - \dot{\theta}_t^d) + \mathbf{J_t}^T \lambda = 0 \qquad (4.7) \\ \frac{\partial L}{\partial \dot{\mathbf{x}}_t} &= \mathbf{W}_\mathbf{x} (\dot{\mathbf{x}}_t - \dot{\mathbf{x}}_t^d) - \lambda = 0 \qquad (4.8) \end{aligned}$$

and thus

$$\begin{aligned} \dot{\theta}_t &= \dot{\theta}_t^d + \mathbf{W}_\theta^{-1} \mathbf{J_t}^T \lambda \qquad (4.9) \\ \dot{\mathbf{x}}_t &= \dot{\mathbf{x}}_t^d - \mathbf{W}_\mathbf{x}^{-1} \lambda \qquad (4.10) \end{aligned}$$

Inserting those equations into (4.5) yields

$$\mathbf{J_t}\big(\dot{\theta}_t^d + \mathbf{W}_\theta^{-1} \mathbf{J_t}^T \lambda\big) = \dot{\mathbf{x}}_t^d - \mathbf{W}_\mathbf{x}^{-1} \lambda \qquad (4.11)$$

$$\lambda = \big(\mathbf{W}_\mathbf{x}^{-1} + \mathbf{J_t} \mathbf{W}_\theta^{-1} \mathbf{J_t}^T\big)^{-1} (\dot{\mathbf{x}}_t^d - \mathbf{J_t} \dot{\theta}_t^d) \qquad (4.12)$$

This equation is then used to replace λ in (4.9).

$$\dot{\theta}_t = \dot{\theta}_t^d + \mathbf{W}_\theta^{-1} \mathbf{J_t}^T \big(\mathbf{W}_\mathbf{x}^{-1} + \mathbf{J_t} \mathbf{W}_\theta^{-1} \mathbf{J_t}^T\big)^{-1} (\dot{\mathbf{x}}_t^d - \mathbf{J_t} \dot{\theta}_t^d) \qquad (4.13)$$

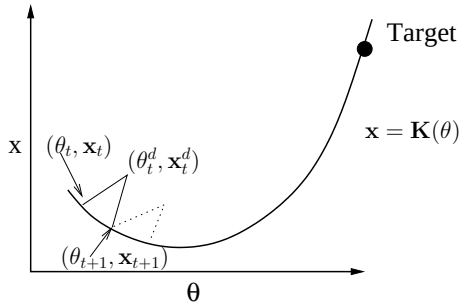


Figure 4.3: Coherence enforcement between the two dynamical systems. At time t, the system is in position $(\theta_t, \mathbf{x}_t)$ which is coherent ($\mathbf{x}_t = \mathbf{K}(\theta_t)$). Each of the sub-controllers bring the system to an incoherent position $(\theta_t^d, \mathbf{x}_t^d)$, which is then projected on the closest coherent position $(\theta_{t+1}, \mathbf{x}_{t+1})$.

An alternate representation of (4.13) can be formulated as (see [Billard et al., 2006])

$$\dot{\theta}_t = (\mathbf{W}_\theta + \mathbf{J_t}^T \mathbf{W_x} \mathbf{J_t})^{-1} (\mathbf{J_t}^T \dot{\mathbf{x}}_t^d + \dot{\theta}_t^d). \tag{4.14}$$

Although simpler as (4.13), this formulation is disadvantageous from an implementation perspective. Using $\mathbf{W}_\theta^{-\mathbf{1}}$ and $\mathbf{W}_\mathbf{x}^{-\mathbf{1}}$ instead of $\mathbf{W}_\theta$ and $\mathbf{W_x}$ allows to avoid infinity when dealing with a pure angular controller. It is indeed equivalent to have $\mathbf{W}_\theta^{-\mathbf{1}}$ (respectively $\mathbf{W}_\mathbf{x}^{-\mathbf{1}}$) or $\mathbf{W_x}$ (respectively $\mathbf{W}_\theta^{-\mathbf{1}}$) equal to zero. Moreover, this formulation is faster to compute, because it requires a matrix inversion of degree m, whereas (4.14) requires the inversion of a matrix of degree $n > m$ (for redundant manipulators).

We can notice that the inverse kinematics method given by (4.13) or (4.14) is a generalization of the Weighted Least Norm and also of the Damped Least Square (DLS) inverse kinematics described in Section 2.3.2. Indeed, by setting $\mathbf{W}_\theta$ to zero and $\mathbf{W_x}$ to identity, one gets the Weighted Least Norm method which makes use of the Moore-Penrose pseudo-inverse of the Jacobian to compute the adequate joint velocities. Moreover setting $\dot{\theta}_t^d$ to zero and $\mathbf{W}_\theta$ to identity, one gets the DLS inverse method which avoids singularities.
The parameters $\mathbf{W}_\theta$ and $\mathbf{W_x}$ control the influence of each of the sub-controllers. By setting $\mathbf{W_x}$ to zero, one obtains a pure joint angle controller and by setting $\mathbf{W}_\theta$ to zero, the result is a pure end-effector location controller.

Another, perhaps more intuitive, way of understanding the coherence constraints enforcement is to consider the joint angles and the end-effector location in a joint (or hybrid) space of dimension $n + m$. In this space, (2.1) defines a n-dimensional manifold of consistent positions. Thus the system can be seen as a VITE-like dynamical system in this hybrid space, whereby the position is constantly projected on the manifold of consistent positions, as illustrated by Fig. 4.3. In this view, the parameters $\mathbf{W}_\theta$ and $\mathbf{W_x}$ determine the direction of the projection.

4.4.3 Target Configuration Redundancy

As mentioned above, the joint angle dynamical system takes a target arm configuration $\theta_{\mathbf{T}}$ as input. In case of a redundant manipulator, there can be an infinite number of such $\theta_{\mathbf{T}} \in \Theta_T$ corresponding to a desired target location. In order to decide which one to choose, we take (at each time step) the one that is closest (in the standard Euclidean norm) to the actual arm configuration. This way, the joint displacement prescribed by the joint angle controller is minimized. If we already have a $\theta_{\mathbf{T}} \in \Theta_T$, it is possible to update this value by $\tau\dot{\theta}_{\mathbf{T}}$ to bring it closer to the actual arm configuration θ, while enduring that $\mathbf{K}(\theta_{\mathbf{T}} + \tau\dot{\theta}_{\mathbf{T}}) = \mathbf{x}_T$. [1] Note that the target itself may be moving with velocity $\dot{\mathbf{x}}_{\mathbf{T}}$. The update value $\tau\dot{\theta}_{\mathbf{T}}$ can be found by solving the following constrained optimization problem

$$\min_{\dot{\theta}_{\mathbf{T}}} \quad \frac{1}{2}(\theta_{\mathbf{T}} + \tau\dot{\theta}_{\mathbf{T}} - \theta)^T(\theta_{\mathbf{T}} + \tau\dot{\theta}_{\mathbf{T}} - \theta) \tag{4.15}$$

$$\text{u.c.} \quad \tau\mathbf{J}_{\mathbf{T}}\dot{\theta}_{\mathbf{T}} = \tau\dot{\mathbf{x}}_{\mathbf{T}} + \mathbf{x}_T - \mathbf{K}(\theta_{\mathbf{T}}). \tag{4.16}$$

Here, $\mathbf{J}_{\mathbf{T}}$ is the Jacobian of the kinematic function at $\theta_{\mathbf{T}}$. The term $\mathbf{x}_T - \mathbf{K}(\theta_{\mathbf{T}})$ is there to avoid a numerical drift, which would bring $\mathbf{K}(\theta_{\mathbf{T}})$ away from $\mathbf{x}_T$. Again, it is a positive quadratic optimization problem subjected to linear constraints and thus has a single minimum. Using the Lagrange multipliers for solving this problem, we have

$$\frac{d}{\partial\dot{\theta}_{\mathbf{T}}}\frac{1}{2}(\theta_{\mathbf{T}} + \tau\dot{\theta}_{\mathbf{T}} - \theta)^T(\theta_{\mathbf{T}} + \tau\dot{\theta}_{\mathbf{T}} - \theta) - \lambda^T\Big(\tau\mathbf{J}_{\mathbf{T}}\dot{\theta}_{\mathbf{T}} - \tau\dot{\mathbf{x}}_{\mathbf{T}} - \big(\mathbf{x}_T - \mathbf{K}(\theta_{\mathbf{T}})\big)\Big) = 0 \tag{4.17}$$

Thus

$$\begin{aligned}
0 &= \theta_{\mathbf{T}} + \tau\dot{\theta}_{\mathbf{T}} - \theta - \tau{\mathbf{J}_{\mathbf{T}}}^T\lambda \\
\Rightarrow\ \tau\dot{\theta}_{\mathbf{T}} &= \tau{\mathbf{J}_{\mathbf{T}}}^T\lambda - \theta_{\mathbf{T}} + \theta \\
\Rightarrow\ \mathbf{J}_{\mathbf{T}}(\tau{\mathbf{J}_{\mathbf{T}}}^T\lambda - \theta_{\mathbf{T}} + \theta) &= \tau\dot{\mathbf{x}}_{\mathbf{T}} + \mathbf{x}_T - \mathbf{K}(\theta_{\mathbf{T}}) \\
\Rightarrow\ \tau\lambda &= (\mathbf{J}_{\mathbf{T}}{\mathbf{J}_{\mathbf{T}}}^T)^{-1}\big(\tau\dot{\mathbf{x}}_{\mathbf{T}} + \mathbf{x}_T - \mathbf{K}(\theta_{\mathbf{T}}) + \mathbf{J}_{\mathbf{T}}(\theta_{\mathbf{T}} - \theta)\big) \\
\Rightarrow\ \tau\dot{\theta}_{\mathbf{T}} &= {\mathbf{J}_{\mathbf{T}}}^T(\mathbf{J}_{\mathbf{T}}{\mathbf{J}_{\mathbf{T}}}^T)^{-1}\big(\tau\dot{\mathbf{x}}_{\mathbf{T}} + \mathbf{x}_T - \mathbf{K}(\theta_{\mathbf{T}}) + \mathbf{J}_{\mathbf{T}}(\theta_{\mathbf{T}} - \theta)\big) \\
&\quad -\theta_{\mathbf{T}} + \theta
\end{aligned}$$

This results in

$$\begin{aligned}
\tau\dot{\theta}_{\mathbf{T}} &= \big({\mathbf{J}_{\mathbf{T}}}^T(\mathbf{J}_{\mathbf{T}}{\mathbf{J}_{\mathbf{T}}}^T)^{-1}\mathbf{J}_{\mathbf{T}} - \mathbf{I}_{\mathbf{n}}\big)(\theta_{\mathbf{T}} - \theta) + \\
&\quad {\mathbf{J}_{\mathbf{T}}}^T(\mathbf{J}_{\mathbf{T}}{\mathbf{J}_{\mathbf{T}}}^T)^{-1}\big(\tau\dot{\mathbf{x}}_{\mathbf{T}} + \mathbf{x}_T - \mathbf{K}(\theta_{\mathbf{T}})\big),
\end{aligned} \tag{4.18}$$

where $\mathbf{I}_{\mathbf{n}}$ denotes the identity matrix of size $n \times n$.

This amounts to performing a gradient descent on the squared Euclidean distance to the actual arm configuration in angle space. As such, $\theta_{\mathbf{T}}$ may end up in a local minimum if Θ_T is disjoint (see [Burdick, 1989]) and if the initial $\theta_{\mathbf{T}}$ is too far from the optimal one. Thus, in order to find an initial value for $\theta_{\mathbf{T}}$ we sample Θ_T using a geometrical or CCD (cf Section 2.3.3) inverse kinematics algorithm and take the value closest to θ. This initialization is performed again when a sudden target displacement occurs.

[1] Here we omit the time index t to lighten the notation.

4.4.4 Summary

Putting together the elements described above, results in the following dynamical system. At each time step inconsistent velocities $\dot{\theta}_t^d$ and $\dot{\mathbf{x}}_t^d$ are obtained by each of the sub-controllers, using the Euler approximation of the VITE-like system:

$$\begin{aligned}
\ddot{\theta}_t^d &= \alpha(-\dot{\theta}_t + \beta(\theta_{\mathbf{T}} - \theta_t)) && (4.19)\\
\dot{\theta}_t^d &= \dot{\theta}_t + \tau\ddot{\theta}_t^d && (4.20)\\
\ddot{\mathbf{x}}_t^d &= \alpha(-\dot{\mathbf{x}}_t + \beta(\mathbf{x}_T - \mathbf{x}_t)) && (4.21)\\
\dot{\mathbf{x}}_t^d &= \dot{\mathbf{x}} + \tau\ddot{\mathbf{x}}_t^d && (4.22)
\end{aligned}$$

Coherence constraints are then enforced using (4.13):

$$\begin{aligned}
\dot{\theta}_{t+1} &= \dot{\theta}_t^d + \mathbf{W}_\theta^{-\mathbf{1}}\mathbf{J}^T(\mathbf{W}_\mathbf{x}^{-\mathbf{1}} + \mathbf{J}\mathbf{W}_\theta^{-\mathbf{1}}\mathbf{J}^T)^{-1}(\dot{\mathbf{x}}_t^d - \mathbf{J}\dot{\theta}_t^d) && (4.23)\\
\dot{\mathbf{x}}_{t+1} &= \mathbf{J}\dot{\theta}_{t+1} && (4.24)\\
\theta_{t+1} &= \theta_t + \tau\dot{\theta}_{t+1} && (4.25)\\
\mathbf{x}_{t+1} &= \mathbf{K}(\theta_{t+1}). && (4.26)
\end{aligned}$$

Finally, the target arm configuration $\theta_{\mathbf{T}}$ is updated using (4.18):

$$\begin{aligned}
\theta_{\mathbf{T}} &= \theta_{\mathbf{T}} + \big({\mathbf{J_T}}^T(\mathbf{J_T}{\mathbf{J_T}}^T)^{-1}\mathbf{J_T} - \mathbf{I_n}\big)(\theta_{\mathbf{T}} - \theta_{t+1}) + \\
&\quad {\mathbf{J_T}}^T(\mathbf{J_T}{\mathbf{J_T}}^T)^{-1}\big(\tau\dot{\mathbf{x}}_{\mathbf{T}} + \mathbf{x}_T - \mathbf{K}(\theta_{\mathbf{T}})\big). && (4.27)
\end{aligned}$$

As the robot is controlled in position, θ_{t+1} is sent to the robot at each time step. Those three sets of equations are iterated until the target is reached.

4.4.5 Joint limit avoidance

In this section we explore an application offered by an on-line modulation of the sub-controller weights. We show how interesting properties can be obtained by a judicious weight modulation policy. In particular, we focus on joint limit avoidance. When controlling a robotic arm, it is always important to avoid bumping into the joint boundaries. Indeed, this is needed to avoid jerky movements and to avoid restraining the movements of the arm.
The solution presented here can be applied when the joint angle workspace is convex. This is the case when the joint angle workspace is specified by a fixed upper and lower bound for each of the joint angles. The hyper-volume defined in this way is a parallelepiped, and is hence convex. This implies that when reaching a target from a starting position, the joint angle sub-controller will never bring the system to a joint angle boundary because there is no trajectory overshoot. This property can be exploited to effectively avoid the joint boundaries, by switching to the joint angle sub-controller when approaching the joint limit. So joint angle limits can be easily avoided by making the weights $\mathbf{W}_\theta$ and $\mathbf{W}_\mathbf{x}$ dependent on the robotic arm configuration. As the arm gets closer to one of the joint limits, say θ^i, the corresponding element $w^{\theta,i}$ of the matrix $\mathbf{W}_\theta$ gets bigger to eventually have a ratio $w^x/w^{\theta,i}$ equal to zero, which amounts to have a pure joint angle controller and thus avoid the joint limit. This can be achieved by making the weight ratio $w_t^x/w_t^{\theta,i}$ depend on the joint angle position

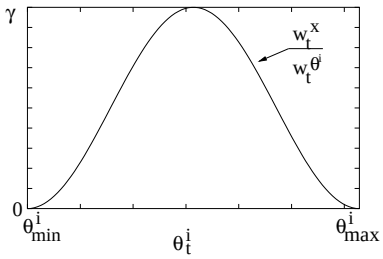


Figure 4.4: The behavior of the function described in (4.28).

θ_t^i as follows.

$$\frac{w_t^x}{w_t^{\theta,i}} = \frac{1}{2}\gamma\Big(- \cos\big(2\pi \cdot \frac{\theta_t^i - \theta_{min}^i}{\theta_{max}^i - \theta_{min}^i}\big) + 1\Big), \tag{4.28}$$

where θ_{min}^i and θ_{max}^i are the joint angle boundaries and γ is a constant setting the maximum value for $w^x/w^{\theta,i}$. The right-hand side of this equation is plotted on Fig. 4.4. By applying this equation, the control is purely angular ($w^x = 0$) when the system is approaching the joint boundary, thus avoiding it.
For the implementation of this modulation, it is again simpler to use (4.13) instead of (4.14). In fact, working with the inverse of the weight matrices, one can set the $\mathbf{W_x^{-1}}$ to identity and the diagonal elements of $\mathbf{W_\theta^{-1}}$ can be computed according to the right-hand side of (4.28).

This method is quite different from other joint limit avoidance methods such as [Liégeois, 1977], [Chan and Dubey, 1995] or [Chaumette and Marchand, 2001]. Those authors typically consider a main task, given by a end-effector trajectory and use the redundant degrees of freedom to optimize a secondary task, in this case joint limit avoidance. This secondary task is only performed if it does not disrupt the main task. Because we use the dynamical system approach, we do not have such a constraint. In fact, our joint angle avoidance method does influence the end-effector trajectory, but this is of no concern to us, as the target will nevertheless be reached, thanks to the attractor attached to the target.

4.4.6 Robustness to Singularities

We have mentioned before that our controller can be seen as a generalization of the DLS inverse method. The DLS inverse $\mathbf{J}^*$ of the Jacobian matrix $\mathbf{J}$ is defined by

$$\mathbf{J}^* = (\epsilon\mathbf{I_n} + \mathbf{J}^T\mathbf{J})^{-1}\mathbf{J} = \mathbf{J}^T(\epsilon\mathbf{I_m} + \mathbf{J}\mathbf{J}^T)^{-1}, \tag{4.29}$$

where $\epsilon > 0$, $\mathbf{I_n}$ and $\mathbf{I_m}$ are the identity matrices of dimension n and m respectively. This inverse was introduced by [Wampler, 1986] and [Nakamura and Hanafusa, 1986] as an alternative to the Moore-Penrose pseudo-inverse solution [Whitney, 1969]. The advantage of this method is that it effectively avoids singularities because $\mathbf{J}^*$ is always defined. This comes at the cost of precision in the tracking of a desired end-effector velocity.

Equations 4.13 and 4.14 make essentially use of this DLS inverse (up to a diagonal matrix multiplication) and therefore our controller also avoids singularities. Because the matrices

$$\mathbf{J}_n^{\#} = \mathbf{W}_\theta + \mathbf{J}^T\mathbf{W}_\mathbf{x}\mathbf{J} \tag{4.30}$$
$$\mathbf{J}_m^{\#} = \mathbf{W}_\mathbf{x}^{-1} + \mathbf{J}\mathbf{W}_\theta^{-1}\mathbf{J}^T \tag{4.31}$$

are positive definite (as long as $\mathbf{W}_\theta$ and $\mathbf{W}_\mathbf{x}^{-1}$ are positive definite), they are never singular. Moreover, an upper bound on the condition number of $\mathbf{J}_m^{\#}$ is given by:

$$\mathrm{cond}(\mathbf{J}_m^{\#}) \leq 1 + \gamma \cdot \max_\theta(\sigma_1^2(\theta)), \tag{4.32}$$

where $\sigma_1(\theta)$ is the biggest singular value of $\mathbf{J}(\theta)$. This formula can give a maximal value for the variable γ for a given kinematic function $\mathbf{K}$, over which computational problems may arise. This is an additional advantage of using (4.13) and the corresponding weight parametrization.

4.4.7 Convergence

As shown in Appendix A.3, each of the sub-controllers has one single fixed point. This fixed point is the target location and acts as an attractor. This, however, does not ensure that the controller as a whole, i.e. the combination of the two sub-controllers, also has a unique fixed point. In fact, the nonlinear interaction between the two sub-controllers may give rise to the appearance of spurious fixed points. Those appear when the two controllers exactly cancel each other. As can be seen in (4.14), this is the case when

$$\mathbf{W}_\theta\dot{\theta}_t^d + \mathbf{J_t}^T\mathbf{W}_\mathbf{x}\dot{\mathbf{x}}_t^d = \mathbf{0} \tag{4.33}$$
$$\dot{\theta}_t = \mathbf{0} \tag{4.34}$$
$$\dot{\mathbf{x}}_t = \mathbf{0} \tag{4.35}$$

which means that

$$\theta_\mathbf{T} - \theta_t = -\mathbf{W}_\theta^{-1}\mathbf{J_t}\mathbf{W}_\mathbf{x}\big(\mathbf{K}(\theta_\mathbf{T}) - \mathbf{K}(\theta_t)\big). \tag{4.36}$$

The existence of configurations θ_t and $\theta_\mathbf{T}$ satisfying this equation depends on the function $\mathbf{K}$ and of the weights $\mathbf{W}_\mathbf{x}$ and $\mathbf{W}_\theta$. For $\mathbf{W}_\mathbf{x} = \mathbf{0}$, there are no such points apart those for which $\theta = \theta_\mathbf{T}$, i.e., the only fixed point is the target. But for other values of the weights, those spurious fixed points may well appear. Hence the parameter γ of (4.28) is a bifurcation parameter of the system. Section 4.6.1 presents simulation results confirming this hypothesis.
The presence or absence of cycles in the system is harder to prove, especially considering the weight dynamics. Simulation results seem to indicate that there are no such cycles.
Both spurious attractors and cycles can be avoided by making γ decay slowly. This way, the system will necessarily cross the bifurcation threshold at some point and reach the target.

4.4.8 Stability

The VITE-like dynamical system is a linear system with a single attractor. As such it is asymptotically stable. Because the system has no singularity, the

velocities remain bounded. The joint angles also remain bounded by the joint limit avoidance mechanism. The whole system thus remains bounded, as long as the input, i.e., the target location is bounded. Since the system needs an input, it must be provided with a default input in case the target is not tracked. It can be for example the last target position, the current manipulator configuration or a "rest" position.

4.5 Implementation

This controller was implemented for the control manipulators with an arbitrary number of joints. For the experiments, the arms of the Hoap humanoid robot of Fujitsu (see Fig. 4.5), as well as the iCub robot of the RobotCub consortium were used (see Fig. 3.15).
The Hoap arms have four DoFs assembled in a kinematic chain depicted in Fig 4.6. We describe the end-effector location by its three spatial coordinates and discard its orientation. We thus have $n = 4$ and $m = 3$, which means that we are dealing with a redundant manipulator.
For controlling the Hoap, the experimental setting comprises a stereo-vision system composed of two cameras, the robot and two personal computers running Linux. The stereo-vision system and the controller run on the first computer, while the second computer is used as an interface with the robot. Using color recognition, the stereo-vision system provides the target location to the controller. The controller updates the simulated joint angle trajectories and gives it to the second computer at a fixed rate of 20 Hz. The second computer, which runs Real-Time Linux, performs a linear interpolation and sends the robot the positions at a rate of 1 kHz. The robot has an on-board high-gain feedback controller, i.e. we control the robot in position and the on-board controller takes care of the trajectory tracking.

The arms of the iCub are anthropomorphic and similarly to human arms, they have seven DoFs. A model of the iCub right arm showing its DoFs is presented in Fig. 4.7. They thus make a redundant manipulator even when considering the position and orientation. Both versions have been implemented, with position only ($m = 3$) and with position and orientation ($m = 6$). Position control is performed using the Yarp [Metta et al., 2006] middleware. Again, processing is distributed on two computers, one performing visual tracking of the target, and one performing the control of the robot.

Most of the results presented below were obtained with the Hoap robot. This is due first to the availability of the robot in the lab, whereas the iCub robot is located in Genova, and was only ready for use by the end of this thesis. Controlling a manipulator with more DoFs requires additional computing ressources, but it does not change much in terms of implementation, and effectivity of the principles and computations involved.

Figure 4.5: The Hoap2 robot reaching for a target tracked by a stereo-vision system.

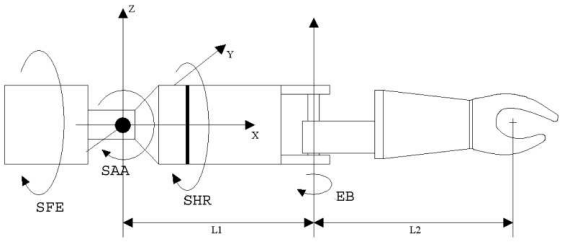


Figure 4.6: A schematic representation of the Hoap arm. It has four DoFs: SFE, SAA, SHR, EB.

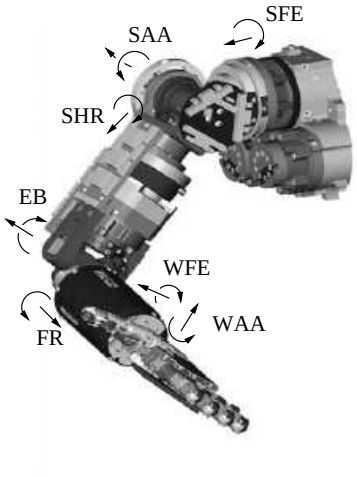


Figure 4.7: A schematic representation of the iCub arm. It has seven DoFs: SFE, SAA, SHR, EB, FR, WFE, WAA. Adapted from [Tsagarakis et al., 2007].

4.6 Results

4.6.1 Convergence properties

In order to see when and where spurious attractors appear, we performed a simulative analysis of the system. We simulated two millions trajectories performed with the Hoap. Those trajectories are given by randomly and uniformly sampled starting configurations and target locations. Each trajectory was simulated for various values of γ. The ratio of trajectories that did not reach the target is plotted in Fig 4.8. One can see that for small γ, there are no spurious attractors and all trajectories reach their target. But when γ gets bigger, spurious attractors appear and some trajectories cannot reach their goal. Fig. 4.9 shows the location of the targets that could not be reached. They almost all lie in the vicinity of lower boundary of the workspace. The region of unreachable targets is centered around a fully downward stretched vertical arm position. As illustrated in Fig 4.9, this region grows with γ. It must be noted that the points shown by this figure are not always unreachable, they are unreachable by a non-empty set of starting configurations. Fig. 4.10 shows the location of the unreachable target and the spurious attractor where the trajectory is trapped in joint angle space. One sees that for particular locations of the target joint angle (x-axis), the trajectory might be brought to a spurious attractor (y-axis).

4.6.2 Point-to-point reaching trajectories

Our controller can perform accurate point-to-point reaching motions. An example is shown in Fig. 4.11 which shows a reaching movement in three dimensions, performed in simulation and with the real robot. One sees that the robot matches well the simulated trajectory. Thanks to this close match between the simulated and the real trajectory, one can assume that the results obtained through simulations carry over to the real robot trajectories. Consequently, un-

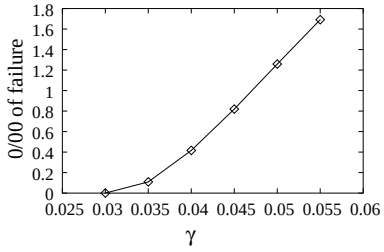


Figure 4.8: The ratio of reaching failure for increasing values of γ. For a very small γ all reaching tasks could be successfully performed. But when $\gamma = 0.035$, spurious attractors appear.

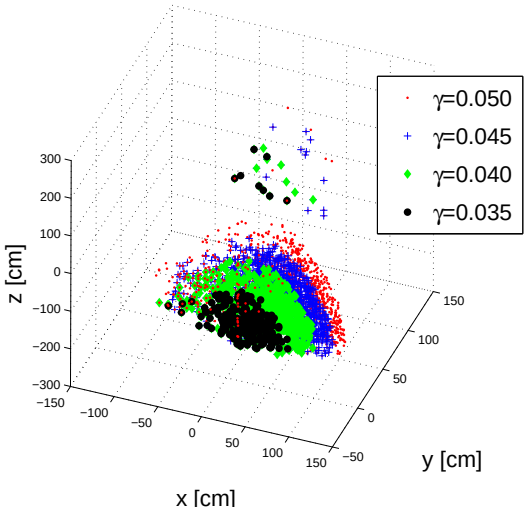


Figure 4.9: The set of unreachable targets, in Cartesian space for different values of γ. The origin lies on the shoulder.

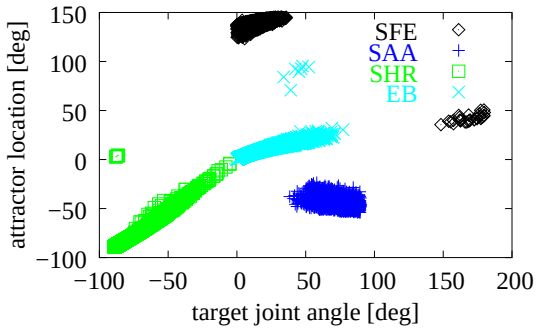


Figure 4.10: The attractor location depending on the joint angle target. This graph represents an eight-dimensional space, because there are two dimensions for each angle (the target and the attractor value). SFE, SAA, SHR and EB correspond to the four DoFs of the arm. Here, $\gamma = 0.05$.

less specified otherwise, the experiments presented in the following sections are done in simulations.
Fig. 4.12 shows additional properties of the movement displayed in Fig 4.11. On the right, one sees the bell-shaped end-effector velocity profile and the joint angle velocity profiles. On the left, one sees how the system handles the target arm configuration redundancy. In joint space, the system aims at the closest joint angle configuration, which correspond to the target in Cartesian space.
Quasi-horizontal movements in front of the robot are depicted in Figure 4.13. The controller produces quasi-straight trajectories, when the reaching is performed in the workspace center. This is clearly because the controller produces trajectories which are some sort of compromise between straight lines in end-effector location space (produced by the end-effector location sub-controller) and straight lines in joint angle space (produced by the joint angle sub-controller). Moreover, the velocity profiles are bell-shaped and smooth, due to the use of the VITE-like dynamical system. Bell-shaped velocity profiles and quasi-straight hand paths are typical of human reaching movements [Morasso, 1981]. Moreover, it has been suggested that human movements are the result of "a compromise between a straight line in workspace and a straight line in joint space" [Cruse and Brüwer, 1987].

The same controller can be used to reach a given position and a given orientation. In this case the Cartesian space comprises six dimensions, three for the end-effector position and three for the end-effector orientation. The orientation is uniquely described by a rotation vector $\mathbf{b}$ as described in Appendix A.2.1. This vector describes the rotation with respect to the orientation of the end-effector when the arm is in a zero angle configuration. The corresponding weights in the $\mathbf{W_x}$ matrix are set to small values (typically 1), because it seems not crucial to have a straight path in orientation space. Moreover the orientation space is closed (it "wraps around itself" as a torus in 2D), so there is no need to constrain the end-effector orientation path too much. A example of such a reaching motion with target position and orientation is provided in Fig. 4.14, using the iCub arm. For the sake of clarity, only two dimensions for end-effector position and for end-effector orientation are shown. Again, one sees that the end-effector position path is quite straight. This is not the case for the end-effector orientation path, as mentioned above. On the right one sees the joint angle trajectories for the seven joints of the iCub arm.
.

4.6.3 Singularity avoidance

As explained in section 4.4.6, the controller presented here avoids singularities. By this, we do not merely mean that the controller avoids singular configurations (as in [Baillieul, 1985]), but that there are no singularities for our controller, whatever the configuration. This fact is illustrated here on a task involving reaching to and from points liable to produce a singularity in traditional controllers. This happens when the Jacobian matrix of the kinematic function is degenerate, i.e when its rank is smaller than m. The starting position is set as the arm fully stretched downward and the target position is with the arm fully and horizontally stretched to the side. For those two positions, the Moore-Penrose pseudo-inverse of the Jacobian matrix cannot be computed and hence

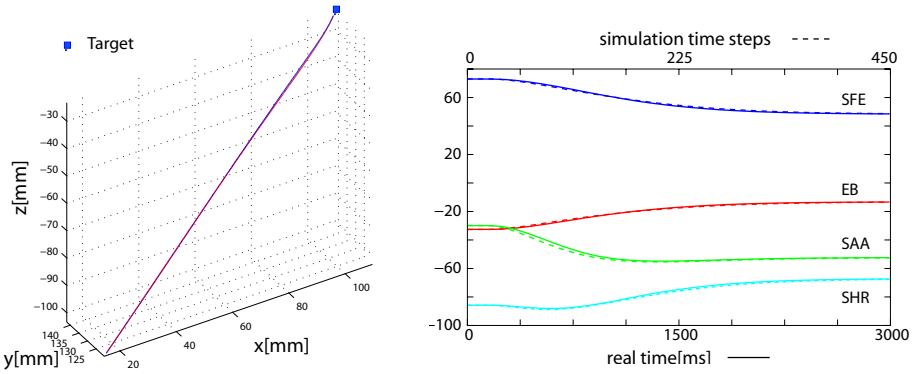


Figure 4.11: Comparison between simulated movement and real movements performed with the Hoap robot. Left: a 3D reaching trajectory, in simulation (dashed line) and on the robot (solid line). Right: the corresponding joint angle trajectories in simulation (dashed) and as measured on the robot sensors (solid). Angles are given in degrees. The delay was removed to ease the comparison. One simulation time step corresponds to roughly 7 milliseconds in real time.

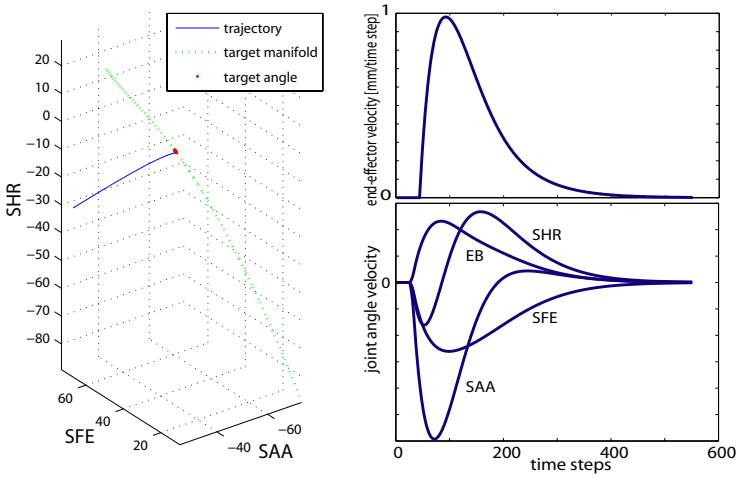


Figure 4.12: Left: the redundancy resolution for the movement depicted in Fig. 4.11. The 1D target manifold Θ_T is represented by the dotted line and one can see that the joint angle trajectory (solid line) aims at the closest point of Θ_T. This point (shown with the dot) moves on Θ_T according to 4.18. Right: the end-effector velocity (above) and the joint angle velocities (below) for the same movement.

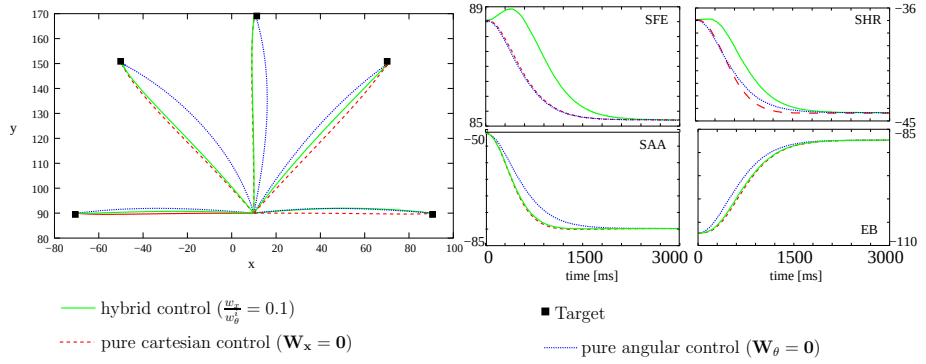


Figure 4.13: Left: Hoap reaching trajectories for various weight configurations. The trajectories lie in the center of the workspace. Right: the joint angle trajectories (measured on the robot sensors) for the rightmost movement.

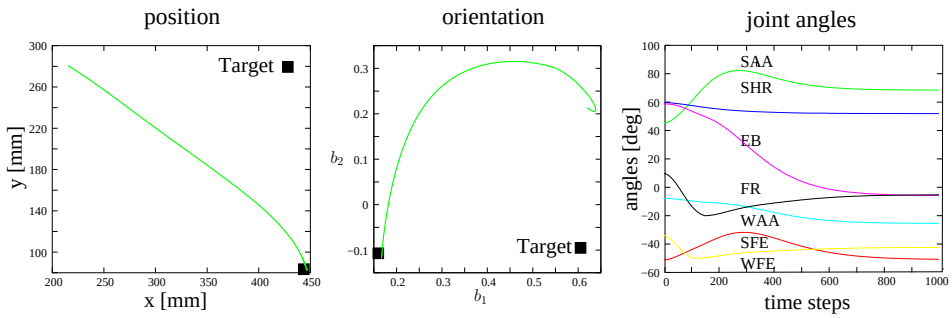


Figure 4.14: A reaching trajectory performed with the iCub robot arm comprising 7 DoFs. Left: the horizontal projection of the end-effector trajectory. Center: the 2D projection of the end-effector orientation trajectory. Right: joint angle trajectories for the seven joints. Joint labels refer to Fig 4.7

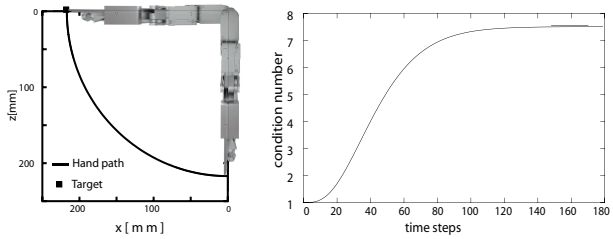


Figure 4.15: Left: the trajectory for reaching to a potentially singular point. There is no singularity. Right: the evolution of matrix condition number corresponding to this trajectory. The inverted matrices remain well-conditioned throughout the movement.

the task cannot be accomplished using that method. This limitation, however, does not apply to the hybrid controller presented here, as can be seen in Fig. 4.15. This figure shows the hand path trajectory obtained when performing this task and the condition number of the matrix which is inverted at every time step. One sees that this condition number remains in the first order of magnitude, which means that the inversion can be reliably performed without numerical instability.

4.6.4 Joint limit avoidance

The joint limit avoidance method described above prevents that the robotic arm reaches its joint angle boundaries. An example is given in Fig. 4.16, which shows the end-effector and joint angle trajectories, with and without the joint limit avoidance mechanism. In this example the robot is asked to reach from behind his neck to the front. One sees that the weight modulation forces the system to remain within the joint boundaries. Without the joint limit avoidance mechanism, the boundaries are reached for all four joint angles. The following table shows how close the trajectories get from the workspace boundaries.

joint	SFE	SAA	SHR	EB
lower boundary	-90	-180	-90	-115
without avoidance	-90	-180	-90	-115
with avoidance	-83.5	-164.1	-89.9	-107.2

For the kind of tasks displayed in Fig. 4.16, the joint limit avoidance mechanism has a significant influence on the end-effector trajectories. This may be a disadvantage in some cases, but if there is no particular constraint on the end-effector it seems adequate to take advantage of this freedom.

4.6.5 Robustness to perturbations

The controller described here can essentially be understood as a dynamical system having the target as an attractor. It therefore comes as no surprise that the system is robust to perturbations. Here, two kinds of perturbations are considered: a sudden target displacement and a transient external force applied on the manipulator. The target displacement example is illustrated in

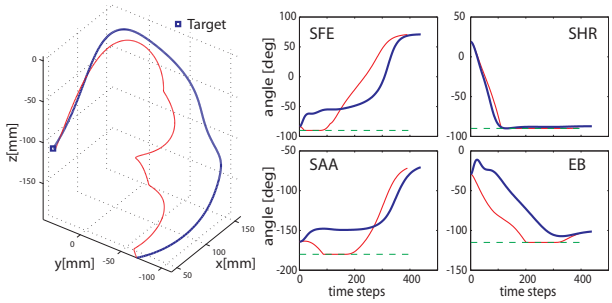


Figure 4.16: A reaching trajectory, with and without the joint limit avoidance mechanism. Left: end-effector trajectories. Right: Joint angle trajectories. Without the joint limit avoidance mechanism (thin line) the system bumps into the joint limits (shown with a dashed line). This does not occur with the joint limit avoidance mechanism (thick lines). The weight ratio (and γ respectively) is 0.05. SFE, SAA, SHR and EB refer to the four DoFs, as displayed in Fig. 4.6.

Fig. 4.17. One sees that by moving the target, one simply moves the attractor of the dynamical system. Hence, the system reaches the target while retaining a continuous velocity, because it is a second-order system.
The effect of an external perturbation acting on the manipulator is depicted in Fig. 4.18. Again, due the attractor nature of the system, this perturbation is handled on-line in an appropriate manner. The system smoothly adapts its trajectory to reach the target.

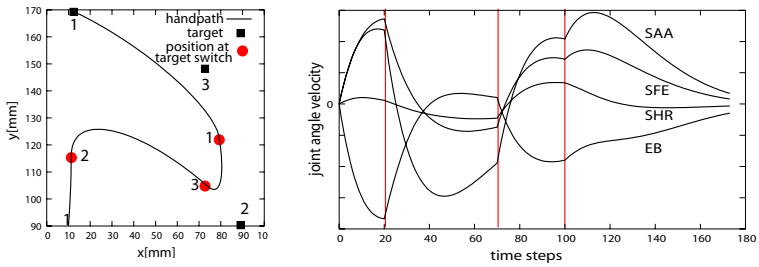


Figure 4.17: The end-effector trajectory altered by sudden target displacements. Left: end-effector trajectory. The system is at the location indicated by the circles when the target suddenly moves to square with the corresponding number. Right: the corresponding joint angle velocities. The vertical bars indicate the occurrences of a target displacement.

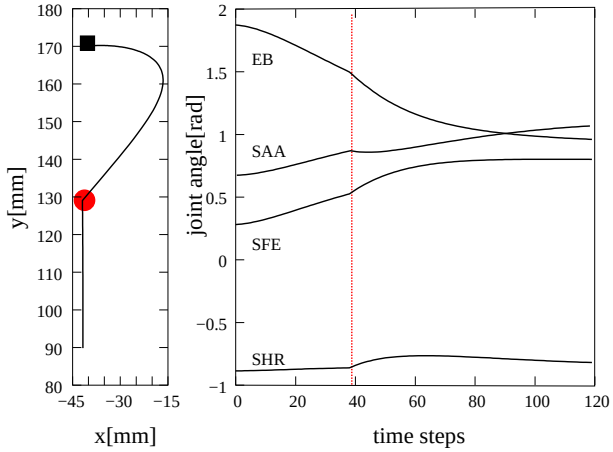


Figure 4.18: Robustness to external perturbations. The end-effector trajectory is depicted on the left and the joint angle trajectories appear on the right. The location and time of the perturbation are indicated by a circle and a vertical bar respectively.

4.7 Discussion

In the preceding sections, we described a robotic manipulator controller based on two basic principles drawn from the studies of human reaching movements, multi-referential movement representation and dynamical system control. The controller comprises two VITE dynamical systems, one acting on the end-effector described in a body-centered Cartesian frame of reference and the other acting on the joint angle arm configuration. The controller is interesting both for robotics and for biological control modeling.

4.7.1 Strengths of the controller

From a pure roboticist perspective, the multi-referential controller presented here has several advantages over classical controllers. First, it does not have any singularity because it uses a generalized version of the DLS inverse which has been shown to avoid the singularity problem [Wampler, 1986]. The inverse in (4.13) can always be computed as long as $\mathbf{W_x^{-1}}$ is positive definite because $\mathbf{J_t W_\theta^{-1} J_t}^T$ is positive semi-definite. Another advantage of the controller is that it allows a simple and elegant solution to the joint limit avoidance problem. Our results show that this method is effective and yields smooth and short end-effector trajectories. Note that this method makes the assumption that the joint angle workspace is convex, which is generally the case in the absence of obstacles. If there are obstacles, or in order to prevent self collisions, a classical potential field method [Khatib, 1985] can easily be (and has been) integrated into our dynamical system framework.
Finally, the controller is robust to unexpected changes in the target location and smoothly adapts its trajectory accordingly. This is due to the robustness of the dynamical system underlying the controller.

4.7.2 Biological relevance

This robotic controller draws its inspiration from two principles purportedly underlying biological arm control. Those principles are on one hand a dynamical system control and on the other hand a multi-referential control. They have been argued to lie at the core of human movement control [Kelso, 1995, Paillard, 1991]. The particular choice of dynamical system (the VITE-like system) and of frames-of-reference (joint angles and end-effector locations) do have some biological plausibility and have also been suggested to underlie human reaching control [Bullock and Grossberg, 1988, Cruse and Brüwer, 1987, Carrozzo and Lacquaniti, 1994]. The resulting movements share some of their properties with human reaching movements, either due to the VITE model such as the speed-accuracy trade-off and the speed-to-distance proportionality, or due to the multi-referential control, such as the quasi-straight paths. However, we do not claim that the VITE-like system (4.3) is the one used by humans. In fact, the rather sharp initial accelerations obtained are not typical of human reaching motions. Indeed much of the similarity between the velocity profiles of the human movements and the VITE-generated movements is due the $G(t)$ function in 4.1. By dropping $G(t)$ in (4.3), we simplify the system at the cost of reduced similarity with human velocity profiles, thus favouring the roboticist over the naturalist. Of course, other directions can be taken, as in [Petreska and Billard, 2009], where the VITE-like equation is combined with force-fields to closely reproduce human reaching trajectories. Anyhow, it is quite possible that the human control system uses some kind of similar spring-like system with damping as suggested by [Hogan, 1985]. This could be done by taking advantage of the spring and damping properties of the muscles.

In the controversy between a joint and a Cartesian control of reaching movements, it has been noted by [Desmurget et al., 1997] that experiments in favor of a Cartesian control were performed in constrained settings, for example by constraining the movements on an horizontal plane with a table or a manipulandum, as in [Morasso, 1981, Abend et al., 1982]. On the contrary, experiments arguing for a joint angle control were performed in a free setting, as in [Soechting and Lacquaniti, 1981, Magescas and Prablanc, 2006]. The ability of our controller to balance between a joint angle and a cartesian controller could explain this observation. It could be that in the case of movements for which constraints are provided visually in the Cartesian space (the subject sees the plane of the table), more influence is given to the Cartesian controller, while the joint angle controller has influence for unconstrained movements, or movements constrained in joint angle.

The controller presented here is kinematic only and does not take the dynamical properties of the robot limbs into account. This is why in the present implementation the specified position is fed into a PID-like controller. The exact nature of the relationship between the controller and the actual limbs in biological systems is still unclear, but the use of internal models to account for the limb dynamics has been hypothesized [Wolpert et al., 1995, Ariff et al., 2002]. This could be a further extension of our system.

It is interesting to note that the way the target redundancy is handled

(see Section 4.4.3) results in a dependence between final and initial postures, which can also be found in human reaching motions [Soechting et al., 1995, Desmurget et al., 1998a]. In other words, this model presents a solution to the "equifinality problem". Although disputed [Feldman and Latash, 2005], this point has often be used to argue against the Equilibrium Point hypothesis [Desmurget et al., 1998b, Braun and Wolpert, 2007]. This argument comes from the observation that a person reaching to a particular target, will reach different final arm configurations when starting from different initial positions. Thus, if the arm is controlled by an attractor dynamical system as assumed by the Equilibrium Point hypothesis, the arm should always reach this attractor, and thus the same final arm configuration. The controller described in this chapter shows that this is not necessarily the case. With our controller, starting from different initial position, will bring the robot to different final configurations. This is due to the fact that the attractor in the configuration space is always updated as a function of the arm position. So what we have is an attractor manifold in joint space and an attractor point (or possibly also a manifold) in Cartesian space. This shows that the "equifinality problem" cannot be taken as evidence against the equilibrium point hypothesis, when admitting that the "equilibrium point " can be an "equilibrium manifold", or that it depends on the actual configuration of the arm. This "equilibrium manifold" can seen as the Uncontrolled Manifold [Scholz and Schöner, 1999] of the reaching task.

4.8 Summary

In this chapter, we have extended the model of peripersonal space representation presented in the preceding chapter to include a motor ability. The motor control algorithm developed here is inspired by two principles thought to lie at the heart of biological movement control, namely dynamical system control and multimodal control. The suggested controller consists of two PD controllers running in parallel on different yet redundant variables, joint angles on one hand and visual end-effector location on the other hand. The mapping between those two sets of variables is given by the adaptive model of the body-schema described in the previous chapter. This allows the robot to control its hand but also tools that it may hold.

The resulting controller has several advantages over other, traditional robotic controllers such as the absence of singularity, a simple solution to the joint limit avoidance problem, and a robustness to perturbations. Those advantages were illustrated in an experimental setup involving a humanoid robot.

Moreover, the model suggested here yields several insights into the biological control of motion. Among them, it emphasizes the possible advantages of using multimodal and redundant representations of the movement, and suggests a solution to the equifinality problem, often used to argue against dynamical system theories of movement control.

Chapter 5

Learning motor primitives from examples

> Et tout ça se pellicule et tout ça se met en boîte.
>
> Leo Ferré

5.1 Introduction

In the previous chapters we have seen how a humanoid robot can learn its body schema and control simple reaching movements, with or without controlling the final hand orientation. However, most purposeful movements are not plain reaching movements. They usually bear some additional constraints such avoiding obstacles or approaching the target from a particular direction. To control such movements (an many others), it has been suggested that humans use so-called *motor primitives*. In this chapter, we show how the reaching controller presented in the previous chapter can be extended to enable a humanoid robot learn how to perform such constraints movements from demonstrations. Part of this work has been described in [Hersch et al., 2006] and [Hersch et al., 2008a].

5.2 Motor primitives in primates

In this section, we introduce the notion of *motor primitives* or *synergies*, that has been invoked to explain the adaptivity of motor control. More detailed reviews and accounts of those concepts can be found in [Latash, 2008, Turvey, 2007, Flash and Hochner, 2005, Konczak, 2005].

The main question that brings forth the idea of motor primitives has been formulated by Bernstein [Bernstein, 1967, Bernstein, 1996]. How can the control of the high-dimensional musculo-skeletal system be reduced to a few control variables? Indeed, it seems that the brain does usually not directly control each muscle. Rather, it sends high level commands to the spinal system, which then somehow generates the low level commands sent to each muscle. Within this hierarchical control, motor primitives form the low level control. They have

been defined as some kind of building blocks of motion, that can be combined together for a given task. This definition is rather vague and no clear consensus has emerged about the nature of those building blocks. For some authors such as [D'Avella et al., 2003], motor synergies are profiles $f_i(t)$ of muscle activities that can be captured through electromyographic recordings. Those profiles are then be shifted in time and linearly combined to generate a movement with muscle activity $\sum_i a_i f_i(t - d_i)$. In this view, instead of controlling all the trajectories the brain just controls the scaling coefficients a_i and the time delays d_i. For other authors [Mussa-Ivaldi and Bizzi, 2000] primitives are position and velocity dependent force fields generated by the spinal system, that bring the limb to a particular location [Giszter et al., 1993]. Those force fields are then vectorially added by higher-level commands to move the limbs. An extension of this model with proven control properties, which make the force field dependent also on the initial position, has been suggested by [Nori and Frezza, 2005]. These definitions are still too restrictive to [Turvey, 2007], who suggests that synergies are self-preserving, functionally defined dynamical systems involving motor as well as perceptual aspects. While the case of rhythmic movements such as locomotion, Central Pattern Generators (CPGs) in the spinal chord have been identified and modeled as coupled oscillators [Ijspeert, 2008], for non-rhythmic movements, a satisfying model of synergies still remains to be found. An attempt to encapsulate motor patterns in more general sensorimotor structures is the *Schema Theory* [Arbib, 2002], where motor schemata can be combined to perceptual schemata to form some sort of "control programs" implemented at the neural level that accomplish a specific task. The exact variables involved in those synergies and schemas seem to depend on the task demand.
In sum, there is little agreement on the level on which those motor primitives take place, and it may well be that multiple levels of control coexist.

5.3 Objective

Given the poor understanding of the concept of motor primitives and synergies, apparent in the lack of consensus among scholars on those notions, it would be a little bit pretentious to aim at implementing humanoid motor primitives comparable to human motor primitives within the scope of this chapter. More modestly, this chapter presents how the framework developed in the preceding chapters can be extended to allow a robot to perform somewhat more sophisticated goal-directed movements. Following [Ijspeert et al., 2002], we take the liberty to call them motor primitives. Those movements are learned by the robot, which should be able to reproduce them despite different initial conditions and possible changes occurring during execution.
To keep things relatively simple, we assume that the robot has a set of successful examples of the motion, in various initial conditions. In our case the set of movements was acquired by kinesthetic demonstrations, but it could also be acquired through tele-operation, exploration or human visual demonstrations. The latter case is the most complex as the robot would then need to match the human movements to its own body, i.e. solve the difficult correspondence problem [Nehaniv and Dautenhahn, 2002]. Furthermore, we do not address the problem of sequencing an planning, and consider only simple constrained reaching tasks. But the system is intended to be general enough so that a variety of

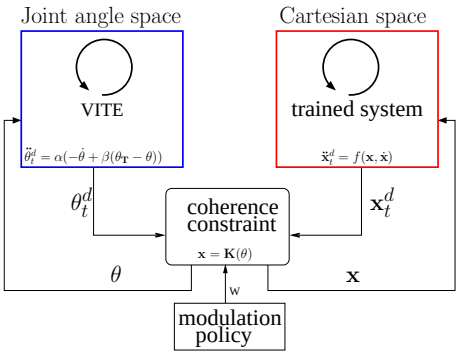


Figure 5.1: The structure of the controller for simple motor primitives (acceleration model). In the Cartesian FoR, the controller is learned from the data. In the joint angle FoR, the same VITE-like controller is used as for the reaching controller described in Chapter 4.

gestures can be learned. An example is illustrated in Fig. 5.3 for the two tasks for which the system was tested and described in section 5.5.

5.4 A controller for simple motor primitives

5.4.1 Overview

Controller structure

The basic idea underlying this controller is to reuse the reaching controller described in the previous chapter (see Fig. 4.2) and replace the VITE controller in Cartesian space by a learned controller, as depicted in Fig 5.1. While the attractor dynamical system in joint angle space guarantees that the target will be reached, the controller in Cartesian space can encapsulate the constraints associated with the task. In order to ensure the convergence of the global system, the weight modulation must be such that the Cartesian weights reach zero at the end of the movement. So the controller in Cartesian space can be seen as a *task model*, and is specific for each task. In contrast, the attractor joint angle controller is the same for all tasks. Two kinds of task models were investigated, which are described in Section 5.4.2.

System architecture

The system entails a learning stage and an execution stage, as shown in Fig. 5.2 which schematizes its global structure. During the training stage, features of the demonstrations (end-effector velocity profiles for the velocity model, or end-effector positions, velocities and accelerations for the acceleration model) are extracted and used to train a Gaussian Mixture Model (GMM) of the task (see Appendix A.4). During the reproduction stage, the trajectory is specified by a spring-and-damper dynamical system modulated by the GMM (see section 5.4.3). The target is tracked by a stereo-vision system and is set to be the attractor point of the dynamical system. The resulting velocity is then given to

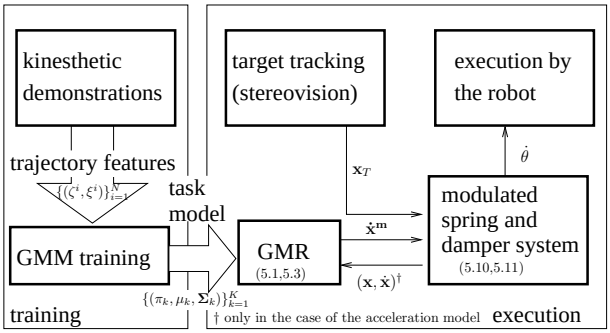


Figure 5.2: The architecture of the system. During training a set of features (ζ, ξ) is extracted from the demonstrations and used to train a GMM. During task execution, this model is used to modulate a spring-and-damper system. $\dot{\mathbf{x}}^{\mathbf{m}}$ represents the end-effector velocity specified by the task model. $\mathbf{x}_T$ is the target location, and $\mathbf{x}, \dot{\mathbf{x}}, \dot{\theta}$ are respectively the actual current end-effector position and velocity and the joint angles velocity. The numbers in parentheses refer to the corresponding equations in the text.

the robot which executes it. This does not hinder the online adaptation of the movement.

5.4.2 Task Model

Two different ways to model the demonstrated trajectories are investigated, the velocity model and the acceleration model. Both of them use Gaussian Mixture Regression (see Appendix A.4) for generalization, but on a different set of variables.

Velocity Model

The first way to encode a motion in a GMM, is to consider the velocity profile of the end-effector as a function of time $\dot{\mathbf{x}}_t$. Thus, the input variable ζ is the time and the output variable ξ is the velocity, like in the following velocity model:

$$\dot{\mathbf{x}}^{\mathbf{m}} = \tilde{\mathcal{F}}_{\dot{\mathbf{x}}}(t) \tag{5.1}$$

In words, the movement is modeled as a velocity profile, given by a function of time, which is learned as described in Appendix A.4. Here and henceforth, $\dot{\mathbf{x}}^{\mathbf{m}} \in \mathbf{R}^m$ is the end-effector velocity specified by the task model. $\tilde{\mathcal{F}}_{\dot{\mathbf{x}}}$ is obtained by applying (A.34) with the appropriate variables.

Acceleration Model

A second way of encoding a trajectory is to take as input the position $\mathbf{x}$ and velocity $\dot{\mathbf{x}}$, and as output the acceleration $\ddot{\mathbf{x}}$. The rationale of this is to consider a trajectory not as a function of time, but as the realization of a second-order dynamical system of the form:

$$\ddot{\mathbf{x}}^{\mathbf{m}} = \tilde{\mathcal{F}}_{\ddot{\mathbf{x}}}(\mathbf{x}, \dot{\mathbf{x}}). \tag{5.2}$$

Again, $\tilde{\mathcal{F}}_{\dot{\mathbf{x}}}$ is obtained by applying (A.34) with the appropriate variables. The velocity specified by the acceleration model is then given by

$$\dot{\mathbf{x}}^{\mathbf{m}} = \dot{\mathbf{x}} + \tau \tilde{\mathcal{F}}_{\ddot{\mathbf{x}}}(\mathbf{x}, \dot{\mathbf{x}}), \tag{5.3}$$

where τ is the time integration constant (set to 1 in this thesis). Since the position $\mathbf{x}$ and velocity $\dot{\mathbf{x}}$ depend on the acceleration $\ddot{\mathbf{x}}$ at previous times, this representation introduces a feedback loop, which is not present in the representation given by (5.1).

5.4.3 Modulated Spring-and-Damper System

We now show how the task model described above is used to modulate a VITE-like (or spring-and-damper) dynamical system active in joint space to reproduce the task with sufficient flexibility. Just like in the preceding chapter, we consider consider a VITE-like controller active in joint space:

$$\ddot{\theta}^d = \alpha(-\dot{\theta} + \beta(\theta_{\mathbf{T}} - \theta)) \tag{5.4}$$

where $\theta \in \mathbf{R}^n$ is the actual vector of joint angles (or arm configuration vector), and $\ddot{\theta}^d$ is the joint angle acceleration specified by spring-and-damper system. It can be converted as a desired velocity $\dot{\theta}^d = \theta + \tau\ddot{\theta}^d$, where τ is again the integration constant (set to 1 here). As mentioned in the preceding chapter, this dynamical system produces straight reaching paths (in joint space) to the target $\theta_{\mathbf{T}}$, which acts as an attractor of the system. This guarantees that the robot reaches the target smoothly, despite possible perturbations.
The above dynamical system is modulated by the variable $\dot{\mathbf{x}}^{\mathbf{m}}$ given by the task model (5.1) or (5.3). In order to weigh the modulation, we introduce a modulation factor $\gamma \in \mathbf{R}_{[0\ 1]}$, which weighs the importance of the task model relatively to the spring-and-damper system. If $\gamma = 0$, only the spring-and-damper system is considered, and when $\gamma = 1$ only the task model is considered. In order to guarantee the convergence of the system to $\theta_{\mathbf{T}}$, γ has to tend to zero at the end of the movement. In the experiments described here, γ is given by:

$$\ddot{\gamma} = \alpha_\gamma(-\dot{\gamma} - \frac{1}{4}\alpha_\gamma\gamma) \quad \text{with } \gamma_0 = 1, \tag{5.5}$$

where γ_0 is the initial value of γ and $\alpha_\gamma \in \mathbf{R}_{[0\ 1]}$ is a scalar.

Similarly to what was done in the previous chapter, we combine impose coherence constraints between the two controller by solving the same constrained minimization problem:

$$\begin{aligned} \dot{\theta} = \quad & \underset{\dot{\theta}}{\text{argmin}} \quad (1-\gamma)(\dot{\theta} - \dot{\theta}_t^d)^T \bar{\mathbf{W}}_\theta (\dot{\theta} - \dot{\theta}_t^d) + \\ & \qquad \gamma(\dot{\mathbf{x}} - \dot{\mathbf{x}}^{\mathbf{m}})^T \bar{\mathbf{W}}_{\mathbf{x}} (\dot{\mathbf{x}} - \dot{\mathbf{x}}^{\mathbf{m}}) && (5.6) \\ & \text{u.c.} \qquad \dot{\mathbf{x}} = \mathbf{J}\dot{\theta}, && (5.7) \end{aligned}$$

where $\mathbf{J}$ is the Jacobian of the robot arm kinematic function $\mathbf{K}$ and $\bar{\mathbf{W}}_\theta \in \mathbf{R}^{n\times n}$ and $\bar{\mathbf{W}}_{\mathbf{x}} \in \mathbf{R}^{m\times m}$ are diagonal matrices necessary to compensate for the different scale of the $\mathbf{x}$ and θ variables. As a rough approximation, the diagonal elements of $\bar{\mathbf{W}}_{\mathbf{x}}$ are set to one and those of $\bar{\mathbf{W}}_\theta$ are set to the average distance

between the robot base and its end-effector.
The solution to this minimization problem was computed in Section 4.4.2:

$$\dot{\theta} = \dot{\theta}^d + \mathbf{W}_\theta^{-1}\mathbf{J}^T\left(\mathbf{W}_\mathbf{x}^{-1} + \mathbf{J}\mathbf{W}_\theta^{-1}\mathbf{J}^T\right)^{-1}(\dot{\mathbf{x}}^\mathbf{m} - \mathbf{J_t}\dot{\theta}^d) \quad (5.8)$$

$$\text{where} \quad \mathbf{W}_\theta = (1-\gamma)\bar{\mathbf{W}}_\theta, \quad \mathbf{W}_\mathbf{x} = \gamma\bar{\mathbf{W}}_\mathbf{x}. \quad (5.9)$$

To summarize, the task is performed by integrating the following dynamical system:

$$\ddot{\theta}^d = \alpha(-\dot{\theta} + \beta(\theta_\mathbf{T} - \theta)) \quad (5.10)$$

$$\dot{\theta} = \dot{\theta}^d + \mathbf{W}_\theta^{-1}\mathbf{J}^T\left(\mathbf{W}_\mathbf{x}^{-1} + \mathbf{J}\mathbf{W}_\theta^{-1}\mathbf{J}^T\right)^{-1}(\dot{\mathbf{x}}^\mathbf{m} - \mathbf{J}\dot{\theta}^d) \quad (5.11)$$

where $\mathbf{W}_\mathbf{x}$ and $\mathbf{W}_\theta$ are given by (5.5) and (5.9), and $\dot{\mathbf{x}}^\mathbf{m}$ is given either by (5.1) (velocity model) or by (5.3) (acceleration model). Integration is performed using a first-order Newton approximation ($\dot{\theta}_t^d = \dot{\theta} + \tau\ddot{\theta}_t^d$).

Since the target location is given in Cartesian coordinates, inverse kinematics must be performed in order to obtain the corresponding target joint angle configuration which will serve as input of the spring-and-damper dynamical system. In the case of a redundant manipulator (such as the robot arm used in the following experiments) the desired redundant parameters of the target joint angle configuration can be extracted from the demonstrations. This is done by using the GMR technique described in Appendix A.4 to build a model of the final arm configuration as a function of the target location.

5.5 Experiments

5.5.1 Setup

We validate and compare the systems described in this paper on two experiments. The first experiment involves a robot putting an object into a box and the second experiment consists in reaching and grasping for an object. Those experiments were chosen because (1) they can be considered as simple goal-directed tasks (for which the system is intended), (2) they are tasks commonly performed in human environments and (3) they presents a clear success or failure criterion.
All the experiments presented below are performed with a Hoap3 humanoid robot. This robot has four back-drivable degrees of DoFs at each arm. Thus, the robot arms are redundant, as we do not consider end-effector orientation. The robot is endowed with a stereo-vision system enabling it to track color blobs. A small color patch is fixed on the box and on the object to be grasped, enabling their 3D localization. Pictures of the setup are shown in Fig. 5.3.

5.5.2 Data acquisition and pre-processing

During the kinesthetic demonstrations, the robot joint angles were recorded at a rate of 1000 Herz. End-effector positions were computed using the arm kinematic function. All recorded trajectories were linearly normalized in time ($T = 500$ time steps) and filtered to remove high-frequency noise. Velocities

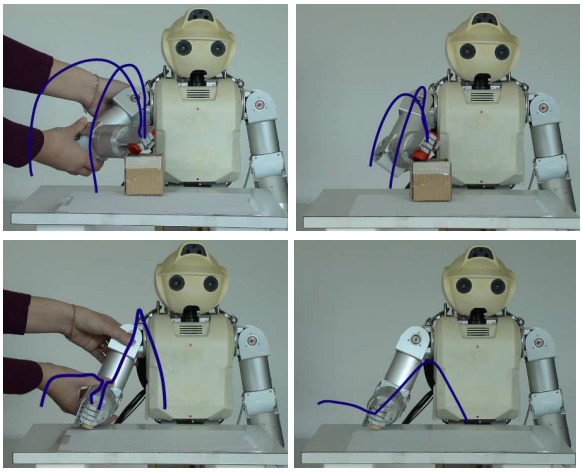


Figure 5.3: The setup of the experiments. The top pictures show the first task and the lower picture sow the second task Left: a human operator demonstrates a task to the robot by guiding its limbs. Right: the robot performs the task, starting from different initial positions.

and acceleration were obtained by simply subtracting consecutive positions, and were subsequently also filtered. The number of Gaussian components for the task models were found using the Bayesian Information Criteria (BIC) [Schwarz, 1978], which finds a trade-off between the explanatory power of the model and its complexity (its number of free parameters):

$$BIC = -2\log(L) + k\log(n). \tag{5.12}$$

In this equation, L is the likelihood of the data given the trained model, k is the number of free parameters of the model and n is the amount of data. After training models with different number of components, the one with the lowest BIC score is chosen. The Gaussian components were initialized using the k-means algorithm [MacKay, 2003], which is itself randomly initialized.the parameter values used were $\alpha_\gamma = 0.06$, $\alpha = 0.12$ and $\beta = 0.06$. Those values were found "by hand".

5.5.3 Putting an object into a box

Description

The experimental setup can be seen on Fig. 5.3, top. To perform this task, the robot has an object in his hand and must reach above the box and release the object. The execution is considered successful if (and only if) the object falls into the box. The object is a 6-centimeter long cylinder, and the box is a 6-centimeter wide cube made out of light cardboard and lies on the table. In order to accomplish the task, the robot has to avoid hitting the box while performing the movement. Otherwise, the box falls, or is displaced outside the work space of the robot thus making it impossible for the robot to correctly place the object. If the hand holding object initially lies below the top of the box,

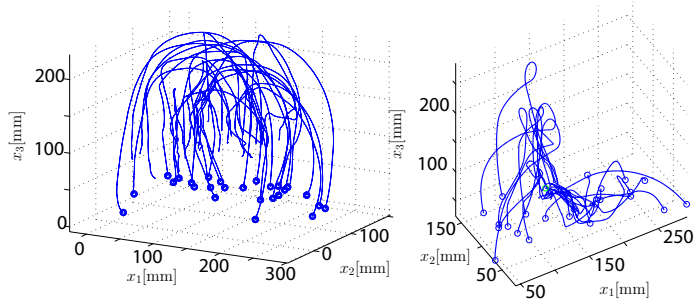


Figure 5.4: The demonstrated trajectories for the box task (left) and the grasping task (right). Circles indicate starting positions.

a straight line reaching will in general yield a failed attempt. This is why the modulation is expected to be helpful. The task model is expected to capture the information that the robot has to first reach up above the box, and then down to the box. This task is hypothesized to be suitable for learning with the present system. The hypothesis was that the horizontal movement components would be averaged out by the GMM and that the system would retain the vertical components. In other words, the system would learn that it has to reach up above the box and then down. Otherwise, the robot would hit the box while reaching the target, and would thus fail in the task.

Training

A set of 26 kinesthetic demonstrations were performed, with different initial positions and box locations. The box was placed on a little table. Thus its location only varies in the horizontal plane. Similarly, the initial position of the object (and thus of the end-effector) laid on the table. The set of demonstrated trajectories is depicted in Fig. 5.4, left. The velocity models trained on this data are shown in Fig. 5.5, left.

5.5.4 Reach and Grasp

Description

In order to accomplish this task, the robot has to reach and correctly place its hand to grasp a chess piece. In other words, it has to place its hand so that the chess piece stands between its thumb and its remaining fingers, as shown in Fig. 5.9, left. This figure illustrates that the approaching the object can only be done in one of two directions: downwards or forwards. This task is more difficult than the previous one, as the movement is more constrained. Moreover, a higher precision is required on the final position, since the hand is relatively small.

Training

A set of 24 demonstrations were performed starting from different initial positions located on the horizontal plane of the table. The chess piece remained

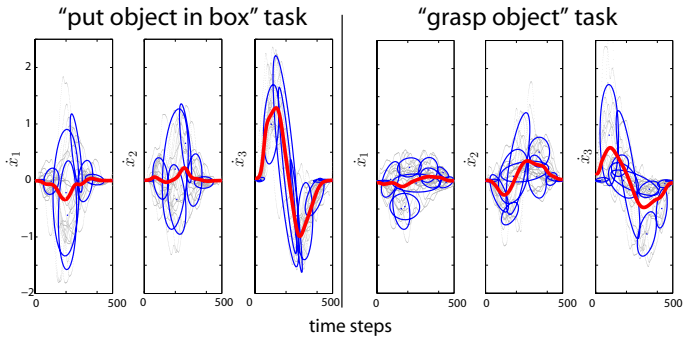


Figure 5.5: The velocity models for both tasks. The dots represent the training data, the ellipses the Gaussian components and the thick lines the trajectory obtained by GMR alone. The thick lines show that, for the first task, the horizontal components $\dot{x}_1$ and $\dot{x}_2$ are averaged out by the model, but the vertical component $\dot{x}_3$ shows a marked upward movement. For the second task, all components are almost averaged out.

in a fixed location. Depending on the initial position, the chess piece was approached either downward or forward (as illustrated on Fig. 5.9). The set of demonstrations is represented in Fig. 5.4, right. The resulting velocity model is shown in Fig. 5.5, right. One can notice that there is no velocity feature that is common to all demonstrated trajectories. The acceleration model is shown in Fig. 5.6. This model captures well the fact that the vertical acceleration component depends on the position in the horizontal plane.

5.5.5 Results

Endowed with the system described above, the robot is able to successfully perform both tasks. For the first task, both the velocity and the acceleration models can produce adequate trajectories (see Fig. 5.7, left for examples). Fig. 5.8 shows the effects of the velocity model and the VITE-like controller for the box task. The veloctity model makes the trajectory move upwards at the movement onset, while the VITE-like controller make the movement reach the target. The system can adapt its trajectory online if the box is moved during movement execution (see Fig. 5.7, right). For the second task, examples of resulting trajectories are displayed in Fig. 5.9, right.
In order to evaluate the generalization abilities of the systems, both tasks were executed from various different initial positions arbitrarily chosen on the horizontal plane of the table, and covering the space reachable by the robot. Fig. 5.10 shows the results and starting positions for both experiments. For the box experiment (left), the velocity model was successful for 22 out of the 24 starting locations (91%). The two unsuccessful trials, indicated by empty circles, correspond to initial positions close to the work space boundaries. The acceleration model was successful for all trials (100%).
For the chess piece experiment (Fig. 5.10, right), the velocity model was successful for 5 out of 21 (24%) trials, whereas the acceleration model was successful for 18 trials (86%). This performance gap is due to the fact that this task does not require a fixed velocity modulation. The adequate modulation depends on

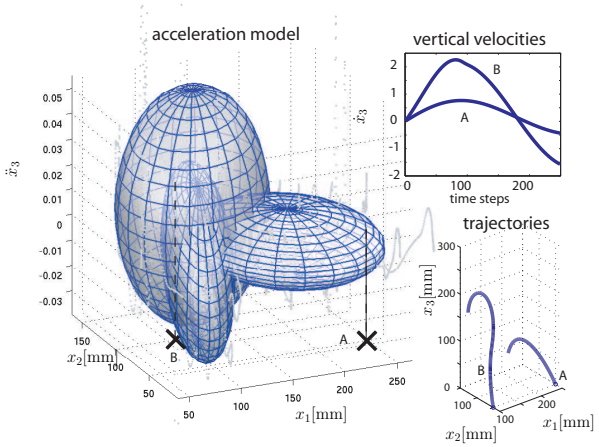


Figure 5.6: Left: the acceleration model for the second task. The ellipsoids show the Gaussian components at twice their standard deviation. Only three projections (out of nine) are shown. The vertical acceleration strongly depends on the position in the horizontal plane. On the lower right, two trajectories encoded by this model but starting from different positions A and B (indicated by the crosses) are shown. The corresponding vertical velocity profiles appear on the upper right. They differ significantly, as the model is not homogeneous across the horizontal plane.

the position. This position-dependent modulation can be captured by the acceleration model, but not by the velocity model. As illustrated in Fig. 5.6, the acceleration model is able to produce different velocity profiles, depending on the starting position and is thus more versatile than the velocity model.

5.6 Discussion

The results described in the preceding section show that the framework suggested here can enable a robot to learn constrained reaching tasks from kinesthetic demonstrations, and generalize them to different initial conditions. It also shows that the multi-referential dynamical system approach described in Chapter 4 can be applied to control more sophisticated movements. In the present chapter, it has been adapted and extended to enable the robot to learn and perform simple goal-directed tasks. The dynamical system approach provides enough flexibility to combine different components of the gesture, a goal-directed component and a task-specific component. Furthermore, it allows to deal with perturbations occurring during the task execution. This framework can be used with various task models and has been tested for two of them, the velocity model and the acceleration model. The results indicate that the velocity model is too simplistic if the task requires different velocity profiles when starting from different positions in the workspace. The acceleration model is somewhat more sophisticated and can model more constrained movements, but may fail to provide an adequate trajectory when brought away from the demonstrations in the phase space $(\mathbf{x}, \dot{\mathbf{x}})$. If the demonstrations did not provide data in a particular region of the phase space, the task model will be at loss to provide relevant in-

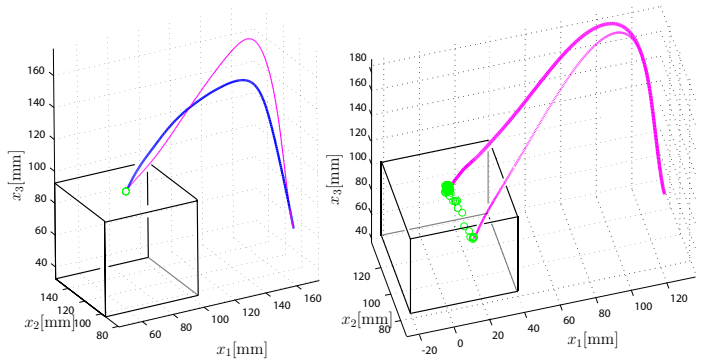


Figure 5.7: Left: end-effector trajectories of the robot putting the object into the box. The thin line corresponds to the velocity model and the thick line corresponds to the acceleration model. Right: online trajectory adaptation to a target displacement using the velocity model. The circles indicate to location of the box, as tracked by the stereo-vision system. The thick line shows the produced trajectory and the thin line shows the original trajectory if the box remained unmoved. Similar results were obtained with the acceleration model.

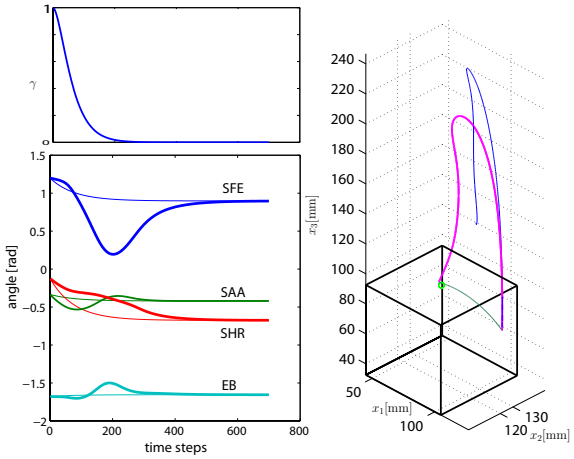


Figure 5.8: The effects of the velocity model and the VITE-like model on the trajectory. Bottom left: the joint angle trajectories specified by the angular controller (thin lines) and the joint trajectories after the modulation with the Cartesian controller (thick lines). Top left: The evolution of γ. Right: the trajectory generated by the VITE-like angular controller (flat thin line) would hit the box and fail, but when it is modulated by the velocity model (ascending thin line), it results in a trajectory that can perform the task (think line).

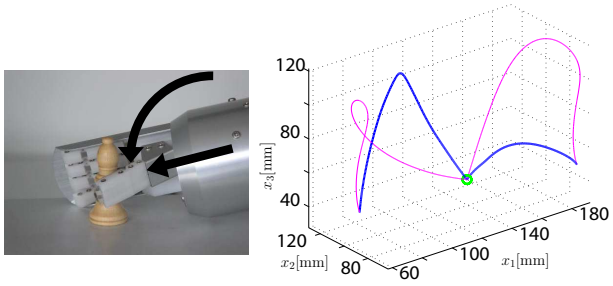


Figure 5.9: Left: the chess piece to be grasped. For a successful grasp, the robot has to approach it as indicated by the arrows. Right: resulting trajectories for the grasping task, starting from two different initial positions. The acceleration model (thick lines) adapts the modulation to the initial position, while the velocity model (thin lines) starts upward in both cases. The trajectory produced by the velocity model and starting left of the target is not successful.

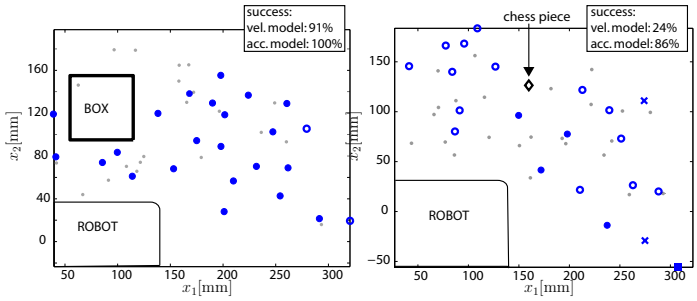


Figure 5.10: The robustness to initial end-effector position for both tasks. The plots represent top views of the first (right) and second (left) experiment. The filled markers (circles or squares) indicate all initial positions for which the velocity model was successful. The circles (filled and non-filled) indicate all initial positions for which the acceleration model was successful. The crosses indicate initial end-effector positions for which both models failed. The dots indicate the starting positions of the training set.

struction. However, the advantage of using a probabilistic representation is that such a case can be easily detected by looking the probability density function of this region of the phase space. Other regression techniques such as Locally Weighted Regression, could also be used for the task model.
In its present form, the modulation factor between the dynamical system and the task model (γ) is not learned. Its dynamics has been designed "by hand", so as to reach zero at the end of the motion. Learning those dynamics from the demonstrations is likely to further improve the performance of the system. Indeed, some tasks may require that the task model keeps its bigger influence on the movement for a longer period of time, especially tasks requiring a particular modulation at the end of the movement. It would also be desirable to have a system that extracts the relevant variables, and automatically selects the adequate model. A first step in this direction has been taken in [Calinon et al., 2007], where a balance between different sets of variables is achieved.

Of course, the adequacy of this framework is restricted to relatively simple tasks, such as those described in the experiments. More complicated tasks, such as obstacle avoidance in complex environments or stable grasping of particular objects require a detailed model of the environment and more elaborate planning techniques. The tasks considered for this framework are those that cannot be accomplished by simple point-to-point reaching, but still simple enough to avoid the complete knowledge of the environment. But this framework could be extended to learn more complicated tasks. In a first step in this direction,[Guenter et al., 2007] investigates how Reinforcement Learning can be added to this framework to deal with obstacle avoidance.

5.7 Summary

In this chapter, we moved beyond basic point-to-point reaching motions to constrained reaching motions. Indeed, we showed that the framework developed in the preceding chapters can be extended to perform goal-directed movements with constraints on the arm kinematics. Those constrained reaching motions are learned from kinesthetic demonstrations performed by a human. A Gaussian Mixture Regression technique is used to learn a probabilistic model of the movement, and is then used to modulate the dynamical system based reaching controller described in the previous chapter.
The effectivity of this approach is illustrated with experiments involving the humanoid robot grasping an object and putting it into a box. Using the system developed in this thesis the robot is able to accomplish the task in a robust way with various initial configurations. Those experiments show that all the rather theoretical developments made in those last three chapters of the thesis can be successfully implemented and used for a real, non-trivial, robotic application.

Chapter 6

Discussion

Voilà que j'ai touché l'automne des idées,
Et qu'il faut employer la pelle et les râteaux

Charles Baudelaire

6.1 Main contributions of this thesis

In the previous pages, we suggested a framework allowing a humanoid robot to learn how to control its limbs, perform reaching movements and learn simple constrained goal-directed motions from demonstrations. Although this works uses standard and well-known mathematical tools, the novelties its entails are manifold.
In Chapter 3, a new method enabling a humanoid robot to learn its body schema is presented. Concedingly, enabling a robot to autonomously learn its body schema has been done before. But (to the best of my knowledge) previous attempts could not reach the same level of scalability and did not operate in such an on-line manner, where robot can learn its body schema "on the side", while performing other tasks. This performance was achieved by finding a "middle way" between bio-inspired approach to kinematic learning and self-calibration methods, combining the on-line learning of the former and the explicit parametrization of the latter.
Moreover, while previous solutions relied on the robot observing its end-effector, either the field of view was wide enough so that no movement of the camera was required, or the tracking was hard-coded, so that the robot knew *a priori* the kinematic function of the visual apparatus. Here, the robot learns its body schema without knowing *a priori* how to track its body parts. This is first learned by considering the visual flow when moving its head. Once the kinematic function of he head and eyes is approximated well enough to enable a visual tracking behavior, the robot can learn how to control its other limbs.
Finally, by explicitly modeling the joint rotations and limbs translations, the suggested model of the body schema allows a control of the end-effector position but also of its orientation and of other body parts, such as the elbow, which could not be done by classical bio-inspired approaches.

The control problem is addressed in Chapter 4, which presents a new con-

troller for reaching movements. This controller consists in two attractor dynamical systems sub-controllers active concurrently in joint space and in task space respectively. This results in a controller which is robust to perturbations, provides a simple solution to the joint limit avoidance problem and has no singularity, even when dealing with highly redundant manipulators. Those properties are obtained by taking advantage of the characteristics of each sub-controller. This controller converges to the target and can be used for point to point reaching as well as for reaching with a given target orientation. It has been implemented in two different humanoid robots, the Hoap3 that has 4 DoFs arms and the iCub that has 7 DoFs arms.
In addition to its usage for robot control, the controller provides a computational formulation of the hybrid joint space - task space control hypothesis expressed in [Cruse and Brüwer, 1987, Carrozzo and Lacquaniti, 1994].
While this surely is not the first controller suggested to control the arms of a robot, it is the first time (to my knowledge) that this controller is operating concurrently in joint space and in task space. It results in a generalization of the classical DLS control technique [Chiaverini et al., 1994]. And it constitutes a further exploration step in the still largely unknown field of multi-modal and decentralized control.

This controller is extended in Chapter 5 to enable the robot to learn new tasks from demonstrations. Using GMMs, the dynamical system in task space is learned from the demonstration, while keeping the attractor dynamical system in joint space. The robot is then able to perform the demonstrated tasks in a way that is robust to the initial conditions and to perturbations occurring during movement execution. Experiments using the Hoap3 robot show that this framework is effective for learning tasks such as putting an object into a box and reach to grasp an object. While the system may not perform as well as one specifically programmed for a given task, it has a versatility which cannot be observed in traditional robotic systems. This versatility is typical of mammals and birds, who can learn new tasks from observation and practice. The system suggested in this thesis is a first step towards learning motor primitives for a task-oriented control that can exhibit adaptivity to a changing environment.

6.2 Main limitations of this thesis

Despite many efforts, a few aspects of this thesis remain uncleared. First, while convergence of the body schema learning has been demonstrated for a single joint (see Section 3.3.4), the same could not be shown for more complex body structures. Simulation studies showed that in a few cases, the algorithm does not converge in a very large number of steps. However, I could not derive analytically which body structures can be expected to converge and which cannot. This would have been a very interesting result, and could then form the basis of a taxonomy of body schemata. One could have then studied what kinds of body schemata can be found in vertebrates.

In Chapter 4, the joint limit avoidance mechanism suggested works well and was relied upon when using the robot. However, a rigorous analysis of its behavior, and a way to determine an adequate γ depending on the dynamical system

parameters, the maximal allowable speed and the allowable joint angle range is still missing.

Another point that would need clarification is what kind of tasks exactly can be learned using the system described in Chapter 5. While it works well for the tasks presented in the experiments, it cannot, in its present form, be considered as an all-purpose method. But it is very difficult to know beforehand what tasks can be learned, and those that cannot. Moreover, we did not provide a way to adapt the dynamics of the modulation factor γ to the task at hand. We found a modulation that worked well for the two tasks that we studied, but other tasks would probably require other modulation dynamics. Neither did we specify the behavior of the system when the acceleration model encounters unexplored regions of the phase space. To make the learning system really robust, those two elements should be addressed.

6.3 Directions for future work

If the body schema is to be understood as a representation of peripersonal space, and as suggested already by [Wallon, 1954], maybe objects lying within reach should be included into the body schema. Indeed the distinction between objects and tools appears quite arbitrary. Objects could also have a frame of reference attached to them and interactions with them could be achieved by mapping the frame of reference of the hand to the frame of reference of the object. The inclusion of objects, should probably not be akin to an "objective" representation of the surrounding environment, but rather they should appear and disappear from the body schema according to the "intention" of the robot to interact with them. This would be a possible implementation of the "world as outside memory" concept [O'Regan, 1992], basically the idea that humans do not have a complete representation of their environment, but turn to the environments itself when in need of information to interact with it. This could also fit into the view of the body schema as a dynamic "melody" [Sheets-Johnstone, 2005], where things appear and disappear according to the situation at hand. It would be interesting to see what advantages, if any, could be gained by including objects in the body schema, for example for manipulation and planning.

The reformulation of the control problem as a FoR transformation problem, and the possibility to extract kinematic chains out of the kinematic tree (described in Section 3.3) opens up a number of possibilities for control, for example in bi-manual manipulation or center of mass control. For many bi-manual manipulation tasks, what matters is the relationship (i.e. the FoR transformation) between the two hands and not the position of one particular hand. Hence, considering a kinematic chain going from one hand to the other may facilitate the control by not imposing unnecessary constraints on the actual position of the hand.
In the same vein, one could try to control not one single kinematic chain, but many, partially overlapping kinematic chains. This could be useful for example to control the center of mass of the robot (assuming the knowledge of the limbs weight distribution), or movements involving more than two limbs like putting one hand on each eye. A further application would be to control hand and finger

movements for a grasping or other manipulation task. Further investigations would be necessary to assess the applicability of this approach.

According to [Turvey, 2007], motor primitives (or synergies) include also some perceptual aspects. In the model of motor primitives suggested in Chapter 5, perception is restricted to the localization of the target, which is fed as input to the dynamical system. For the simple tasks experimented in this chapter, this is enough, but more complicated tasks may also require a model of the perceptions accompanying the execution of the task, for example seeing the hand moving or feeling the contact and resistance of the object. Further research could be done to include more complex and time-dependent perceptual input into the motor primitive.

6.4 Relationships to other theories

The relevance of a model lies not only in the accuracy of its predictions but also in its ability to broach relevant questions and in its relationships to previous theories. In this section, I rediscuss this work in light of several theories on motor control and cognition, trying to sort out their compatibility and contradictions.

6.4.1 Forward and inverse models

The literature on the human control of reaching movements has devoted a lot of attention to the issue of internal forward and inverse models. Although those concepts have been used to describe different constructs [Karniel, 2002], the following definitions are widely accepted in the context of reaching movements. A forward model is basically a model the arm in its environment which can predict the state of the arm, given a sequence of commands. An inverse model can retrieve the commands to the arm needed to bring it to a particular state. It has been argued that humans have internal forward models of the kinematics as well as the dynamics of their arm [Ariff et al., 2002, Shadmehr and Mussa-Ivaldi, 1994]. Some scholars have also argued for the existence of inverse models in the brain [Kawato et al., 1990]. This view has been challenged by theoretical considerations showing the inherent difficulties in learning an inverse model [Jordan and Rumelhart, 1992]. Indeed inverse models are ill-defined as many different commands can yield the same sensory input. Moreover the set of commands causing a given sensory signal is usually not convex, implying that learning methods based on averaging (virtually all methods) are bound to failure. Furthermore, as argued in [Latash, 2008, pp.63-65] inverse models seem hardly compatible with the biological substrate of the arm. Executing a simple reaching movement, would indeed require a cascade of inverse models solving redundancy at the levels of target arm configuration, arm trajectories, arm dynamics and nervous signal generation.

The model of the body schema presented in Chapter 3 can be seen as a forward model, as it can predict the sensory consequence (here the visual position of the end-effector, or intermediate joint) of a motor command (given by an arm configuration). The global inverse kinematics algorithms used, be it CCD (described in Section 2.3.3) or the geometrical inverse can be seen as inverse

models. However,those are not learned directly, they are derived from the forward models. This way the problems associated with learning inverse model are avoided. And using a dynamical system approach the redundancy of the inverse model is taken advantage of for keeping movements to a minimum as described in Section 4.4.3.

6.4.2 Gestalt and motor theories of perception

The Gestalt theory of perception [Guillaume, 1937] developed in the early 20^{th} century states that perception is about finding structure or form in the stimuli. Experiments involving, among others, optical illusions showed that what is perceived is not a perfect match of the stimuli, but rather its general structure or shape (gestalt). Maybe the most rigorous and mathematical expression of the Gestalt theory of perception has been given by [Leyton, 2001]. In this "Generative Theory of Shape", Leyton suggests an algebraic model of human perception inspired by Gestalt psychology. According to this theory, perceiving something amounts to decomposing the stimulus into a hierarchy of algebraic groups. For example, the perception of a cylinder is a process by which the stimulus is recognized as generated by a two-level hierarchy of rotations and translations, as depicted in Fig 6.1. Mathematically, the cylinder is described as

$$SO(2) \wr \mathbb{R}, \tag{6.1}$$

where $SO(2)$ is the 2D rotation group,$\mathbb{R}$ is the 1D translation group and $\wr$ is the *wreath product* of two groups. Basically, the wreath product of the group $SO(2)$ by the group $\mathbb{R}$ is the replication of multiple copies of $SO(2)$, indexed by the elements of $\mathbb{R}$. If $S0(2)$ represents a circle, $SO(2) \wr \mathbb{R}$ can be seen as the stacking up of multiple circles, and thus represents a cylinder as shown in Fig. 6.1. In this framework, perceiving a stimulus as a cylinder can be understood as figuring out that the points of the stimulus can be generated by translating a circle a along an axis. In other words, perceiving means uncovering the hierarchical group structure leading to the stimulus. According to this theory, localizing or identifying a point on the cylinder amount to identifying the elements of the groups belonging to the hierarchy, that is finding the coordinates the the frame of reference defined by the group hierarchy. For example in Fig 6.1, points on the cylinder are identified by the rotation angle and the height from the base. This was experimentally validated by showing subjects stimuli typical of Gestalt experiments such as optical illusions, asking them to report what they see and how they perceive the stimulus.
Adopting this framework, the peripersonal space can be described as

$$SO(2)^n \wr SO(2)^{n-1} \cdots \wr SO(2)^1, \tag{6.2}$$

where $SO(2)^i$ is the 2D rotation group around the axis given by $\mathbf{a}_i$ (see Section 3.3). Localizing a point in this space amounts to finding the elements of the groups in (6.2) that correspond to this point. In other words, it is the same as finding the global inverse kinematics of $\mathbf{x}$, as described in Section 2.3.3. Thus the inverse kinematics function can be seen as a perceptual process. This would imply that perceiving a point in space amounts to computing the command to reach that point with a particular body part. This is equivalent to the motor theory of perception [Fadiga et al., 2000], which says that perceiving someone

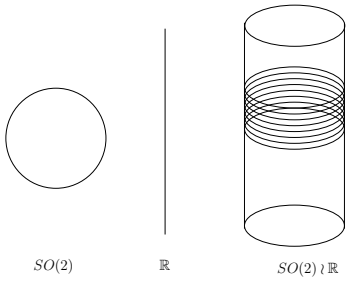


Figure 6.1: The cylinder as a wreath product of $SO(2) \wr \mathbb{R}$. The circle (represented by the algebraic group $SO(2)$) is swept across the line (represented by the group $\mathbb{R}$), resulting in a cylinder (represented by the group $SO(2) \wr \mathbb{R}$).

performing an action involves "internally simulating" the execution of that action. This gets along with a motor perception of space [Fadiga, 2006], which was already suggested more than a century ago, for example by [Kirkpatrick, 1899] or Poincaré, who wrote:

> Quand on dit d'autre part que nous "localisons" tel objet en tel point de l'espace, qu'est-ce que cela veut dire? *Cela signifie simplement que nous nous représentons les mouvements qu'il faut faire pour atteindre cet objet.* [Poincaré, 1902, p.82][1]

We thus see that with the help of the model presented in this thesis, it is possible to articulate two old, and apparently unrelated theories of perception, such as the motor and the Gestalt theories of human perception.

6.4.3 Affine geometry as a control space

One of the most puzzling invariants found in the study of human movement is the so-called *two-third power law*. This invariant was found by [Lacquaniti et al., 1983], who noticed that when subjects are asked to draw 2D ellipses or to scribble, the tangential hand velocity was always proportional to the radius of curvature to the power 2/3. This relationship also holds for movements perceived as having a constant speed by observing subjects. While this has been confirmed by many studies [], it has also be shown that this property is not present for movements in three dimensions [Schaal and Sternad, 2001]. As an explanation for this, it has been suggested that the geometry underlying movement production and perception is the affine geometry rather than the standard Euclidean geometry [Pollick and Sapiro, 1997, Flash and Handzel, 2007]. It has indeed been mathematically shown that movements at constant "affine" velocity in the plane obey the two third power law. Affine geometry differs from Euclidean geometry in that it is invariant to linear distortion. In other words, no notion of angle can be defined and distances can only be compared in parallel directions. Affine geometry is not as general as projective geometry but more general than Euclidean geometry. Affine space can be thought of a vector space where there is

[1] "On the other hand, when we say that we "localize" an object at this or that point in space, what does it mean? It simply means that we imagine the movements that we need to perform in order to reach that object." (my translation)

no origin. In other words, there is no addition (as this involves the origin), only subtractions.

The hypothesis that movement perception in the peripersonal space is grounded in an affine space can be compatible with the model of peripersonal space representation presented in this thesis. As suggested earlier (Section 3.5.3), the model could be extended to include the projection from monocular to stereovision. Thus the visual modality would not be in a Euclidean 3D space, but rather a pair of two 2D images. It could well be (although this should be checked further) that those two images are related by an affine transformation, as lines seen as parallel by one eye will also be seen as parallel by the other eyes. Thus, affine space would be a natural space for the visual modality as affine geometry could unite the views from both eyes. One could then try to reformulate the reaching controller presented in Chapter 4 so that the visual space becomes affine. Indeed, the VITE-like dynamical system (4.3) can be expressed using only subtractions and can thus potentially take place in an affine space. Those are of course only conjectures, but they could be interesting lines of research to explore.

6.4.4 Autopoietic perspective on cognition

This theory, described in [Varela, 1989], takes a biological view on cognition and studies it in the light of other complex biological entities. This led Varela to suggest the concept of autopoiesis as a property of biological systems. Autopoietic systems are systems that autonomously maintain their identity in their environment. An explicit example of such a system is the biological cell. It autonomously maintains its identity, by maintaining the cellular membrane. This identity is not a material identity (as the molecules of the membrane change all the time), but rather an identity in terms of structure and organization, that comes along a continuity in time. Unlike inanimate machines like computers, such systems do not need to be specified in terms of input and output, they just maintain their structure and organization in a dynamic environment. Sure enough, for an external observer, a perturbation in the system environment can be seen as an input (like light hitting a retinal cell) and if the resulting adjustments (like the release of ions outside the cell) perturbs its environment, it can be seen as an output. But the input-output relationship is not the entire story, it is more a side-effect of the self-maintaining dynamics. For Varela, animals and plants are autopoietic systems made of a hierarchy of autopoietic systems. This hierarchy includes the nervous system (and thus cognition), which should thus be studied as such rather than as an input-output machine.

The model of the body schema presented in this thesis can be related to autopoietic systems. Indeed, its main property is to keep some kind of invariance between the visual and the proprioceptive modalities, and the invariance is represented by the kinematic function. Like autopoietic systems, it can also be seen from the outside as an input-output system, but from the "inside" it is more about keeping this invariance. Similar to autopoietic systems, past "experiences" have an influence on its present behavior, not because it has "memorized" them, but rather because the present behavior is a result of the system ontogeny. And more fundamentally, our model of the body schema

illustrates well Varela's thesis that perception is not a mirroring of an existing external reality, but rather the construction of such a reality, based on the "*invariance of the anatomical and functional organization of the nervous system during its interactions*" [Varela, 1989, p.166]. Our model of the body schema does not necessarily reflects the true shape of the robot, rather it constructs this reality, based on some invariant that are internal (the kinematic tree structure) as well as mediated by the environment.

6.5 Hard problems

The models described in this thesis evolve from a low-level sensorimotor contingencies to higher-level task learning. Those topics touch upon a few "hard problems" [Chalmers, 1995], mainly the ones related to intentionality, i.e. the property of being directed toward something, of having an object. Not being being so bold, I carefully avoided those hard problems in the course of this thesis. They are briefly evoked here, in order to counter a potential suspicion that those problems where ignored or underestimated.

The question of intentionality already arises in the body schema learning. As mentioned earlier, it has been shown that in humans tools are integrated in the body schema. However this integration takes place only when the subject is actively using the tool [Maravita et al., 2002, Witt et al., 2005]. So body schema adaptation is not only a "passive" phenomenon, it also has an intentional content, as already suggested by [Merleau-Ponty, 1945, p.117]. Furthermore, it has been suggested that the body schema constitutes the basis for pre-reflexive consciousness [Legrand et al., 2007]. According to this theory, the peripersonal space representation grounds a subjective and intentional experience of spatiality. Subjectivity and intentionality are thought to be the two main ingredients of consciousness [Searle, 2000].

It is questionable whether learning to accomplish a task without capturing its intended outcome makes sense. This is why in the algorithm described in this thesis the target state is hard-coded. Ideally, however, the robot should extract by itself, what is the aim of the task. This could be done by considering the statistical regularities in the target states of the demonstrations. It remains unclear however, if this can be enough, without having an "intentional stance" [Dennett, 1987] to the demonstrator, i.e. considering what the demonstrator is trying to achieve and what are his beliefs. And in some tasks, this would require extensive practical and cultural knowledge, which can be very difficult to acquire for a robot. As long as we are dealing with "toy problems" in well controlled environments, this should not be an issue. However it may become one in an unknown environment and naive users.

6.6 Perspectives for developmental robotics

Throughout this thesis, the tension between the naturalist and the roboticist posture described in Section 1.4 can be felt. And as Chapters 3 to 5 unfold from low-level sensorimotor coordination to higher level task learning, we can

observe a drift away from the naturalist perspective. The model of the body schema described in Chapter 3 takes into account experimental observations on human space representation and could reproduce them in a robot. The reaching controller presented in Chapter 4 is inspired by principles putatively active in human control. But the differences in the actuation system between human and robots (see Section 4.2) are such that trying to reproduce precise features of human reaching motions on a robot without accounting for those differences would be questionable. Thus, by using a robot with standard actuation, one is bound to yield to the roboticist and to frustrate the naturalist. In Chapter 5, the concepts of objects, task, and goal, along with the semantic aspect of action in humans are so fuzzy to the naturalist, that the model suggested can only remotely be connected to what happens in humans. This drift can be surprising, as a common view holds that since the hardware is different, the lower level which interacts with this hardware is different, but at a higher level, robot and human cognition can be similar. This view is inspired by the computer analogy, where the low-level binary code is dependent on the processor, on contrast to higher level programming languages which are common to all machines. So the drift observed in this thesis can either be attributed to my own incompetence or the still immaturity of the research on cognition, or it can hint at the inappropriateness of the computer analogy. It could indeed well be, that higher and lower level cognitive and motor functions are so intermingled that they cannot really be stratified into a hierarchy of layers that abstract one another. And that our "bones and flesh" really determine the way we think and perform cognitive functions. Were this to be the case, the only way for developmental robotics to further develop would be to use more human-like actuators and hardware, as done in [Holland, 2007]. This could eventually lead to the use of biological material in robotics, with all the implications that this would entail. Whether this is desirable is beyond the scope of science [Weizenbaum, 1976, Jonas, 1979].

6.7 Final words

When looking back on my work of the last few years, I found it to be an appealing metaphor of the development of the self, as we humans may experience it. Beyond the robotics application and the scientific developments conveyed in the preceding chapters, one can also a find a psychological or pedagogical message in this work, to which I was receptive, probably due to my experience as a parent, concomitant to the development of this thesis. As mentioned in Section 3.4.2, in practice the robot cannot learn its body schema only by looking at its limbs. This is due to its limited field of view, which make it quite improbable for its gaze to encounter its end-effector. In order to be able to look at itself, the robot must first learn the kinematic chain of its head, and this is done by moving the head and analyzing the resulting optical flow. This analysis is done with the assumption of a relatively stable background, i.e. the optical flow is assumed to be largely the result of the robot head and eye motion, and not of a moving background. Thus, it is only against a stable background that the robot can learn a representation of itself, which will then enable it to perform accurate movements and then learn new movements from experience. Similarly, a stable environment is often a prerequisite for the harmonic development of the child, for the building of a self which will then enable her to act meaningfully in its

social environment and learn from others [Bettelheim, 1987]. For my robot, as well as for the humans, the building of a self anchored and grounded in their environment is the basis for meaningful action and interaction with peers. The stability of this environment is a crucial factor in this process.

Appendix A

Mathematical background

A.1 Useful relationships

In this section we recall a some of identities used in this work. They are taken directly from [Zwillinger, 2003].

A.1.1 Trigonometric identities

$$\sin^2\alpha + \cos^2\alpha = 1 \tag{A.1}$$
$$\sin 2\alpha = 2\sin\alpha\cos\alpha \tag{A.2}$$
$$\cos 2\alpha = 1 - 2\sin^2\alpha \tag{A.3}$$
$$\sin\frac{\alpha}{2} = \pm\sqrt{\frac{1-\cos\alpha}{2}} \tag{A.4}$$
$$\sin(\alpha+\beta) = \sin\alpha\cos\beta + \cos\alpha\sin\beta \tag{A.5}$$
$$\sin(\alpha-\beta) = \sin\alpha\cos\beta - \cos\alpha\sin\beta \tag{A.6}$$
$$\cos(\alpha+\beta) = \cos\alpha\cos\beta - \sin\alpha\sin\beta \tag{A.7}$$
$$\cos(\alpha-\beta) = \cos\alpha\cos\beta + \sin\alpha\sin\beta \tag{A.8}$$
$$\cos\alpha\cos\beta = \frac{1}{2}\big(\cos(\alpha+\beta) + \cos(\alpha-\beta)\big) \tag{A.9}$$
$$\sin\alpha\sin\beta = \frac{1}{2}\big(\cos(\alpha-\beta) - \cos(\alpha+\beta)\big) \tag{A.10}$$
$$\sin\alpha - \sin\beta = 2\sin\frac{\alpha-\beta}{2}\cos\frac{\alpha+\beta}{2} \tag{A.11}$$

A.1.2 Vectorial identities

$$\mathbf{v}\times(\mathbf{u}\times\mathbf{w}) = \mathbf{w}^T\mathbf{v}\mathbf{u} - \mathbf{u}^T\mathbf{v}\mathbf{w} \tag{A.12}$$
$$(\mathbf{v}\times\mathbf{u})^T(\mathbf{w}\times\mathbf{z}) = \mathbf{v}^T\mathbf{w}\mathbf{u}^T\mathbf{z} - \mathbf{v}^T\mathbf{z}\mathbf{u}^T\mathbf{w} \tag{A.13}$$

A.1.3 Vector derivatives

Various conventions concerning vector derivatives can be found in the literature. In this thesis, derivatives of a scalar a with respect to a column vector $\mathbf{u} \in \mathbb{R}^n$

is a row vector:

$$\frac{\partial a}{d\mathbf{u}} = [\frac{\partial a}{du_1} \dots \frac{\partial a}{du_n}] \tag{A.14}$$

Hence the derivative of a column vector $\mathbf{v} \in \mathbb{R}^m$ by $\mathbf{u}$ is a matrix:

$$\frac{\partial \mathbf{v}}{d\mathbf{u}} = \begin{pmatrix} \frac{\partial v_1}{du_1} & \dots & \frac{\partial v_1}{du_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial v_m}{du_1} & \dots & \frac{\partial v_m}{du_n} \end{pmatrix} \tag{A.15}$$

Thus, the derivative with respect to $\mathbf{u}$ of the squared norm of a vector $\mathbf{v}(\mathbf{u})$ is given by

$$\frac{\partial \mathbf{v}^T\mathbf{v}}{d\mathbf{u}} = 2\mathbf{v}^T\frac{\partial \mathbf{v}}{d\mathbf{u}}. \tag{A.16}$$

A.2 Rotations

A.2.1 Rodrigues parametrization

Throughout this thesis, a vectorial parametrization of rotation [Bauchau and Trainelli, 2003] has been used. This parametrization, described below, has many advantages over the more standard matrix representation [Stuelpnagel, 1964]. The following is inspired by [Altmann, 1986], which provides an excellent treatment of rotations.

A rotation can be defined by its rotation axis and the rotation angle. It thus requires three parameters, two for the axis and one for the rotation angle ϕ. The parametrization we adopt was first published by Rodrigues [Altmann, 1989], and is given by a vector $\mathbf{b}$ belonging to the filled 3D ball of radius one centered on the origin, $\mathbf{b} \in \mathcal{B}_1$.

$$\mathbf{b} = \sin\frac{\phi}{2}\mathbf{a}, \tag{A.17}$$

where $\mathbf{a}$ is of unit length and colinear to the rotation axis. This parametrization amounts to considering only the first three components (the complex components) of the unit quaternion parametrization (see Section A.2.2). The fourth component of the quaternion can be easily retrieved from $\mathbf{b}$ and is given by

$$\alpha = \cos\frac{\phi}{2} = \sqrt{1 - \|\mathbf{b}\|^2}. \tag{A.18}$$

With this 3D parametrization, the inverse of a rotation $\mathbf{b}$ (i.e., rotating in the other direction) is very simply given by $-\mathbf{b}$, as $\sin(-\theta) = -sin(\theta)$. The image $\mathbf{y}$ of a vector $\mathbf{v}$ by a rotation parametrized by $\mathbf{b}$ is given by

$$\mathbf{y} = (1 - 2\mathbf{b}^T\mathbf{b})\mathbf{v} + 2\sqrt{(1 - \mathbf{b}^T\mathbf{b})}\mathbf{b} \times \mathbf{v} + 2(\mathbf{b}^T\mathbf{v})\mathbf{b}. \tag{A.19}$$

Using (A.3) and (A.2), in the three terms of this sum, one can see that this is equivalent to the Rodrigues formula

$$\mathbf{y} = \cos(\phi)\mathbf{v} + \sin(\phi)\mathbf{a} \times \mathbf{v} + (1 - \cos\phi)(\mathbf{a}^T\mathbf{v})\mathbf{a}. \tag{A.20}$$

From an implementation perspective (A.19) is preferable to (A.20) as square roots are faster to compute than sines and cosines.

The result of the composition of two rotations parametrized respectively by $\mathbf{b}_1$ and $\mathbf{b}_2$ (where $\mathbf{b}_1$ is performed first) is a rotation parametrized by $\mathbf{b}_3$

$$\mathbf{b}_3 = \mathbf{b}_2 * \mathbf{b}_1 \doteq \sqrt{1 - \|\mathbf{b}_1\|^2}\mathbf{b}_2 + \sqrt{1 - \|\mathbf{b}_2\|^2}\mathbf{b}_1 - \mathbf{b}_1 \times \mathbf{b}_2. \tag{A.21}$$

Note that this operation is not commutative as it involves the cross product, but nonetheless we have $\|\mathbf{b}_2 * \mathbf{b}_1\| = \|\mathbf{b}_1 * \mathbf{b}_2\|$ since those two vectors are related by a planar symmetry (the plane of symmetry is the one containing $\mathbf{b}_1$ and $\mathbf{b}_2$).

A.2.2 Quaternions

As mentioned above, quaternions are extensions of the Rodrigues vector with a fourth scalar component. A quaternion $\mathbf{q}$ represents a rotation of angle θ around a unit axis $\mathbf{a}$ if and only if it is of unit norm. Thus, following the notation of [Hestenes, 1999], the quaternion (also called a spinor) is defined by:

$$\mathbf{q} \doteq [\mathbf{b} \quad \alpha] \doteq [\sin(\frac{\theta}{2})\mathbf{a}^T \quad \cos(\frac{\theta}{2})]^T \tag{A.22}$$

The rotation resulting from the composition of two rotations is given by the (non-commutative) *quaternion product* between the two corresponding quaternions:

$$\mathbf{q}_3 = \mathbf{q}_2\mathbf{q}_1 = [(\alpha_1\mathbf{b}_2 + \alpha_2\mathbf{b}_1 - \mathbf{b}_1 \times \mathbf{b}_2)^T \quad \alpha_1\alpha_2 + \mathbf{b}_1^T\mathbf{b}_2]^T \tag{A.23}$$

While the quaternion representation is redundant, it has the advantage that the composition (or multiplication) operator, as defined in (A.23) is linear. Thus, when considering the derivative of the composition of rotations, we have:

$$\frac{\partial \mathbf{q}_2\mathbf{q}_1\mathbf{q}_0}{d\theta_1} = \mathbf{q}_2\frac{\partial \mathbf{q}_1}{d\theta_1}\mathbf{q}_0 \tag{A.24}$$

where $\frac{\partial \mathbf{q}}{d\theta}$ is given by the following quaternion

$$\frac{\partial \mathbf{q}}{d\theta} = \frac{1}{2}[\cos\frac{\theta}{2}\mathbf{a}^T \quad -\sin\frac{\theta}{2}]^T, \tag{A.25}$$

and (quaternion) multiplication is given by (A.23).
Similarly we have

$$\frac{\partial \mathbf{q}_2\mathbf{q}_1\mathbf{q}_0}{d\mathbf{a}_1} = \mathbf{q}_2\frac{\partial \mathbf{q}_1}{d\mathbf{a}_1}\mathbf{q}_0, \tag{A.26}$$

where $\frac{\partial \mathbf{q}}{d\mathbf{a}}$ is a "quaternion matrix" given by the concatenation of three quaternions:

$$\frac{\partial \mathbf{q}}{d\mathbf{a}} = \sin\frac{\theta}{2}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}. \tag{A.27}$$

The multiplications in (A.26) are of course quaternion multiplications, so that the multiplication of the "quaternion matrix" by other quaternions is performed

column by column. yielding another "quaternion matrix" as a result. So if the three columns of the "quaternion matrix" in (A.27) are denoted respectively by $\frac{\partial q}{da^1}$, $\frac{\partial q}{da^2}$ and $\frac{\partial q}{da^3}$, we have the following "quaternion matrix":

$$\frac{\partial \mathbf{q}_2\mathbf{q}_1\mathbf{q}_0}{d\mathbf{a}_1} = \mathbf{q}_2\frac{\partial \mathbf{q}_1}{d\mathbf{a}_1}\mathbf{q}_0 = \begin{pmatrix}\mathbf{q}_2\frac{\partial q_1}{da_1^1}\mathbf{q}_0 & \mathbf{q}_2\frac{\partial q_1}{da_1^2}\mathbf{q}_0 & \mathbf{q}_2\frac{\partial q_1}{da_1^3}\mathbf{q}_0\end{pmatrix}. \tag{A.28}$$

Finally, it also possible to compute the derivative of a quaternion with respect to both rotation axis and rotation angle:

$$\frac{\partial}{d\theta}\frac{\partial}{d\mathbf{a}}\mathbf{q} = \frac{1}{2}\cos\frac{\theta}{2}\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & 0\end{pmatrix}. \tag{A.29}$$

A.3 Dynamical System Theory

Here we recall some elements of dynamical system theory that are used in this thesis. A more complete introduction can be found in [Strogatz, 1994].
A continuous time dynamical system describes the evolution in time of a state variable $\mathbf{x}(t) \in \mathcal{X}$, where $t \in \mathbb{R}$ is the time and $\mathcal{X}$ the state-space, by an equation of the following type:

$$\dot{\mathbf{x}}(t) = f\big(\mathbf{x}(t), \mathbf{u}(t)\big), \tag{A.30}$$

where $\mathbf{u}(t)$ is an input to the system occurring at time t and f is a well-defined function describing how the system changes depending on its actual state and the input. In other words, at each infinitesimal time step t the system is in a state $\mathbf{x}(t)$, gets some external input $\mathbf{u}(t)$ and moves of an amount $\dot{\mathbf{x}}(t)$.

As an example, we can consider the spring-and-damper system depicted in Fig. A.1. This system is made of a mass attached at one end of a spring fixed on its other hand. Two forces act on the mass, the force f_s of the spring bringing the mass the the spring rest position, and a damping f_d proportional to the velocity due to friction. The state $\mathbf{x}$ of the system can be described by its position p and its velocity v. By Newton, we have

$$\ddot{p} = \frac{1}{m}(f_s + f_d) = k(p_o - p) - \alpha(v) = \alpha(-v + \beta(p_o - p)) \tag{A.31}$$

where $k, \alpha, \beta = k/\alpha$, are scalar positive constants. This can be rewritten as

$$\begin{pmatrix}\dot{p}\\ v\end{pmatrix} = \begin{pmatrix}0 & 1\\ -k & -\alpha\end{pmatrix}\begin{pmatrix}p\\ v\end{pmatrix} + \begin{pmatrix}0\\ k\end{pmatrix}p_o, \tag{A.32}$$

which is in the form of A.30, with $\mathbf{x} = [p\ v]$, $\mathbf{u} = p_o$.
If, as in (A.32) the function $f(\mathbf{x}(t), \mathbf{u}(t))$ is can be expressed in the matrix form

$$f(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{A}\ \mathbf{x}(t) + \mathbf{B}\ \mathbf{u}(t), \tag{A.33}$$

where $\mathbf{A}$ and $\mathbf{B}$ are matrices, the system is *linear*.

A question often arising in the context of dynamical systems is the question of the *fixed points*. Fixed points occur at the zeros of f, i.e., at values $\mathbf{x}, \mathbf{u}$

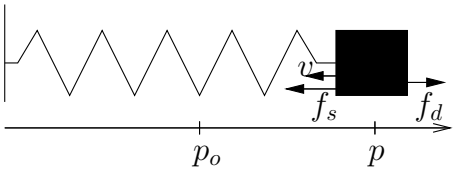


Figure A.1: The spring-and-damper system. A mass is attached to a spring fixed at its other end. The rest position of the spring is p_o. The mass is subject to the force f_s exerted by the spring and a damping force f_d proportional to its velocity v.

such that $f(\mathbf{x}, \mathbf{u}) = 0$. On those points the system remains theoretically in the same state. A central point of interests of fixed points is their stability under perturbation. A fixed point is *stable* if, when the system is moved slightly off the fixed point, it will return to it. Otherwise, it is *unstable*. Examples of stable and unstable fixed points are shown in Fig. A.2. Stable fixed points are also called *attractors* of the dynamical system, since a system in a nearby state will be attracted to the fixed point and remain there. If there is only one stable fixed point, it is said to be a *global attractor* of the system, as the system will eventually end up in this state, no matter what.
The spring and damper system described by (A.32) has $[p\ v] = [p_o\ 0]$ as a fixed point. Intuitively, it is easy to see that it is stable, because if one slightly moves the mass away from its resting point p_o, it will go back to it.

Linear continuous-time dynamical systems have a single global attractor if and only if all eigenvalues of $\mathbf{A}$ have a strictly negative real part. For non-linear systems, a sufficient condition for the existence of a global attractor, is the existence of a so-called *Lyapunov function* of the system. A function $L(\mathbf{x}) : \mathcal{X} \to \mathbb{R}^+$ is said to be a Lyapunov function of the system A.30 if there exists a point $\mathbf{x}_0 \in \mathcal{X}$ satisfying the three following conditions:

1. $L(\mathbf{x}) > 0 \quad \forall \mathbf{x} \neq \mathbf{x}_0$
2. $\frac{\partial}{dt} L < 0 \quad \forall \mathbf{x} \neq \mathbf{x}_0$
3. $L(\mathbf{x}_0) = \frac{\partial}{dt} L \mid_{\mathbf{x}_0} = 0$

In other words, a Lyapunov function can be seen as a kind of "potential" defined over all states of the system, whereby the system is always driven to states of lower potential, except in one particular point $\mathbf{x}_0$ which is the point with the lowest potential. This point is thus the attractor of the dynamical system. A example is given in Fig A.2. For the spring-and-damper system given by (A.32), a possible Lyapunov function is given by the total energy of the system: $L(\mathbf{x}) = k(p_o - p)^2 + \alpha v^2$

A.4 Gaussian Mixture Regression

Gaussian Mixture Regression (GMR) is a method suggested by [Ghahramani and Jordan, 1994] for statistically estimating a function $\mathcal{F}_\xi(\zeta)$ given by a "training set" of N examples $\{(\zeta^i, \xi^i)\}_{i=1}^N$, where ξ^i is a noisy measurement of $\mathcal{F}_\xi(\zeta^i)$:

$$\xi^i = \mathcal{F}_\xi(\zeta^i) + \epsilon^i$$

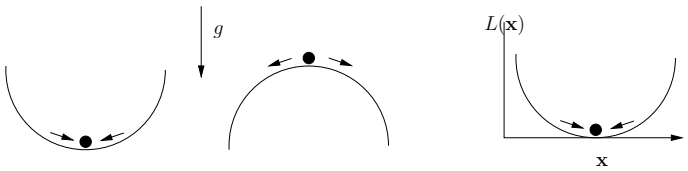


Figure A.2: Stability of fixed points. Left: a ball submitted to gravity g in a bowl. The bottom of the bowl is a stable fixed point of the system, as the bowl will return to it after being slightly moved out of it. Center: A ball submitted to gravity in equilibrium on a bowl turned upside down. The ball is on an unstable fixed point, as the ball will not return to it if slightly moved away from it. Right: An example of a Lyapunov function for the system described on the left. The state $\mathbf{x}$ is given by the horizontal position of the ball, and the Lyapunov function is given the height of the bowl at the corresponding point. The ball will always go down this function, which has a single minimum.

(ϵ^i is the Gaussian noise). The idea is to model the joint distribution of the "input" variable ζ and an "output" variable ξ as a Gaussian Mixture Model. If we join those variables in a vector $\upsilon = [\zeta^T \xi^T]^T$, it is possible to model its probability density function as a mixture of K Gaussian functions

$$p(\upsilon) = \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\upsilon; \mu_k, \mathbf{\Sigma}_k\big), \quad \text{such that} \quad \sum_{k=1}^{K} \pi_k = 1$$

where the $\pi_k \in [0\ 1]$ are the priors, and $\mathcal{N}(\upsilon; \mu_k, \mathbf{\Sigma}_k)$ is a Gaussian function with mean μ_k and covariance matrix $\mathbf{\Sigma}_k$:

$$\mathcal{N}\big(\upsilon; \mu_k, \mathbf{\Sigma}_k\big) = \big((2\pi)^d |\mathbf{\Sigma}_k|\big)^{-\frac{1}{2}} \exp\big(-\frac{1}{2}(\upsilon - \mu_k)^T \mathbf{\Sigma}_k^{-1} (\upsilon - \mu_k)\big),$$

where d is the dimensionality of the vector υ. The mean vectors μ_k and covariance matrices $\mathbf{\Sigma}_k$ can be separated into their respective input and output components:

$$\mu_k = [\mu_{k,\zeta}^T\ \mu_{k,\xi}^T]^T \qquad \mathbf{\Sigma}_k = \begin{pmatrix} \mathbf{\Sigma}_{k,\zeta} & \mathbf{\Sigma}_{k,\zeta\xi} \\ \mathbf{\Sigma}_{k,\xi\zeta} & \mathbf{\Sigma}_{k,\xi} \end{pmatrix}$$

This GMM can be trained using a standard E-M algorithm, taking the demonstrations as training data. Gaussian centers are initialized using k-means.
We thus obtain a joint probability density function for the input and the output. Because it is a GMM, the conditional probability density function, i.e., the probability of the output conditioned on the input is also a GMM. Hence, it is possible, after training, to recover the expected output variable $\tilde{\xi}$, given the observed input variable ζ.

$$\tilde{\xi} = \tilde{\mathcal{F}}_\xi(\zeta) = \sum_{k=1}^{K} h_k(\zeta)\big(\mu_{k,\xi} + \mathbf{\Sigma}_{k,\xi\zeta}\mathbf{\Sigma}_{k,\zeta}^{-1}(\zeta - \mu_{k,\zeta})\big), \tag{A.34}$$

where the $h_k(\zeta)$ are given by:

$$h_k(\zeta) = \frac{\pi_k \mathcal{N}(\zeta; \mu_{k,\zeta}, \mathbf{\Sigma}_{k,\zeta})}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\zeta; \mu_{k,\zeta}, \mathbf{\Sigma}_{k,\zeta})}, \tag{A.35}$$

and the tilde ($\tilde{}$) sign indicates that we are dealing with expectation values.

Appendix B

Mathematical Developments

B.1 Proofs of convergence for single joint cases

B.1.1 Proof of Proposition 1

.

Proof We consider equations (3.2) and (3.3) as a continuous time dynamical system in the parameter space defined by $\mathbf{l}$ and $\mathbf{a}$. Indeed, the learning step ϵ can be interpreted as the integration constant of the dynamical system, in which case $\epsilon^{-1}\Delta\mathbf{l}$ and $\epsilon^{-1}\Delta\mathbf{a}$ tend respectively to $\frac{\partial}{dt}\mathbf{l}$ and $\frac{\partial}{dt}\mathbf{a}$, where t denotes the time.
In order to simplify the notation, we put the two parameter vectors $\mathbf{l}$ and $\mathbf{a}$ into a single parameter vector $\mathbf{p} = [\mathbf{l}^T\mathbf{a}^T]^T$. We then consider the function $\mathbf{E}(\mathbf{p})$ defined by:

$$\mathbf{E}(\mathbf{p}) = \langle \mathbf{E}^\theta(\mathbf{p}) \rangle, \quad \text{with} \quad \mathbf{E}^\theta(p) = \frac{1}{2}\Big|\Big|\mathcal{T}^*(\mathbf{v}_n) - \mathcal{T}_\mathbf{p}(\mathbf{v}_n)\Big|\Big|^2, \tag{B.1}$$

where $\mathcal{T}_\mathbf{p}$ is the estimated rigid-body transformation parametrized by $\mathbf{p}$ and dependent on the joint configuration θ ($\mathcal{T}_\mathbf{p} = \mathbf{l}+\mathbf{R}^\theta_\mathbf{a}(\mathbf{v}_n)$),$\mathcal{T}^*(\mathbf{v}_n) = \mathbf{l}^*+\mathbf{R}^\theta_{\mathbf{a}^*}(\mathbf{v}_n)$ is the real underlying transformation, and $\langle\rangle$ denotes the expectation operator assuming θ follows symmetric pdf centered on 0. Since grad is a linear operator, the $\langle[\frac{\partial}{dt}\mathbf{l}\ \frac{\partial}{dt}\mathbf{a}]\rangle = -\text{grad}\mathbf{E}(\mathbf{p})$. So (3.2) and (3.3) correspond to a gradient descent on $\mathbf{E}$.
It remains to be shown that if $\frac{\partial}{dt}\mathbf{E}(\mathbf{p}) = 0$, then $\mathcal{T}_p = \mathcal{T}^*$, which amounts to saying that there is no local minima to $\mathbf{E}$. To this end we compute the gradient of $\mathbf{E}$.

$$\begin{aligned} \text{grad}_\mathbf{p}\mathbf{E} &= \langle \big(\mathcal{T}_\mathbf{p}(\mathbf{v}) - \mathcal{T}^*(\mathbf{v})\big)^T \frac{\partial}{d\mathbf{p}}\mathcal{T}_\mathbf{p}\rangle & \text{(B.2)} \\ &= \langle \big(\mathcal{T}_\mathbf{p}(\mathbf{v}) - \mathcal{T}^*(\mathbf{v})\big)^T [\mathbf{I}\ \bar{\mathbf{R}}^\theta_\mathbf{a}(\mathbf{v})]\rangle & \text{(B.3)} \end{aligned}$$

We can divide the gradient vector onto its two components $\text{grad}_\mathbf{p}\mathbf{E} = [\text{grad}_\mathbf{l}\mathbf{E}\ \text{grad}_\mathbf{a}\mathbf{E}]$.

We then have

$$\begin{aligned}
\mathrm{grad}_{\mathbf{l}}\mathbf{E} &= \big(\langle\sin\theta\rangle(\mathbf{a}-\mathbf{a}^*)\times\mathbf{v} + \langle 1-\cos\theta\rangle(\mathbf{v}^T\mathbf{a}\mathbf{a} - \mathbf{v}^T\mathbf{a}^*\mathbf{a}^*) + \mathbf{l}-\mathbf{l}^*\big)^T && \text{(B.4)}\\
\mathrm{grad}_{\mathbf{a}}\mathbf{E} &= \langle\big(\sin\theta(\mathbf{a}-\mathbf{a}^*)\times\mathbf{v} + (1-\cos\theta)\big(\mathbf{v}^T\mathbf{a}\mathbf{a} - \mathbf{v}^T\mathbf{a}^*\mathbf{a}^*\big) + \mathbf{l}-\mathbf{l}^*\big)^T \\
&\quad \big(\sin\theta\mathbf{v}\uparrow + (1-\cos\theta)(\mathbf{a}\mathbf{v}^T + (\mathbf{a}^T\mathbf{v})\mathbf{I})\big)\rangle && \text{(B.5)}
\end{aligned}$$

Developing this product leads to the following sum:

$$\begin{aligned}
\mathrm{grad}_{\mathbf{a}}\mathbf{E} &= \langle\sin^2\theta\rangle\mathbf{v}\times\big((\mathbf{a}-\mathbf{a}^*)\times\mathbf{v}\big) + \langle(1-\cos\theta)\sin\theta\rangle\big(\mathbf{v}\times(\mathbf{a}-\mathbf{a}^*)\big)^T \\
&\quad (\mathbf{a}\mathbf{v}^T + \mathbf{a}^T\mathbf{v}\mathbf{I}) + \langle\sin\theta(1-\cos\theta)\rangle\big(\mathbf{v}\times(\mathbf{v}^T\mathbf{a}\mathbf{a} - \mathbf{v}^T\mathbf{a}^*\mathbf{a}^*)\big) \\
&\quad +\langle(1-\cos\theta)^2\rangle(\mathbf{v}^T\mathbf{a}\mathbf{a} - \mathbf{v}^T\mathbf{a}^*\mathbf{a}^*)^T(\mathbf{a}\mathbf{v}^T + \mathbf{a}^T\mathbf{v}\mathbf{I}) \\
&\quad +(\mathbf{l}-\mathbf{l}^*)^T\big(\langle\sin\theta\rangle\mathbf{v}\uparrow + \langle 1-\cos\theta\rangle(\mathbf{a}\mathbf{v}^T + \mathbf{a}^T\mathbf{v}\mathbf{I})\big) && \text{(B.6)}
\end{aligned}$$

As the θ are zero-centered and symmetric, we have $\langle\sin\theta\rangle = \langle\sin(\theta)(1-\cos\theta)\rangle = 0$. Thus, we have

$$\begin{aligned}
\mathrm{grad}_{\mathbf{l}}\mathbf{E} &= \big(\langle 1-\cos\theta\rangle(\mathbf{v}^T\mathbf{a}\mathbf{a} - \mathbf{v}^T\mathbf{a}^*\mathbf{a}^*) + \mathbf{l}-\mathbf{l}^*\big)^T && \text{(B.7)}\\
\mathrm{grad}_{\mathbf{a}}\mathbf{E} &= \langle\sin^2\theta\rangle\big(\|\mathbf{v}\|^2(\mathbf{a}-\mathbf{a}^*) - \mathbf{v}^T(\mathbf{a}-\mathbf{a}^*)\mathbf{v}\big) \\
&\quad +\langle(1-\cos\theta)^2\rangle(\mathbf{v}^T\mathbf{a}\mathbf{a} - \mathbf{v}^T\mathbf{a}^*\mathbf{a}^*)^T(\mathbf{a}\mathbf{v}^T + \mathbf{a}^T\mathbf{v}\mathbf{I}) \\
&\quad +\langle 1-\cos\theta\rangle(\mathbf{l}-\mathbf{l}^*)^T(\mathbf{a}\mathbf{v}^T + \mathbf{a}^T\mathbf{v}\mathbf{I}) && \text{(B.8)}
\end{aligned}$$

Now, $\mathbf{p}$ is a fixed point of the dynamical system if and only $\mathrm{grad}_{\mathbf{l}}\mathbf{E} = \mathbf{0}$ and $\hat{\mathbf{a}}$ is colinear to $\mathbf{a}$ (thus taking the normalization of $\mathbf{a}$ into account). Setting $\mathrm{grad}_{\mathbf{l}}\mathbf{E}$ to zero, we obtain

$$\begin{aligned}
\mathbf{l}-\mathbf{l}^* &= -\langle 1-\cos\theta\rangle(\mathbf{v}^T\mathbf{a}\mathbf{a} - \mathbf{v}^T\mathbf{a}^*\mathbf{a}^*) && \text{(B.9)}\\
\mathrm{grad}_{\mathbf{a}}\mathbf{E} &= \langle\sin^2\theta\rangle\big(\|\mathbf{v}\|^2(\mathbf{a}-\mathbf{a}^*) - \mathbf{v}^T(\mathbf{a}-\mathbf{a}^*)\mathbf{v}\big)^T + \\
&\quad \big(\langle(1-\cos\theta)^2\rangle - \langle(1-\cos\theta)\rangle^2\big)(\mathbf{v}^T\mathbf{a}\mathbf{a} - \mathbf{v}^T\mathbf{a}^*\mathbf{a}^*)^T(\mathbf{a}\mathbf{v}^T + \mathbf{a}^T\mathbf{v}\mathbf{I}) && \text{(B.10)}\\
&= \langle\sin^2\theta\rangle\big(\|\mathbf{v}\|^2(\mathbf{a}-\mathbf{a}^*) - \mathbf{v}^T(\mathbf{a}-\mathbf{a}^*)\mathbf{v}\big)^T + \\
&\quad \mathrm{var}(\cos\theta)\big(\mathbf{v}^T\mathbf{a}\mathbf{v} + (\mathbf{v}^T\mathbf{a})^2\mathbf{a} - \mathbf{v}^T\mathbf{a}^*\mathbf{a}^T\mathbf{a}^*\mathbf{v} - \mathbf{v}^T\mathbf{a}^*\mathbf{v}^T\mathbf{a}\mathbf{a}^*\big)^T && \text{(B.11)}
\end{aligned}$$

With no loss of generality and to simplify the notation, we can drop $\|\mathbf{v}\|^2$ and consider $\mathbf{v}$ to be of unit norm. To further lighten the notation, we define $c = \mathrm{var}(\cos\theta)$ and $s = \langle\sin^2\theta\rangle = \mathrm{var}(\sin\theta)$. We have a fixed point if and only if $\mathrm{grad}_{\mathbf{a}}\mathbf{E}$ is colinear to $\mathbf{a}$ or equivalently if it is perpendicular to the projection of any vector $\mathbf{r}$ on the plane orthogonal to $\mathbf{a}$, $(\mathbf{I}-\mathbf{a}\mathbf{a}^T)\mathbf{r}$. Hence we have

$$\begin{aligned}
\mathbf{0} &= \mathrm{grad}_{\mathbf{a}}\mathbf{E}\cdot(\mathbf{I}-\mathbf{a}\mathbf{a}^T) && \text{(B.12)}\\
&= s(-\mathbf{a}^* + \mathbf{a}^T\mathbf{a}^*\mathbf{a} - \mathbf{v}^T\mathbf{a}\mathbf{v} + \mathbf{v}^T\mathbf{a}^*\mathbf{v} + (\mathbf{v}^T\mathbf{a})^2\mathbf{a} - \mathbf{v}^T\mathbf{a}\mathbf{v}^T\mathbf{a}^*\mathbf{a}) + \\
&\quad c(\mathbf{v}^T\mathbf{a}\mathbf{v} - (\mathbf{v}^T\mathbf{a})^2\mathbf{a} - \mathbf{v}^T\mathbf{a}^*\mathbf{a}^T\mathbf{a}^*\mathbf{v} + \mathbf{v}^T\mathbf{a}\mathbf{v}^T\mathbf{a}^*\mathbf{a}^T\mathbf{a}^*\mathbf{a} - \\
&\quad \mathbf{v}^T\mathbf{a}^*\mathbf{v}^T\mathbf{a}\mathbf{a}^* + \mathbf{v}^T\mathbf{a}^*\mathbf{v}^T\mathbf{a}\mathbf{a}^T\mathbf{a}^*\mathbf{a}) && \text{(B.13)}\\
&= s(\mathbf{v}^T\mathbf{a}^* - \mathbf{v}^T\mathbf{a})(\mathbf{v}-\mathbf{a}^T\mathbf{v}\mathbf{a}) - \mathbf{a}^* + \mathbf{a}^T\mathbf{a}^*\mathbf{a} \\
&\quad +c(\mathbf{v}^T\mathbf{a} - \mathbf{v}^T\mathbf{a}^*\mathbf{a}^T\mathbf{a}^*)(\mathbf{v}-\mathbf{a}^T\mathbf{v}\mathbf{a}) - \mathbf{v}^T\mathbf{a}\mathbf{v}^T\mathbf{a}^*(\mathbf{a}^* - \mathbf{a}^T\mathbf{a}^*\mathbf{a}) && \text{(B.14)}\\
&= \big(s(\mathbf{v}^T\mathbf{a}^* - \mathbf{v}^T\mathbf{a}) + c(\mathbf{v}^T\mathbf{a} - \mathbf{v}^T\mathbf{a}^*\mathbf{a}^T\mathbf{a}^*)\big)(\mathbf{v}-\mathbf{a}^T\mathbf{v}\mathbf{a}) \\
&\quad -\big(c\mathbf{v}^T\mathbf{a}\mathbf{v}^T\mathbf{a}^* + s\big)(\mathbf{a}^* - \mathbf{a}^T\mathbf{a}^*\mathbf{a}) && \text{(B.15)}
\end{aligned}$$

We now notice that this last expression is the weighted sum of two vectors $\mathbf{v} - \mathbf{a}^T\mathbf{v}\mathbf{a}$ and $\mathbf{a}^* - \mathbf{a}^T\mathbf{a}^*\mathbf{a}$. Those are respectively the projections of $\mathbf{v}$ and $\mathbf{a}^*$ on the plane orthogonal to $\mathbf{a}$. This implies that $\mathbf{a}, \mathbf{v}, \mathbf{a}^*$ are coplanar and we can rewrite (B.15) as

$$0 = \big(s(\cos\alpha^* - \cos\alpha) + c(\cos\alpha - \cos\alpha^*\cos\gamma)\big)\sin\alpha - \big(s + c\cos\alpha\cos\alpha^*\big)\sin\gamma, \tag{B.16}$$

where α is the angle between $\mathbf{v}$ and $\mathbf{a}$, α^* is the angle between $\mathbf{a}^*$ and $\mathbf{v}$, and γ is the angle between $\mathbf{a}$ and $\mathbf{a}^*$, as depicted in Fig. B.1, left. Since $\gamma = \alpha^* - \alpha$, we have

$$\sin\gamma = \sin\alpha^*\cos\alpha - \cos\alpha^*\sin\alpha \quad \cos\gamma = \cos\alpha\cos\alpha^* + \sin\alpha\sin\alpha^* \tag{B.17}$$

Inserting (B.17) into (B.16) yields

$$\begin{aligned}
0 \quad &= \quad \big(s(\cos\alpha^* - \cos\alpha) + c(\cos\alpha - \cos\alpha^*(\cos\alpha\cos\alpha^* + \sin\alpha\sin\alpha^*))\big)\sin\alpha \\
&\quad -\big(s + c\cos\alpha\cos\alpha^*\big)(\sin\alpha^*\cos\alpha - \cos\alpha^*\sin\alpha) && \text{(B.18)} \\
&= \quad \big(c - s\big)\cos\alpha\sin\alpha + 2s\cos\alpha^*\sin\alpha - s\sin\alpha^*\cos\alpha + c(-\cos^2\alpha^*\cos\alpha\sin\alpha \\
&\quad -\cos\alpha^*\sin\alpha^*\sin^2\alpha - \cos\alpha^*\sin\alpha^*\cos^2\alpha + \cos^2\alpha^*\cos\alpha\sin\alpha) && \text{(B.19)} \\
&= \quad (c - s)\cos\alpha\sin\alpha + 2s\cos\alpha^*\sin\alpha - s\sin\alpha^*\cos\alpha - c\cos\alpha^*\sin\alpha^* && \text{(B.20)} \\
&\overset{A.2}{=} \quad \frac{1}{2}(c - s)\sin(2\alpha) + s\sin(\alpha - \alpha^*) + s\cos\alpha^*\sin\alpha - \frac{c}{2}\sin(2\alpha^*) && \text{(B.21)} \\
&\overset{A.5,A.6}{=} \quad \frac{1}{2}(c - s)\sin(2\alpha) + \frac{3s}{2}\sin(\alpha - \alpha^*) + \frac{s}{2}\sin(\alpha + \alpha^*) - \frac{c}{2}\sin(2\alpha^*) && \text{(B.22)}
\end{aligned}$$

By performing the following change of variables: $\beta = \alpha - \alpha^*$, $\phi = \alpha + \alpha^*$, we can rewrite this last equation as

$$\begin{aligned}
0 \quad &= \quad \frac{1}{2}(c - s)\sin(\beta + \phi) + \frac{3s}{2}\sin\beta + \frac{s}{2}\sin\phi + \frac{c}{2}\sin(\beta - \phi) && \text{(B.23)} \\
&\overset{A.10}{=} \quad c\sin\beta\cos\phi + \frac{3s}{2}\sin\beta + \frac{s}{2}\big(\sin\phi - \sin(\beta + \phi)\big) && \text{(B.24)} \\
&\overset{A.11}{=} \quad (c\cos\phi + \frac{3s}{2})\sin\beta - s\sin\frac{\beta}{2}\cos(\frac{\beta}{2} + \phi) && \text{(B.25)} \\
&\overset{A.2}{=} \quad (2c\cos\phi + 3s)\sin\frac{\beta}{2}\cos\frac{\beta}{2} - s\sin\frac{\beta}{2}\cos(\frac{\beta}{2} + \phi) && \text{(B.26)} \\
&= \quad \sin\frac{\beta}{2}\big((2c\cos\phi + 3s)\cos\frac{\beta}{2} - s\cos(\frac{\beta}{2} + \phi)\big) && \text{(B.27)} \\
&\overset{A.3}{=} \quad \sin\frac{\beta}{2}\big((2c\cos\phi + 3s)\cos\frac{\beta}{2} - s(\cos\frac{\beta}{2}\cos\phi - \sin\frac{\beta}{2}\sin\phi)\big) && \text{(B.28)} \\
&= \quad \sin\frac{\beta}{2}\Big(\big((2c - s)\cos\phi + 3s\big)\cos\frac{\beta}{2} + s\sin\phi\sin\frac{\beta}{2}\Big)) && \text{(B.29)} \\
&= \quad s\sin\frac{\beta}{2}\cos\frac{\beta}{2}\sin\phi\big(\frac{(2c - s)\cos\phi + 3s}{s\sin\phi} + \tan\frac{\beta}{2}\big) && \text{(B.30)}
\end{aligned}$$

Two zeros of this last equation are given by $\frac{\beta}{2} = 0$ and $\frac{\beta}{2} = \frac{\pi}{2}$, in other words $\alpha = \alpha^*$ and $\alpha = \alpha^* + \pi$. An examination of (B.29), shows that $\phi = 0$ cannot be a zero of (B.29) and $\phi = \pi$ is a zero of (B.29) only if $2c - s = 3s$, in other

words $c = 2s$. The other zeros of (B.30) are given by

$$0 = \frac{(2c - s)\cos\phi + 3s}{s\sin\phi} + \tan\frac{\beta}{2} \quad \text{(B.31)}$$

$$\beta = f(\phi) \doteq 2\ \text{atan}\big(-\frac{1}{\sin\phi}(d\cos\phi + 3)\big), \quad \text{(B.32)}$$

where $d = \frac{2c}{s} - 1$ is introduced to lighten the notation. So for each $\phi = \alpha + \alpha^*$, there is only one value $\beta = \alpha - \alpha^*$ such that (B.31) holds. This however, does not yet mean that for each α^*, there is only one α such that (B.31) holds.
In order to check whether this is the case, let us consider the domain defined by α^* and α. In this domain, it is possible to draw the FoR defined by ϕ and β and in this FoR, the function f defined by (B.31) and the FoR defined by α^* and α as done in Fig. B.1, right. The number of zeros in (B.31) for each α^* is given by the number of intersections between $f(\phi)$ and the parallel to α going through α^*. To see that there is no more than one such intersection, let us consider the behavior of $f(\phi)$, when ϕ is close to zero. In that point, f is ill-defined, as in tends to $-\pi$ if $\phi > 0$ and π if $\phi < 0$. In both cases the slope of f when $\phi \approx 0$ is given by

$$\frac{\partial}{d\phi}f(\phi) = \frac{1}{1 + (\frac{d\cos\phi+3}{\sin\phi})^2}\frac{d\sin^2\phi + d\cos^2\phi + 3\cos\phi}{\sin^2\phi} = \frac{d + 3\cos\phi}{\sin^2\phi + (d\cos\phi + 3)^2}. \quad \text{(B.33)}$$

We now show that this is always smaller than 1,$\forall d \in [-1\ 3]$.

For simplicity we can drop $\sin^2\phi$ in the denominator and show that

$$d+3\cos\phi < (d\cos\phi+3)^2 \Leftrightarrow g(\phi) \doteq d^2\cos^2\phi+(6d-3)\cos\phi+9-d > 0 \quad \text{(B.34)}$$

To find the extrema of this function we derive with respect to ϕ.

$$\frac{\partial}{d\phi}g(\phi) = -d^2\cos\phi\sin\phi - (6d-3)\sin\phi = \sin\phi(-d^2\cos\phi - 6d + 3) \quad \text{(B.35)}$$

The extremas are given by $\phi = 0$, $\phi = \pi$ and possibly $cos\phi = -(6d-3)/d^2$. If $cos\phi = -(6d-3)/d^2$ then

$$g(\phi) = \frac{(6d-3)^2}{d^2} - \frac{(6d-3)^2}{d^2} + 9 - d > 0 \quad \forall d < 9. \quad \text{(B.36)}$$

If $\phi = \pi$ then

$$g(\pi) = d^2 - 7d + 12 = (d-3)(d-4) > 0 \quad \forall d < 3. \quad \text{(B.37)}$$

If $\phi = 0$ then

$$g(0) = d^2 + 5d + 6 = (d+2)(d+3) > 0 \quad \forall d > -2. \quad \text{(B.38)}$$

So, if $-1 < d < 3$, $g(\phi)$ is positive at its minimas and hence over its entire domain. Thus the derivative of f is smaller than 1. This means that the slope of f is always smaller than 1. Thus, the parallel to α (which is of slope 1) through any α^* will have at most one intersection with f (see Fig B.1, right).
To summarize, if $d < 3$, i.e. $c < 2s$, the zeros of (B.22) are $\alpha = \alpha^*$ (the minimum of $\mathbf{E}$), $\alpha = \alpha^* + \pi$, and possibly another zero. So there is a single minimum, a maximum and possibly a plateau as $\mathbf{a}$ lives on a sphere. They are all on the plane defined by $\mathbf{a}^*$ and $\mathbf{v}$. Hence the algorithm will always converge to $\mathbf{a}^*$.

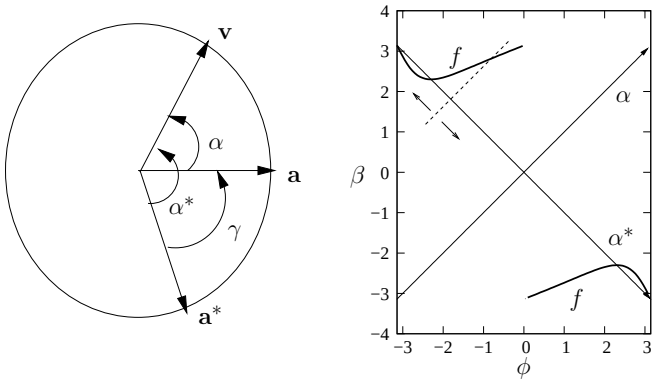


Figure B.1: Left: Going from (B.15) to (B.16). All three fixed points $\mathbf{a}^*$, $\mathbf{a}_1$ and $\mathbf{a}_2$ are located on the plane defined by $\mathbf{a}^*$ and $\mathbf{v}$. Right: the zeros of B.31 are given by the function f, plotted in bold. We show that when sliding the the dashed line along α^*, it as at most one intersection with f. This is because the slope of f is always smaller than the slope of the dashed line.

B.1.2 Proof of Proposition 2

Proof Similarly to what was done in the preceding demonstration, we consider (3.19) as a continuous time dynamical system in $\mathbf{a}$ and show that $\mathbf{a}^*$ is the global attractor of this dynamical system.
We consider the function given by:

$$\mathbf{E}(\mathbf{a}) = \langle \mathbf{E}^\theta(\mathbf{a}) \rangle, \quad \text{with} \quad \mathbf{E}^\theta(\mathbf{a}) = \frac{1}{2} \Big|\Big| (\mathbf{b}' - \mathbf{b}) \Big|\Big|^2, \tag{B.39}$$

where $\mathbf{b}'$ is the "true" resulting rotation and $\mathbf{b}$ is the estimate resulting rotation. Of course, (3.19) performs a gradient descent on $\mathbf{E}(\mathbf{a})$, because the gradient of the mean of $\mathbf{E}^\theta$ is the mean gradient of the $\mathbf{E}^\theta$ (the gradient is a linear operator). It remains to be show that grad$\mathbf{E}$ is zero only if $\mathbf{a} = \mathbf{a}^*$. The single joint version of (3.19) is

$$\Delta \mathbf{a} = -\tau \epsilon (\mathbf{b}' - \mathbf{b})^T \bar{\mathbf{R}}' \dot{\theta} \tag{B.40}$$

Moreover, in the single joint case, $\mathbf{b}' = \frac{1}{2} \sin\dot{\theta} \mathbf{a}^*$ and $\mathbf{b} = \frac{1}{2} \sin\dot{\theta} \mathbf{a}$. Like in the previous demonstration, we find values for $\mathbf{a}$ such that the projection of grad$\mathbf{E}(\mathbf{a})$ is zero or parallel to $\mathbf{a}$, so that after normalization $\mathbf{a}$ remains unchanged. So we have

$$0 = \text{grad}\mathbf{E} \cdot (\mathbf{I} - \mathbf{a}\mathbf{a}^T) = -\tau \langle \sin\frac{\dot{\theta}}{2} (\mathbf{a}^* - \mathbf{a})^T \frac{1}{2} \cos\frac{\theta}{2} \mathbf{I}_3 \dot{\theta} \rangle (\mathbf{I} - \mathbf{a}\mathbf{a}^T) \tag{B.41}$$

$$= -\frac{\tau}{4} \langle \dot{\theta}^2 \rangle \langle \cos\frac{\theta}{2} \rangle (\mathbf{a}^* - \mathbf{a})^T (\mathbf{I} - \mathbf{a}\mathbf{a}^T), \tag{B.42}$$

because for small $\dot{\theta}$, $\sin\dot{\theta} \approx \dot{\theta}$. Developing further, we have

$$0 = -\frac{\tau}{4} \langle \dot{\theta}^2 \rangle \langle \cos\frac{\theta}{2} \rangle (\mathbf{a}^* - \mathbf{a}^T \mathbf{a}^* \mathbf{a}). \tag{B.43}$$

This last equation is zero if and only if $\mathbf{a} = \pm \mathbf{a}^*$, showing that $\mathbf{a}^*$ is the only attractor of $\mathbf{a}$, with $-\mathbf{a}^*$ being the repellor.

B.2 Inverse kinematics for orientation

When implementing the global inverse kinematics, I basically used the Cyclic Coordinate Descent (CCD) method [Wang and Chen, 1991] summarized in Section 2.3.3. However, when dealing with the orientation component, this method was slightly modified. The following describes how.

As explained in Section 2.3.3 at each iteration, for a given configuration, the method tries to update a given single joint in order bring the end-effector closer to the target. When considering position only, this is done by maximizing the dot product between $xT - \xi$ and $xT - \mathbf{x}$, where $\mathbf{x}$ is the position of the end-effector, as shown in Fig. 2.5.
In the current implementation, a rotation is described by a 3D vector as described in Section A.2.1. The end-effector orientation is given by the vector $\mathbf{b}$ describing the rotation of the end-effector from its orientation at the zero position. This rotation is given by

$$\mathbf{b} = \mathbf{b}_1 * \mathbf{b}_2 * \cdots * \mathbf{b}_n, \tag{B.44}$$

where the $\mathbf{b}_i$ are the rotations of joints i (i.e., of direction $\mathbf{a}_i$ and length $\sin\frac{\theta_i}{2}$) and $*$ denotes the rotation composition operator described by (A.21).
At each iteration of the CCD algorithm (for a given joint i), the aim is to minimize set θ_i so that $\mathbf{b}$ gets closer to the target orientation $\mathbf{b}_T$. Unlike in the standard CCD algorithm, this is done by considering the "difference" rotation $\bar{\mathbf{b}}$:

$$\bar{\mathbf{b}} = \mathbf{b} * (-\mathbf{b}_T). \tag{B.45}$$

More precisely, at each time step, the algorithm minimizes the squared norm $\|\bar{\mathbf{b}}\|^2$ of this difference rotation. So if $\mathbf{b}$ and $\mathbf{b}_T$ are equal, $\bar{\mathbf{b}}$ is the identity, which is represented by the null vector and thus has zero norm. Since (see Section A.2.1) $\forall \mathbf{b}_1, \mathbf{b}_2 \quad \|\mathbf{b}_1 * \mathbf{b}_2\| = \|\mathbf{b}_2 * \mathbf{b}_1\|$, we have

$$\begin{aligned}
\|\bar{\mathbf{b}}\| &= \|\mathbf{b} * (-\mathbf{b}_T)\| = \|\mathbf{b}_1 * \mathbf{b}_2 * \cdots * \mathbf{b}_n * (-\mathbf{b}_T)\| && \text{(B.46)}\\
&= \|\mathbf{b}_2 * \cdots * \mathbf{b}_n * (-\mathbf{b}_T) * \mathbf{b}_1\| = \ldots && \text{(B.47)}\\
&= \|\mathbf{b}_i * \cdots * \mathbf{b}_n * (-\mathbf{b}_T) * \mathbf{b}_1 * \cdots * \mathbf{b}_{i-1}\| \quad \forall i \in 1, \ldots, n && \text{(B.48)}
\end{aligned}$$

Thus, we need to compute

$$\min_{\theta_i} \|\mathbf{b}_i * \tilde{\mathbf{b}}\|^2 \quad \text{with} \quad \tilde{\mathbf{b}} = \mathbf{b}_{i+1} * \cdots * \mathbf{b}_n * (-\mathbf{b}_T) * \mathbf{b}_1 * \cdots * \mathbf{b}_{i-1}. \tag{B.49}$$

This is solved by setting the derivative to zero: In order to lighten the notation, we first define $\tilde{\alpha} = \sqrt{1 - \tilde{\mathbf{b}}^2}$, $\alpha_i = \sqrt{1 - \mathbf{b}_i^2}$ and $k_i = \sin\frac{\theta_i}{2}$. We have

$$\begin{aligned}
\mathbf{b}_i * \tilde{\mathbf{b}} &= \alpha_i \tilde{\mathbf{b}} + \tilde{\alpha}\mathbf{b}_i - \mathbf{b}_i \times \tilde{\mathbf{b}} && \text{(B.50)}\\
\frac{\partial}{d\theta_i}\mathbf{b}_i * \tilde{\mathbf{b}} &= \frac{\partial k_i}{d\theta_i}\frac{\partial}{dk_i}\mathbf{b}_i * \tilde{\mathbf{b}} = \frac{\partial k_i}{d\theta_i}(\tilde{\alpha}\mathbf{a}_i - \frac{k_i}{\alpha_i}\tilde{\mathbf{b}} - \alpha_i \times \tilde{\mathbf{b}}) && \text{(B.51)}
\end{aligned}$$

We can now compute

$$
\begin{aligned}
0 &= \frac{\partial}{d\theta_i}\|\mathbf{b}_i * \tilde{\mathbf{b}}\|^2 && \text{(B.52)}\\
&= \frac{\partial k_i}{d\theta_i}(\alpha_i\tilde{\mathbf{b}} + \tilde{\alpha}\mathbf{b}_i - \mathbf{b}_i \times \tilde{\mathbf{b}})^T(\tilde{\alpha}\mathbf{a}_i - \frac{k_i}{\alpha_i}\tilde{\mathbf{b}} - \alpha_i \times \tilde{\mathbf{b}}) && \text{(B.53)}\\
&= \frac{2}{\cos\frac{\theta_i}{2}}\big(k_i\tilde{\alpha}^2 + \tilde{\alpha}\alpha_i\tilde{\mathbf{b}}^T\mathbf{a}_i - k_i^2\frac{\tilde{\alpha}}{\alpha_i}\tilde{\mathbf{b}}^T\mathbf{a}_i - k_i\tilde{\mathbf{b}}^2 + k_i(\mathbf{a}_i \times \tilde{\mathbf{b}})^2\big) && \text{(B.54)}\\
&= \frac{2}{\alpha_i^2}\big(\alpha_i k_i\tilde{\alpha}^2 + \tilde{\alpha}\alpha_i^2\tilde{\mathbf{b}}^T\mathbf{a}_i - k_i^2\tilde{\alpha}\tilde{\mathbf{b}}^T\mathbf{a}_i - \alpha_i k_i\tilde{\mathbf{b}}^2 + \alpha_i k_i(\mathbf{a}_i \times \tilde{\mathbf{b}})^2\big) && \text{(B.55)}\\
&= (2\alpha_i^2 - 1)(\tilde{\alpha}\tilde{\mathbf{b}}^T\mathbf{a}_i) + \alpha_i k_i(\tilde{\alpha}^2 - \tilde{\mathbf{b}}^2 + (\mathbf{a}_i \times \tilde{\mathbf{b}})^2) && \text{(B.56)}
\end{aligned}
$$

Since by (A.2,A.3) $(2\alpha_i^2 - 1) = \cos\theta_i$ and $2\alpha_i k_i = \sin\theta_i$, we have

$$-2\cos\theta_i(\tilde{\alpha}\tilde{\mathbf{b}}^T\mathbf{a}_i) = \sin\theta_i(\tilde{\alpha}^2 - \tilde{\mathbf{b}}^2 + (\mathbf{a}_i \times \tilde{\mathbf{b}})^2), \qquad \text{(B.57)}$$

and hence

$$\tan\theta_i = \frac{-2(\tilde{\alpha}\tilde{\mathbf{b}}^T\mathbf{a}_i)}{\tilde{\alpha}^2 - \tilde{\mathbf{b}}^2 + (\mathbf{a}_i \times \tilde{\mathbf{b}})^2} = \frac{-2(\tilde{\alpha}\tilde{\mathbf{b}}^T\mathbf{a}_i)}{1 - 2\tilde{\mathbf{b}}^2 + (\mathbf{a}_i \times \tilde{\mathbf{b}})^2}. \qquad \text{(B.58)}$$

This yields two solutions for θ_i, one is a maximum and the other a minimum of $\|\bar{\mathbf{b}}\|$. At each iteration, one picks the minimum, i.e. the one yielding a positive second derivative of $\|\bar{\mathbf{b}}\|$. This solves the problem when considering only the end-effector orientation. If both orientation and position are considered, then function to minimize is a weighted sum of the orientation (B.49) and the position (2.19) cases, and the solution is given by (2.22).

Further investigations should be carried out to see whether this version of CCD has any practical advantage over the standard CCD. At first sight, it looks more elegant as it minimizes directly size of the "difference" (in a rotation topology) between the target and the end-effector rotation, whereas the standard version minimized the sum of angles along three arbitrary axes.

Appendix C

Implementation on the iCub humanoid robot

C.1 Introduction

As mentioned before, the reaching and body schema learning algorithms were implemented on the open source iCub robot of the European FP6 Integrated Project RobotCub. The project being open source, the C++ code can be downloaded from the RobotCub website [iCub software repository, 2008]. More detailed executable and class-level documentation is also available on that website. Here we only give a coarse picture of the system to give the reader an idea about how it was implemented.

C.2 Global Structure

Code for the iCub is distributed on so-called modules, which are executable programs that can be run on different machines and that communicate using the Yarp [Fitzpatrick et al., 2008] middleware. The structure of the system is shown in Fig C.1. This is the system used to perform the experiments described in Section 3.4.2. One module implements the body schema learning and takes as input the proprioceptive information as well as the visual information on the end-effector position and the rotational component of the visual flow. The updated versions of the body schema are then constantly sent to two reaching modules, implementing the reaching controller described in Chapter 4, one controlling the arm, and one controlling the head and eyes (for visual tracking, which can be seen as "reaching with the gaze"). In addition, those reaching modules take the target location as input, after it has been converted from a eye-centered FoR to a body-centered FoR. The module that performs this transformation gets regular updates of the body schema. The joint angle trajectories generated by the reaching modules are then realized by the robot using a PID control.

The implementation was done such that those modules can be used with any robot geometry, as long as it can be represented with a tree of rotational links.

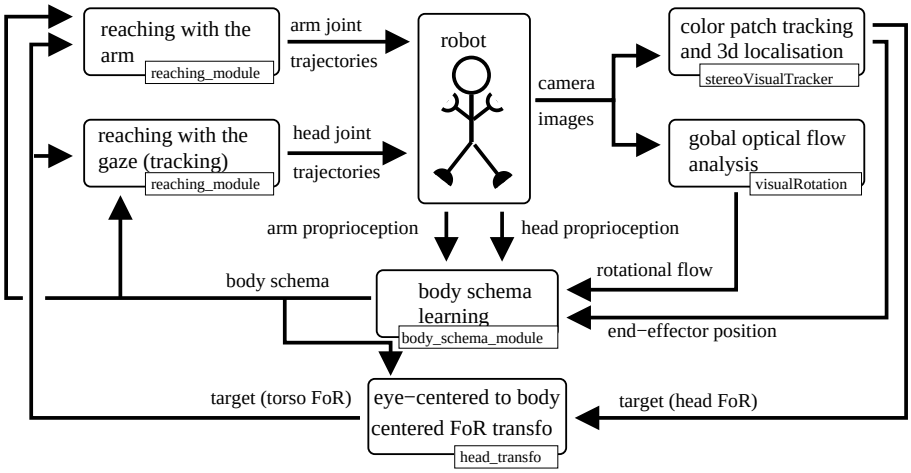


Figure C.1: The module architecture for the iCub implementation

A configuration file describes the robot geometry and the links to be controlled. As an example, Fig C.1 shows that the same executable (the reaching module) can control the head or the arm depending on the parameters given.

C.3 List of Modules

reaching_module

This module performs reaching movements with a serial manipulator of arbitrary geometry and number of degrees of freedom. The target of the reaching movement can be a given position or a given position and orientation. The algorithm used is the dynamical systems - based, hybrid joint angle and end-effector location controller, described in Chapter 4. This module was used to perform reaching with the icub right arm and head tracking with the icub head.

body_schema_learning

This module learns the body schema (that is something similar to the Denavit-Hartenberg parameters) of a humanoid robot. This is performed online, as partly described in Chapter 3. There are two ways of updating the body schema. One is to track a marker on placed on a limb (for example the hand). By knowing the position of the the marker with respect to the eyes, the system can update the entire kinematic chain going from the eyes to the marker. The second way is to analyze the visual flow produced by head movements and update the kinematics of the head accordingly.

head_transfo

This module converts positions from a frame of reference centered on one joint to a frame of reference centered on another joint. It has been used mainly

for converting from eyes-centered to body-centered frames of reference. It can receive updated version of the body schema from the network.

stereoVisualTracker

This module performs color-based 3d tracking with stereovision. Color is selected by selecting a region of that color on the opencv windows. User interface is provided by keyboard commands. Up to 3 objects can be tracked. Calibration is mandatory and done with a checkerboard. The calibration files are respectively params/<dir>/params1 and params/<dir>/params2 for the first and the second camera, where <dir> may be given as argument in the command line

visualRotation

This module takes an image, finds good features to track, and then looks in the next image where those features are. It projects the features in 3d and then computes the rotation that best accounts for the set of features in the first image and their images in the second images. This is done using the algorithm described in [Horn, 1987].

Bibliography

[Abend et al., 1982] Abend, W., Bizzi, E., and Morasso, P. (1982). Human arm trajectory formation. *Brain*, 105:331–348.

[Adamovich et al., 1998] Adamovich, S., Berkinblit, M., Fookson, O., and poizner, H. (1998). Pointing in 3d space to remembered targets. *The Journal of Neurophysiology*, 79(6):2833–2846.

[Admiraal et al., 2004] Admiraal, M., Kusters, M., and Gielen, S. (2004). Modeling kinematics and dynamics of human arm movements. *Motor Control*, 8:312–338.

[Aflalo and Graziano, 2007] Aflalo, T. N. and Graziano, M. S. A. (2007). Relationship between unconstrained arm movements and single-neuron firing in the macaque motor cortex. *Journal of Neuroscience*, 27:2760–2780.

[Albus, 1975] Albus, J. (1975). A new approach to manipulator control: The cerebellar model articulation controller (CMAC). *ASME Journal of Dynamic Systems, Mesurement and Control*, 97:220–227.

[Aleotti and Caselli, 2006] Aleotti, J. and Caselli, S. (2006). Robust trajectory learning and approximation for robot programming by demonstration. *Robotics and Autonomous Systems*, 54(5):409–413.

[Alexander, 1997] Alexander, R. M. N. (1997). A minimum energy cost hypothesis for human arm trajectories. *Biological Cybernetics*, 76:97–105.

[Altmann, 1986] Altmann, S. (1986). *Rotations, Quaternions and Double Groups*. Oxford University Press.

[Altmann, 1989] Altmann, S. (1989). Hamilton, Rodrigues, and the Quaternion Scandal. *Mathematics Magazine*, 62(5):291–308.

[Andry et al., 2001] Andry, P., Gaussier, P., Moga, S., Banquet, J., and Nadel, J. (2001). Learning and communication via imitation: an autonomous robot perspective. *IEEE Transactions on Systems Man and Cybernetics A*, 31(5):431–442.

[Andry. et al., 2002] Andry., P., Gaussier, P., and Nadel, J. (2002). From sensorimotor coordination to low level imitation. In *Second international workshop on epigenetic robotics, EPIROB 02*, pages 7–15.

[Ansky, 1914] Ansky, S. (1992 [1914]). The dybbuk or between two worlds: A dramatic legend in four acts. In *The Dybbuk and Other Writings*. Schoken Books.

[Arbib, 2002] Arbib, M. (2002). Schema theory. In Arbib, M., editor, *The Handbook of Brain Theory and Neural Networks*, pages 993–998. MIT Press, 2 edition.

[Ariff et al., 2002] Ariff, G., Donchin, O., Nanayakkaran, T., and Shadmehr, R. (2002). A real-time state predictor in motor control: Study of saccadic eye movements during unseen reaching movements. *The Journal of Neuroscience*, 22:7721–7729.

[Aristotles, 1978] Aristotles (1978). *De Motu animalium*. Princeton University Press. Edited, translated and commented by M. Craven Nussbaum.

[Atkeson and Hollerbach, 1985] Atkeson, C. and Hollerbach, J. (1985). Kinematic features of unrestrained vertical arm movements. *The Journal of Neuroscience*, 5(9):2318–2330.

[Baillieul, 1985] Baillieul, J. (1985). Kinematic programming alternatives for redundant manipulators. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 722–728.

[Batista et al., 1999] Batista, A., Buneo, C., Snyder, L., and Andersen, R. (1999). Reach plans in eye-centered coordinates. *Science*, 285(5425):257–260.

[Bauchau and Trainelli, 2003] Bauchau, O. and Trainelli, L. (2003). The vectorial parameterization of rotation. *Nonlinear Dynamics*, 32:71–92.

[Bedford, 1999] Bedford, F. (1999). Keeping perception accurate. *Trends in Cognitive Science*, 3(1):4–11.

[Bennett et al., 1991] Bennett, D., Geiger, D., and Hollerbach, J. (1991). Autonomous robot calibration for hand-eye coordination. *International Journal of Robotics Research*, 10(5):550–559.

[Bennett et al., 1992] Bennett, D., Hollerbach, J., and Henri, P. (1992). Kinematic calibration by direct estimation of the jacobian matrix. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 351–357.

[Bernstein, 1967] Bernstein, N. (1967). *The Co-ordination and Regulation of Movements*. Pergamon Press.

[Bernstein, 1996] Bernstein, N. (1996). Dexterity and its development. In Latash, M. and Turvey, M., editors, *On dexterity and its Development*, pages 305–338. Lawrence Erlbaum Associates.

[Berthouze and Lungarella, 2004] Berthouze, L. and Lungarella, M. (2004). Motor skill acquisition under environmental perturbations: On the necessity of alternate freezing and freeing of degrees of freedom. *Adaptive Behavior*, 12(1):47.

[Berti and Frassinetti, 2000] Berti, A. and Frassinetti, F. (2000). When far becomes near: Remapping of space by tool use. *Journal of Cognitive Neuroscience*, 12(3):415–420.

[Bettelheim, 1987] Bettelheim, B. (1987). *A Good Enough Parent*. Knopf.

[Beurze et al., 2006] Beurze, S., Pelt, S. V., and Medendorp, W. (2006). Behavioral reference frames for planning human reaching movements. *Journal of Neurophysiology*.

[Billard, 2000] Billard, A. (2000). Play, Dreams and Imitation in Robota. In *Workshop on Interactive Robotics and Entertainment*.

[Billard et al., 2006] Billard, A., Calinon, S., and Guenter, F. (2006). Discriminative and adaptive imitation in uni-manual and bi-manual tasks. *Robotics and Autonomous Systems*, 54(5):370–384.

[Bizzi et al., 1984] Bizzi, E., Accornero, N., Chapple, W., and Hogan, N. (1984). Posture control and trajectory formation during arm movement. *The Journal of Neuroscience*, 4:2738–2744.

[Boole, 1854] Boole, G. (1854). *An investigation of the laws of thought, on which are founded the mathematical theories of logic and probabilities*. Walton and Maberly.

[Botvinick, 2004] Botvinick, M. (2004). Probing the neural basis of body ownership. *Science*, 305(5685):782 – 783.

[Braun and Wolpert, 2007] Braun, D. and Wolpert, D. (2007). Optimal control: When redundancy matters. *Current Biology*, 17(22):973–975.

[Brooks, 1999] Brooks, R. (1999). *Cambrian Intelligence*. MIT Press.

[Brooks et al., 1999] Brooks, R., Breazeal, C., Marjanovic, M., Scassellati, B., and Williamson, M. (1999). The cog project: Building a humanoid robot. *Lecture Notes in Computer Science*, 1562:52–87.

[Brugger et al., 2000] Brugger, P., Kollias, S., Müri, R., Crelier, G., Hepp-Reymond, M.-C., and Regard, M. (2000). Beyond re-membering: Phantom sensations of congenitally absent limbs. *PNAS*, 97(11):6167–6172.

[Bullock et al., 1999] Bullock, D., Bongers, R., Lankhorst, M., and Beek, P. (1999). A vector-integration-to-endpoint model for performance of viapoint movements. *Neural Networks*, 12(1):1–29.

[Bullock and Grossberg, 1988] Bullock, D. and Grossberg, S. (1988). Neural dynamics of planned arm movements: Emergent invariants and speed-accuracy properties during trajectory formation. *Psychological Review*, 95(1):49–90.

[Bullock et al., 1993] Bullock, D., Grossberg, S., and Mannes, C. (1993). A neural network model for cursive script production. *Biological Cybernetics*, 70(1):15–28.

[Burdick, 1989] Burdick, J. (1989). On the inverse kinematics of redundant manipulators: Characterization of the self-motion manifolds. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 264–270.

[Burnod et al., 1999] Burnod, Y., Baraduc, P., Battaglia-Mayer, A., Guigon, E., Koechlin, E., Ferraina, S., Laquaniti, F., and Caminiti, R. (1999). Parieto-frontal coding of reaching: an integrated framework. *Experimental Brain Research*, 129:325–346.

[Butterworth, 1999] Butterworth, G. (1999). *Neonatal imitation: existence, mechanisms and motives*, pages 63–88. Cambridge University Press.

[Calinon and Billard, 2005] Calinon, S. and Billard, A. (2005). Recognition and reproduction of gestures using a probabilistic framework combining PCA, ICA and HMM. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 105–112.

[Calinon et al., 2007] Calinon, S., Guenter, F., and Billard, A. (2007). On learning, representing and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man and Cybernetics, Part B.*, 37(2):286–298.

[Campbell et al., 2006] Campbell, C., Peters, R., Bodenheimer, R., Bluethmann, W., Huber, E., and Ambrose, R. (2006). Superpositioning of behaviors learned through teleoperation. *IEEE Transactions on Robotics*, pages 79–91.

[Carrozzo and Lacquaniti, 1994] Carrozzo, M. and Lacquaniti, F. (1994). A hybrid frame of reference for visuo-manual coordination. *Neuroreport*, 5:453–456.

[Chalmers, 1995] Chalmers, D. (1995). Facing up to the problem of consciousness. *Journal of Consciousness Studies*, 2(3):200–219.

[Chan and Dubey, 1995] Chan, T. and Dubey, R. (1995). A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators. *IEEE Transactions on Robotics and Automation*, 11(2):286–292.

[Chaumette and Marchand, 2001] Chaumette, F. and Marchand, E. (2001). A redundancy-based iterative approach for avoiding joint limits: Application to visual servoing. *IEEE Transaction on Robotics and Automation*, 17(5):719–730.

[Chiaverini et al., 1994] Chiaverini, S., Siciliano, B., and Egeland, O. (1994). Review of the damped leat-squares inverse kinematics with experiments on an industrial robot manipulator. *IEEE Transactions on Control Systems Technology*, 2(2):123–134.

[Copeland, 1993] Copeland, J. (1993). *Artificial Intelligence. A Philosophical Introduction*. Blackwell Publishers.

[Cordeschi, 2002] Cordeschi, R. (2002). *The discovery of the artificial*. Kluwer Academic Publisher.

[Cordeschi, 2006] Cordeschi, R. (2006). Il metodo sintetico tra scienza cognitiva e robotica. In *3e Convegno Nazionale di Scienze Cognitive*.

[Crevier, 1993] Crevier, D. (1993). *AI: The Tumultuous Search for Artificial Intelligence*. BasicBooks.

[Cruse and Brüwer, 1987] Cruse, H. and Brüwer, M. (1987). The human arm as a redundant manipulator: The control of path and joint angles. *Biological Cybernetics*, 57:137–144.

[D'Avella et al., 2003] D'Avella, A., Saltiel, P., and Bizzi, E. (2003). Combinations of muscle synergies in the construction of a natural motor behavior. *Nature Neuroscience*, 6(3):300–308.

[de Condillac, 1754] de Condillac, E. (1984 [1754]). *Traité des sensations*. Fayard. English translation: Treatise on Sensations.

[Delson and West, 1994] Delson, N. and West, H. (1994). Robot programming by human demonstration: The use of human inconsistency in improving 3d robot trajectories. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*.

[Dennett, 1987] Dennett, D. (1987). *The Intentional Stance*. MIT Press.

[Desmurget et al., 1998a] Desmurget, M., Gréa, H., and Prablanc, C. (1998a). Final posture of the upper limb depends on the initial position of the hand during prehension movements. *Experimental Brain Research*, 119:411–516.

[Desmurget et al., 1997] Desmurget, M., Jordan, M., Prablanc, C., and Jeannerod, M. (1997). Constrained and unconstrained movements involve different control strategies. *Journal of Neurophysiology*, 77:1644–1650.

[Desmurget et al., 1998b] Desmurget, M., Pélisson, D., Rosseti, Y., and Prablanc, C. (1998b). From eye to hand: Planning goal-directed movements. *Neuroscience and Biobehavioural Reviews*, 22(6):761–788.

[Desmurget et al., 1995] Desmurget, M., Prablanc, C., Rossetti, Y., Arzi, M., Paulignan, Y., Urquizar, C., and Mignot, J.-C. (1995). Postural and synergic control for three-dimensional movements of reaching and grasping. *Journal of Neurophysiology*, 74(2):905–910.

[Dillmann, 2004] Dillmann, R. (2004). Teaching and learning of robot tasks via observation of human performance. *Robotics and Autonomous Systems*, 47(2,3):109–116.

[Dreyfus, 1992] Dreyfus, H. (1992). *What Computers Still Can't Do: A Critique of Artificial Reason*. MIT Pres.

[D'Souza et al., 2001] D'Souza, A., Vijayakumar, S., and Schaal, S. (2001). Learning inverse kinematics. In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 298–303.

[Engelbrecht, 2001] Engelbrecht, S. (2001). Minimum principles in motor control. *Journal of Mathematical Psychology*, 45(3):497–452.

[Fadiga, 2006] Fadiga, L. (2006). Personal communication.

[Fadiga et al., 2000] Fadiga, L., Fogassi, L., Gallese, V., and Rizzolatti, G. (2000). Visuomotor neurons: ambiguity of the discharge or 'motor' perception? *International Journal of Psychophysiology*, 35:165–177.

[Feldman, 2006] Feldman, A. (2006). The nature of voluntary control of motor actions. In Latash, M. and Lestienne, F., editors, *Motor Control and Learning*, pages 3–8. Springer.

[Feldman and Latash, 2005] Feldman, A. and Latash, M. (2005). Testing hypotheses and the advancement of science:recent attempts to falsify the equilibrium point hypothesis. *Experimental Brain Research*.

[Feldman and Levin, 1995] Feldman, A. and Levin, M. (1995). The origin and use of positional frames of reference in motor control. *Behavioral Brain Sciences*, 18:723–806.

[Fitts, 1954] Fitts, P. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47:381–391.

[Fitzpatrick et al., 2008] Fitzpatrick, P., Metta, G., and Natale, L. (2008). Towards long-lived robot genes. *Robotics and Autonomous Systems*, 56(1):29–45.

[Fitzpatrick et al., 2003] Fitzpatrick, P., Metta, G., Natale, L., Rao, S., and Sandini, G. (2003). Learning about objects through action - initial steps toward artificial cognition. In *Proceeding of the IEEE International Conference on Robotics and Automation*.

[Flash and Handzel, 2007] Flash, T. and Handzel, A. (2007). Affine differential geometry analysis of human arm movements. *Biological Cybernetics*, 96(6):577–601.

[Flash and Hochner, 2005] Flash, T. and Hochner, B. (2005). Motor primitives in vertebrates and invertebrates. *Current Opinion in Neurobiology*, 15(6):660–666.

[Flash and Hogan, 1985] Flash, T. and Hogan, N. (1985). The coordination of arm movements: An experimentally confirmed mathematical model. *The Journal of Neuroscience*, 5(7):1688–1703.

[Fuke et al., 2007] Fuke, S., Ogino, M., and Asada, M. (2007). Body image constructed from motor and tactile images with visual information. *International Journal of Humanoid Robotics*, 4(2):347–364.

[Galantucci et al., 2006] Galantucci, B., Fowler, C., and Turvey, M. (2006). The motor theory of speech perception reviewed. *Psychonomic Bulletin & Review*, 13(3):361–377.

[Gallagher, 2005a] Gallagher, S. (2005a). Dynamic models of body schematic processes. In Preester, H. D. and Knockaert, V., editors, *Body Image and Body Schema*. John Benjamins Publishing Company.

[Gallagher, 2005b] Gallagher, S. (2005b). *How the body shapes the mind*. Oxford University Press.

[Gandolfo et al., 1996] Gandolfo, F., Mussa-Ivaldi, F., and Bizzi, E. (1996). Motor learning by field approximation. *Proceedings of the National Academy of Science USA*, 93:3843–3846.

[Gaskett and Cheng, 2003] Gaskett, C. and Cheng, G. (2003). Online learning of a motor map for humanoid robot reaching. In *Proceedings of the 2nd International Conference on Computational Intelligence, Robotics and Autonomous Systems*.

[Gatla et al., 2007] Gatla, C., Lumia, R., Wood, J., and Starr, G. (2007). An automated method to calibrate industrial robots using a virtual closed kinematic chain. *IEEE Transactions on Robotics*, 23(6):1105–1116.

[Ghafouri et al., 2002] Ghafouri, M., Archambault, P., Adamovich, S., and Feldman, A. (2002). Pointing movements may be produced in different frames of reference depending on the task demand. *Brain Research*, 929:117–128.

[Ghahramani and Jordan, 1994] Ghahramani, Z. and Jordan, M. (1994). Supervised learning from incomplete data via an em approach. In Cowan, J., Tesauro, G., and J.Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 120–127. Morgan Kaufmann Publishers.

[Ghahramani et al., 1996] Ghahramani, Z., Wolpert, D., and Jordan, M. (1996). Generalization to local remappings if the visuomotor coordinate transformation. *The Journal of Neuroscience*, 16(21):785–796.

[Gibson, 1979] Gibson, J. (1979). *The ecological approach to visual perception*. Houghton Mifflin.

[Giszter et al., 1993] Giszter, S., Mussa-Ivaldi, F., and Bizzi, E. (1993). Convergent force fields organized in the frog's spinal cord. *The Journal of Neuroscience*, 13(2):467–491.

[Graziano and Gross, 1993] Graziano, M. and Gross, C. (1993). A bimodal map of space: somatosensory receptive fields in the macaque putamen with corresponding visual receptive fields. *Experimental Brain Research*, 97:96–109.

[Grimes et al., 2007] Grimes, D., Rashid, D., and Rao, R. (2007). Learning nonparametric models for probabilistic imitation. In *Advances in Neural Information Processing Systems*, volume 19, pages 521–528.

[Guenter et al., 2007] Guenter, F., Hersch, M., Calinon, S., and Billard, A. (2007). Reinforcement learning for imitating constrained reaching movements. *RSJ Advanced Robotics*, 21(13):1521–1544.

[Guillaume, 1937] Guillaume, P. (1979 [1937]). *La psychologie de la forme*. Flammarion, Paris.

[Head and Holmes, 1912] Head, H. and Holmes, G. (1912). Sensory disturbances from cerebral lesions. *Brain*, 34:102–254.

[Hersch and Billard, 2006] Hersch, M. and Billard, A. (2006). A biologically-inspired model of reaching movements. In *Proceedings of the 2006 IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 1067– 1072.

[Hersch and Billard, 2008] Hersch, M. and Billard, A. (2008). Reaching with concurrent dynamical systems. *Autonomous robots*, 25(1-2):71–83.

[Hersch et al., 2006] Hersch, M., Guenter, F., Calinon, S., and Billard, A. (2006). Learning dynamical system modulation for constrained reaching tasks. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 444–449.

[Hersch et al., 2008a] Hersch, M., Guenter, F., Calinon, S., and Billard, A. (2008a). Dynamical system modulation for robot learning via kinesthetic demonstrations. *IEEE Transactions on Robotics*, 24(6):1463–1467.

[Hersch et al., 2008b] Hersch, M., Sauser, E., and Billard, A. (2008b). Online learning of the body schema. *International Journal of Humanoid Robotics*, 5(2):161–181.

[Hestenes, 1999] Hestenes, D. (1999). *New Foudations for Classical Mechanics*, pages 277–305. Fundamental Theories of Physics. Kluwer Academic Publishers, 2 edition.

[Higuchi et al., 2006] Higuchi, T., Imanaka, K., and Patla, A. (2006). Action-oriented representation of peripersonal and extrapersonal space: Insights from manual and locomotor action. *Japanese Psychological Research*, 48(3):126–140.

[Hilbert, 1928] Hilbert, D. (1928). Die grundlagen der mathematik. In *Abhandlungen aus dem Mathematischen Seminar der Hamburgischen Universität*, volume 6. cited by Detlefsen (1986) *Hilbert's Program: An Essay on Mathematical Instrumentalism*, Kluwer.

[Hogan, 1984] Hogan, N. (1984). An organizing principle for a class of voluntary arm movements. *Journal of Neuroscience*, 4:2745–2754.

[Hogan, 1985] Hogan, N. (1985). The mechanics of multi-joint posture and movement control. *Biological Cybernetics*, 52(5):315–331.

[Hogan and Flash, 1987] Hogan, N. and Flash, T. (1987). Moving gracefully: quantitative theories of motor coordination. *Trends in Neurosciences*, 10(4):170–174.

[Holland, 2003] Holland, O. (2003). Exploration and high adventure: the legacy of Grey Walter. *Royal Society of London Philosophical Transactions Series A*, 361:2085–2121.

[Holland, 2007] Holland, O. (2007). A strongly embodied approach to machine consciousness. *Journal of Consciousness Studies*, 14:97–110(14).

[Hollerbach, 1982] Hollerbach, J. (1982). Computers, brains, and the control of movement. *Trends in Neurosciences*, 5:189–192.

[Holmes and Spence, 2004] Holmes, N. and Spence, C. (2004). The body schema and multisensory representation(s) of peripersonal space. *Cognitive Process*, 5:94–105.

[Horn, 1987] Horn, B. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–641.

[iCub software repository, 2008] iCub software repository (2008). http://eris.liralab.it/iCub/dox/html. Look for lasabodyschema. See also http://www.robotcub.org.

[Ijspeert, 2008] Ijspeert, A. (2008). Central pattern generators for locomotion control in animals and robots: a review. *Neural Networks*, 21(4):642–653.

[Ijspeert et al., 2007] Ijspeert, A., Crespi, A., Ryczko, D., and Cabelguen, J.-M. (2007). From swimming to walking with a salamander robot driven by a spinal cord model. *Science*, 315(5817):1416–1420.

[Ijspeert et al., 2002] Ijspeert, A., Nakanishi, J., and Schaal, S. (2002). Learning attractor landscapes for learning motor primitives. In Becker, S., Thrun, S., and Obermayer, K., editors, *Advances in Neural Information Processing Systems 15*, pages 1547–1554.

[Ilg et al., 2004] Ilg, W., Bakir, G., Mezger, J., and Giese, M. (2004). On the representation, learning and transfer of spatio-temporal movement characteristics. *International Journal of Humanoid Robotics*, pages 613–636.

[Imamizu et al., 1998] Imamizu, H., Uno, Y., and Kawato, M. (1998). Adaptive internal model of intrinsic kinematics involved in learning an aiming task. *Journal of experimental psychology. Human perception and performance*, 24:812–829.

[Inamura et al., 2001] Inamura, T., Nakamura, Y., Toshima, I., and Ezaki, H. (2001). Mimesis embodiment and proto-symbol acquisition for humanoids. In *Proceedings if the International Conference on Advanced Intelligent Mechatronics*, pages 159–164.

[Inamura et al., 2004] Inamura, T., Toshima, I., Tanie, H., and Nakamura, Y. (2004). Embodied symbol emergence based on mimesis theory. *International Journal of Robotics Research*, 23(4-5):363–377.

[Iurascu and Park, 2003] Iurascu, C. and Park, F. (2003). Geometric algorithms for kinematic calibration of robots containing closed loops. *ASME Journal of Mechanical Design*, 125(1):23–32.

[Jellema et al., 2004] Jellema, T., Maassen, G., and Perrett, D. I. (2004). Single cell integration and aminate form, motion and location in the superior temporal cortex of the macaque monkey. *Cerebral Cortex*, 14:781–790.

[Jenkins et al., 2007] Jenkins, O., González, G., and Loper, M. (2007). Tracking human motion and actions for interactive robots. In *Proceedings of the Conference on Human-Robot Interaction (HRI07)*, pages 365–372.

[Jiang and Wang, 2003] Jiang, Z. and Wang, S. (2003). A general learning scheme for cmac-based controller. *Neural Processing Letters*, 18(2).

[Jonas, 1979] Jonas, H. (1979). *Das Prinzip Verantwortung*. Suhrkamp Taschenbuch. English translation: *The imperative of responsibility*.

[Jordan and Rumelhart, 1992] Jordan, M. and Rumelhart, D. (1992). Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16:307–354.

[Karniel, 2002] Karniel, A. (2002). Three creatures named 'forward model'. *Neural Networks*, 15:305–307.

[Katić and Vukobratović, 1994] Katić, D. and Vukobratović, M. (1994). Connectionist approaches to the control of manipulation robots at the executive hierarchical level: An overview. *Journal of Intelligent and Robotic Systems*, 10:1–36.

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

[Kawato et al., 1990] Kawato, M., Maeda, Y., Uno, Y., and Suzuki, R. (1990). Trajectory formation of arm movement by cascade neural network model based on minimum torque-change criterion. *Biological Cybernetics*, 62(4):275–288.

[Kelso, 1995] Kelso, J. (1995). *Dynamic Patterns: The Self-Organization of Brain and Behavior*. MIT Press.

[Khatib, 1985] Khatib, O. (1985). Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 500–505.

[Kirkpatrick, 1899] Kirkpatrick, E. (1899). The development of voluntary movement. *The Psychological Review*, 5:275–281.

[Kitazawa et al., 1997] Kitazawa, S., Kimura, T., and Uka, T. (1997). Prism adaptation of reaching movements: Specificity for the velocity of reaching. *The Journal of Neuroscience*, pages 1481–1492.

[Konczak, 2005] Konczak, J. (2005). On the notion of motor primitives in humans and robots. In *Proceedings of the Fifth International Workshop on Epigenetic Robotic: Modeling Cognitive Development in Robotic Systems*, pages 47–53.

[Kozima and Yano, 2001] Kozima, H. and Yano, H. (2001). A robot that learns to communicate with human caregivers. In *Proceedings of the International Workshop on Epigenetic Robotics*.

[Küçük and Bingül, 2004] Küçük, S. and Bingül, Z. (2004). The inverse kinematics solutions of industrial robot manipulators. In *Proceedings of IEEE International Conference on Mechatronics*, pages 274–279.

[Kuhn, 1962] Kuhn, T. (1962). *The structure of scientific revolutions*. Chicago: University of Chicago Press.

[Kuperstein, 1988] Kuperstein, M. (1988). Neural model for adaptive hand-eye coordination for single postures. *Science*, 102(1):148–162.

[Lacquaniti et al., 1986] Lacquaniti, F., Soechting, J., and Terzuolo, S. (1986). Path constraints on point-to-point arm movements in three-dimensional space. *Neuroscience*, 17(2):313–324.

[Lacquaniti et al., 1983] Lacquaniti, F., Terzuolo, C., and Viviani, P. (1983). The law relating the kinematic and figural aspects of drawing movements. *Acta psychologica*, 54(1-3):115–130.

[Làdavas, 2002] Làdavas, E. (2002). Functional and dynamical properties of visual peripersonal space. *Trends in Cognitive Sciences*, 6(1):17–22.

[Latash, 2008] Latash, M. (2008). *Synergy*. Oxford University Press.

[Latash, 1998] Latash, M. (2008 [1998]). *Neurophysiological Basis of Movement*. Human Kinetics, 2 edition.

[Latash and Zatsiorsky, 2001] Latash, M. and Zatsiorsky, V., editors (2001). *Classics Movement Science*. Human Kinetics.

[Latombe, 1991] Latombe, J. (1991). *Robot Motion Planning*. Springer.

[Legrand et al., 2007] Legrand, D., Borzzoli, C., Rossetti, Y., and Farnè, A. (2007). Close to me: Multisensory space representation for action and pre-reflexive consciousness of oneself-in-the-world. *Consciousness and Cognition*, pages 687–699.

[Lemay and Stelmach, 2005] Lemay, M. and Stelmach, G. (2005). Multiple frames of references for pointing to a remembered target. *Experimental Brain Research*.

[Leyton, 2001] Leyton, M. (2001). *A Generative Theory of Shape*. Springer.

[Liégeois, 1977] Liégeois, A. (1977). Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems, Man, and Cybernetics*, 7(12):868–871.

[Lopes and Santos-Victor, 2005] Lopes, M. and Santos-Victor, J. (2005). Visual learning by imitation with motor representations. *IEEE Transactions on Systems, Man and Cybernetics, Part B*.

[Lopes and Santos-Victor, 2007] Lopes, M. and Santos-Victor, J. (2007). A developmental roadmap for learning by imitation in robots. *IEEE Transactions on Systems Man and Cybernetics, Part B*, 37(2):286–298.

[Lozano-Pérez, 1983] Lozano-Pérez, T. (1983). Robot programming. *Proceedings of the IEEE*, 71(7):821–841.

[Lungarella and Metta, 2003] Lungarella, M. and Metta, G. (2003). Beyond gazing pointing and reaching. In *Proceedings Third International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, pages 81–89.

[Lungarella et al., 2003] Lungarella, M., Metta, G., Pfeifer, R., and Sandini, G. (2003). Developmental robotics: a survey. *Connection Science*.

[MacKay, 2003] MacKay, D. (2003). *Information Theory, Inference and Learning Algorithms*. Cambridge University Press.

[Madinier, 1938] Madinier, G. (1967 [1938]). *Conscience et Mouvement. Etude sur la philosophie française de Condillac à Bergson*. Louvain, Nauwelaerts, 2 edition.

[Magescas and Prablanc, 2006] Magescas, F. and Prablanc, C. (2006). A joint-centred model accounts for movement curvature and spatial variability. *Neuroscience Letters*, 403:114–118.

[Malfait et al., 2008] Malfait, N., Henriques, D., and Gribble, P. (2008). Shape distortion produced by isolated mismatch between vision and proprioception. *Journal of Neurophysiology*, 99:231–243.

[Maravita and Iriki, 2004] Maravita, A. and Iriki, A. (2004). Tools for the body (schema). *Trends in Cognitive Sciences*, 8(2).

[Maravita et al., 2002] Maravita, A., Spence, C., Kennett, S., and Driver, J. (2002). Tool-use changes multimodal spatial interactions between vision and touch in normal humans. *Cognition*, 83(2):25–34.

[Maurer et al., 2005] Maurer, A., Hersch, M., and Billard, A. (2005). Extended hopfield network for sequence learning: Application to gesture recognition. In *Artificial Neural Networks: Biological Inspirations ICANN 2005*, Lecture Notes in Computer Science, pages 493–498. Springer.

[Mel, 1990] Mel, B. (1990). *Connectionist robot motion planning: a neurally-inspired approach to visually guided reaching*. Perspectives in Artificial Intelligence. Academic Press.

[Meltzoff and Moore, 1977] Meltzoff, A. and Moore, M. (1977). Imitation of facial and manual gestures by human neonates. *Science*, 198:75–78.

[Meng and Lee, 2007] Meng, Q. and Lee, M. (2007). Automated cross-modal mapping in robotic eye/hand systems using plasitc radial basis function networks. *Connection Science*, 19(1):25–52.

[Merleau-Ponty, 1945] Merleau-Ponty, M. (1945). *Phénoménologie de la perception*. Gallimard. reprinted in 1997.

[Metta et al., 2006] Metta, G., Fitzpatrick, P., and Natale, L. (2006). Yarp: Yet another robot platform. *International Journal of Advanced Robotics Systems*, 3(1).

[Metta et al., 1999] Metta, G., Sandini, G., and Konczak, J. (1999). A developmental approach to visually-guided reaching in artificial systems. *Neural Networks*, 12(10):1413–1427.

[Montesano et al., 2008] Montesano, L., Lopes, M., Bernardino, A., and Santos-Victor, J. (2008). Learning object affordances: From sensory-motor coordination to imitation. *IEEE Transactions on Robotics*, 24(1):15–26.

[Morasso, 1981] Morasso, P. (1981). Spatial control of arm movements. *Experimental Brain Research*, 42:223–227.

[Mussa-Ivaldi and Bizzi, 2000] Mussa-Ivaldi, F. and Bizzi, E. (2000). Motor learning through the combination of primitives. *Philosophical Transactions of the Royal Society B*, 355:1755–1769.

[Nabeshima et al., 2006] Nabeshima, C., Kuniyoshi, Y., and Lungarella, M. (2006). Adaptive body schema for robotic tool-use. *Advanced Robotics*, 20(10):1105–1126.

[Nadel, 1986] Nadel, J. (1986). *Imitation et communication entre jeunes enfants*. PUF.

[Nagel, 1986] Nagel, T. (1986). *The view from nowhere*. Oxford University Press.

[Nakamura and Hanafusa, 1986] Nakamura, Y. and Hanafusa, H. (1986). Inverse kinematics solutions with singularity robustness for robot manipulator control. *ASME Journal of Dynamic Systems, Measurement, and Control*, 108:163–171.

[Nakano et al., 1999] Nakano, E., Imamizu, H., Uno, Y., Gomi, H., Yoshioka, T., and Kawato, M. (1999). Quantitative examinations of internal representations for arm trajectory planning: Minimum commanded torque change model. *Journal of Neurophysiology*, 81:2140–2155.

[Nehaniv and Dautenhahn, 2002] Nehaniv, C. and Dautenhahn, K. (2002). The correspondence problem. In Nehaniv, C. and Dautenhahn, K., editors, *Imitation in Animals and Artifacts*, pages 41–61. MIT Press.

[Nori and Frezza, 2005] Nori, F. and Frezza, R. (2005). A control theory approach to the analysis and synthesis of the experimentally observed motion primitives. *Biological Cybernetics*, 93(5).

[Ogawara et al., 2003] Ogawara, K., Takamatsu, J., Kimura, H., and Ikeuchi, K. (2003). Extraction of essential interactions through multiple observations of human demonstrations. *IEEE Transactions on Industrial Electronics*, 50:667–675.

[O'Regan, 1992] O'Regan, J. (1992). Solving the 'real' mysteries of visual perception: The world as an outside memory. *Canadian Journal of Psychology*, 46:461–488.

[O'Regan and Noë, 2001] O'Regan, J. and Noë, A. (2001). A sensorimotor account of vision and visual consciousness. *Behavioral and Brain Sciences*, 24(5):939–1011.

[Paillard, 1991] Paillard, J., editor (1991). *Brain and Space*. Oxford University Press. Chapters from Arbib, Berthoz and Paillard.

[Paillard, 2005] Paillard, J. (2005). Vectorial versus configural encoding of body space. a neural basis for a distinction betweeen body schema and body image. In Preester, H. D. and Knockaert, V., editors, *Body Image and Body Schema*. John Benjamins Publishing Company.

[Paine et al., 2006] Paine, R., Grossberg, S., and Gemmert, A. V. (2006). A quantitative evaluation of the avitewrite model of handwriting learning. *Human Movement Science*.

[Petreska and Billard, 2009] Petreska, B. and Billard, A. (2009). Movement curvature planning through force field internal models. *Biological Cybernetics*. submitted.

[Philipona et al., 2004] Philipona, D., O'regan, J. K., Nadal, J.-P., m, O. J., and Coenen, D. (2004). Perception of the structure of the physical world using unknown multimodal sensors and effectors. In *Advances in Neural Information Processing Systems 16*, pages 945–952. MIT Press.

[Poincaré, 1902] Poincaré, H. (1968 [1902]). *La Science et l'Hypothèse*. Flammarion.

[Pollick and Sapiro, 1997] Pollick, F. and Sapiro, G. (1997). Constant affine velocity predicts the 1/3 power law of planar motion perception and generation. *Vision Research*, 37(3):347–353.

[Pouget et al., 2002] Pouget, A., Deneve, S., and Duhamel, J. R. (2002). A computational perspective on the neural basis of multisensory spatial representations. *Nature Review Neuroscience*, 3:741–747.

[Price, 2006] Price, E. (2006). A critical review of congenital phantom limb cases and a developmental theory for the basis of body image. *Consciousness and Cognition*, 15(2):310–322.

[Prigogine, 1996] Prigogine, I. (1996). *La fin des certitudes*. Odile Jacob.

[Proffitt, 2006] Proffitt, D. (2006). Embodied Perception and the Economy of Action. *Perspectives on Psychological Science*, 1(2):110–122.

[Prud'homme and Kalaska, 1994] Prud'homme, M. J. and Kalaska, J. F. (1994). Proprioceptive activity in primate primary somatosensory cortex during active arm reaching movements. *Journal of Neurophysiology*, 72(5):2280–2301.

[Reed, 2002] Reed, C. (2002). What is the body schema? In Meltzoff, A. and Prinz, W., editors, *The Imitative Mind*, pages 233–243. Cambridge University Press.

[Rizzolatti et al., 1997] Rizzolatti, G., Fadiga, L., Fogassi, L., and Gallese, V. (1997). The space around us. *Science*, 277(5323):190–191.

[Rossetti, 1998] Rossetti, Y. (1998). Implicit short-lived motor representations of space in brain damaged and healthy subjects. *Conciousness and Cognition*, 7:520–558.

[Sandini et al., 2004] Sandini, G., Metta, G., and Vernon, D. (2004). Robotcub: an open framework for research in embodied cognition. In *Proceedings of the 2004 IEEE/RAS International Conference on Humanoid Robots*, pages 13–32.

[Schaal and Sternad, 2001] Schaal, S. and Sternad, D. (2001). Origins amd violations of the 2/3 power law in rythmic 3d arm movements. *Experimental Brain Research*, 36:60–72.

[Scholz and Schöner, 1999] Scholz, J. and Schöner, G. (1999). The uncontrolled manifold concept: identifying control variables for a functional task. *Experimental Brain Research*, 126:289–306.

[Schwarz, 1978] Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6:461–464.

[Sciavicco et al., 2000] Sciavicco, L., Siciliano, B., and Sciavicco, B. (2000). *Modelling and Control of Robot Manipulators*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Searle, 1980] Searle, J. (1980). Minds, brains and programs. *Behavioral and Brain Sciences*, 3:417–457.

[Searle, 2000] Searle, J. (2000). Consciousness. *Annual Reviews in Neuroscience*, 23(1):557–578.

[Shadmehr and Mussa-Ivaldi, 1994] Shadmehr, R. and Mussa-Ivaldi, F. (1994). Adaptive representation of dynamics during learning of a motor task. *The Journal of Neuroscience*, 14:3208–3224.

[Shadmehr and Wise, 2005] Shadmehr, R. and Wise, S. (2005). *The Computational Neurobiology of Reaching and Pointing*. The MIT Press.

[Sheets-Johnstone, 2005] Sheets-Johnstone, M. (2005). What are we naming? In Preester, H. D. and Knockaert, V., editors, *Body Image and Body Schema*. John Benjamins Publishing Company.

[Soechting et al., 1995] Soechting, J., Buneo, C., and Flanders, M. (1995). Moving effortlessly in three dimensions: Does Donder's law apply to arm movement? *Journal of Neuroscience*, 15(9):6271–6280.

[Soechting and Lacquaniti, 1981] Soechting, J. and Lacquaniti, F. (1981). Invariant characteristics of a pointing movement in man. *The Journal of Neuroscience*, 1(7):710–720.

[Steels, 1998] Steels, L. (1998). The origins of syntax in visually grounded robotic agents. *Artificial Intelligence*, 103(1-2):133–156.

[Strogatz, 1994] Strogatz, S. (1994). *Nonlinear Dynamics and Chaos*. Westview Press.

[Stuelpnagel, 1964] Stuelpnagel, J. (1964). On the parameterization of the three-dimensional rotation group. *SIAM Review*, 6(4):422–430.

[Sturm et al., 2008] Sturm, J., Plagemann, C., and Burgard, W. (2008). Unsupervised body scheme learning through self-perception. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 3328–3333.

[Tillery et al., 1996] Tillery, S. I., Soechting, J. F., and Ebner, T. J. (1996). Somatosensory cortical activity in relation to arm posture: nonuniform spatial tuning. *Journal of Neurophysiology*, 76:2423–2438.

[Todorov, 2004] Todorov, E. (2004). Optimality principles in sensorimotor control. *Nature Neuroscience*, 7:907–915.

[Todorov and Jordan, 2002] Todorov, E. and Jordan, M. (2002). Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, 5(11):1226–1235.

[Tolani et al., 2000] Tolani, D., Goswami, A., and Badler, N. I. (2000). Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical models*, 62(5):353–388.

[Tsagarakis et al., 2007] Tsagarakis, N., Metta, G., Sandini, G., Breira, D. V. R., Becchi, F., Righetti, L., Santos-Victor, J., Ijspeert, A., Carroza, M., and Caldwell, D. (2007). icub: the design and realization of an open humanoid platform for cognitive and neuroscience research. *Advanced Robotics*, 21(19):1151–1175.

[Turvey, 2007] Turvey, M. (2007). Action and perception at the level of synergies. *Human Movement Science*, 26:657–697.

[Ulloa and Bullock, 2003] Ulloa, A. and Bullock, D. (2003). A neural network simulating human reach–grasp coordination by continuous updating of vector positioning commands. *Neural Networks*, 16(8):1141–1160.

[van Ingen Schenau and van Soest, 1996] van Ingen Schenau, G. and van Soest, A. (1996). On the biomechanical basis of dexterity. In Latash, M. and Turvey, M., editors, *On dexterity and its Development*, pages 305–338. Lawrence Erlbaum Associates.

[Varela, 1989] Varela, F. (1989). *Autonomie et Connaissance*. Seuil. Translated in French and extended from the English: *Principles of Biological Autonomy*, 1980, Elsevier.

[Varela et al., 1991] Varela, F., Thomson, E., and Rosch, E. (1991). *The embodied mind: Cognitive science and human experience*. MIT Press.

[Vernon et al., 2007] Vernon, D., Metta, G., and Sandini, G. (2007). A survey of artificial cognitive systems: implications for the autonomous development of mental capabilities in computational agents. *IEEE Transactions on Evolutionary Computation*, 11(2):151–180.

[Vetter et al., 1999] Vetter, P., Goodbody, S., and Wolpert, D. (1999). Evidence for an eye-centered spherical representation of the visuomotor map. *The Journal of Neurophysiology*, 81(2):935–939.

[Vilaplana and Coronado, 2006] Vilaplana, J. and Coronado, J. (2006). A neural network model for coordination of hand gesture during reach to grasp. *Neural Networks*, 19(1):12–30.

[Wallon, 1954] Wallon, H. (1954). Kinesthésie et image visuelle du corps propre chez l'enfant. *Bulletin de Psychologie*, VII(5). Reprinted in Enfance,no 7,1985.

[Wampler, 1986] Wampler, C. (1986). Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Transactions on Systems, Man and Cybernetics, Part C*, 16(1):93–101.

[Wampler et al., 1995] Wampler, C., Hollerbach, J., and Arai, T. (1995). An implicit loop method for kinematic calibration and its application to closed-chain mechanisms. *IEEE Transactions on Robotics and Automation*, 11(5):710–724.

[Wang et al., 2006] Wang, J., Fleet, D., and Hertzmann, A. (2006). Gaussian process dynamical models. In *Advances in Neural Information Processing Systems*, volume 18, pages 1441–1448.

[Wang and Chen, 1991] Wang, L.-C. and Chen, C. (1991). A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. *IEEE Transactions on Robotics and Automation*, 7(4):489–499.

[Webb, 2002] Webb, B. (2002). Robots in invertebrate neuroscience. *Nature*, 417(6886):359–363.

[Weizenbaum, 1976] Weizenbaum, J. (1976). *Computer Power and Human Reason: From Judgment To Calculation*. W.H. Freeman.

[Welch, 1986] Welch, R. (1986). Adaptation of space perception. In Boff, K., Kaufman, L., and Thomas, J., editors, *Handbook of perception and human performance*, volume 1, pages 24:1–24:45.

[Weng, 1998] Weng, J. (1998). The developmental approach to intelligent robots. In *AAAI Spring Symposium Series, Integrating Robotic Research: Taking The Next Leap*.

[Weng et al., 2001] Weng, J., McClelland, J., Pentland, A., Sporns, O., Stockman, I., Sur, M., and Thelen, E. (2001). Artificial intelligence: Autonomous mental development by robots and animals. *Science*, 291(5504):599–600.

[Whitney, 1969] Whitney, D. (1969). Resolved motion rate control of manipulators and human prosteses. *IEEE Transactions on Man-Machine Systems*, 10(2):47–52.

[Whitney, 1972] Whitney, D. (1972). The mathematics of coordinated control of prosthetic arms and manipulators. *ASME Journal of Dynamic Systems, Measurement, and Control*, 94:303–309.

[Witt et al., 2005] Witt, J., Proffitt, D., and Epstein, W. (2005). Tool use affects percieved distance, but only when you intend to use it. *Journal of Experimental Psychology*, 31(5):880–888.

[Wolpert et al., 1995] Wolpert, D., Ghahramani, Z., and Jordan, M. (1995). An internal model for sensorimotor integration. *Science*, 269(5232):1880–1882.

[Woodworth, 1899] Woodworth, R. (1899). The accuracy of voluntary movement. *The Psychological Review*. Partially reprinted in [Latash and Zatsiorsky, 2001].

[Zwillinger, 2003] Zwillinger, D., editor (2003). *CRC Standard Mathematical Tables and Formulae*. Chapman and Hall/CRC, 31 edition.

Thesis Presentation

Adaptive sensorimotor peripersonal space representation and motor learning for a humanoid robot

PhD candidate: Micha Hersch
Advisor: prof. Aude Billard
Learning Algorithms and Systems Lab

Outline

- Motivation and background
- Body schema learning
- A controller for reaching movements
- Learning motor primitives
- Conclusion

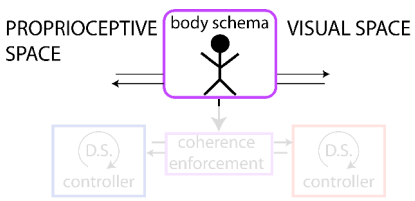
Goal

General goal	Develop basic humanoid upper-body control mechanisms suitable autonomous robot operation.
Emphasis	Adaptivity and robustness to changes in the environment and in the robot hardware.
Inspiration	Current theories on human motor control which displays high adaptivity and robustness.
Hope	Possibly, this synthetic approach can generate insights that can be fed back into our understanding of human motor control.

Main hypotheses

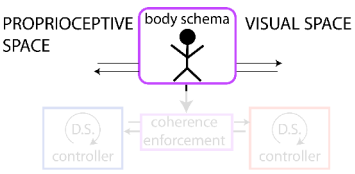
Ecological perspective	Humans perception and action cannot be studied separately from each other.
Multi-modality	Human perception and control are multi-modal. "Representations" are concomittantly expressed in different modalities, with a mapping ensuring coherence across modalities.
Dynamical system control	Movements are not pre-programmed but unfold with the continuous flow of (possibly predicted) sensory information.

From hypothesis to architecture



Only visual and proprioceptive modalities are considered.

Part I: the body schema



Main questions

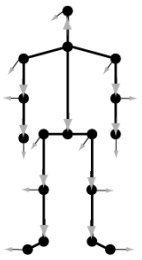
- *representation*	How to represent the mapping between visual and proprioceptive data.
- *acquisition*	How can this mapping emerge as a result of the sensory data flow.
Hypothesis	In humans, this is achieved by the body schema.

A model of the body schema

Body schema:
a tree of rigid-body transformations mirroring the limb structure of the robot.

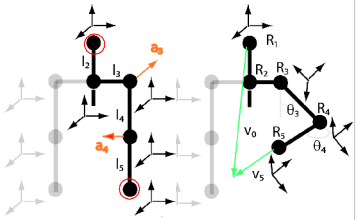
Rotations around a given axis $\mathbf{a}_i$*:*
represent the joints of the robot (tree nodes).

Translations $\mathbf{l}_i$*:*
represent the position of the joint with respect to its proximal joint (its parent) in the tree (tree edges).



Frame of reference transformations

- On can attach a frame of reference to each joint.
- Knowing the joint angular position, one can compute the FoR transformation transforming the FoRs attached to any two joints into one another (forward kinematics).
- This is done by extracting the kinematic chain joining the two joints.

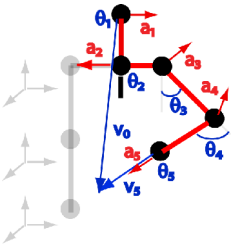


$$\mathbf{v}_0 = \mathbf{v}'_n = \mathcal{T}(\mathbf{v}_n) = \mathbf{l}_1 + \mathbf{R}_1(\mathbf{l}_2 + \mathbf{R}_2(\ldots(\mathbf{l}_n + \mathbf{R}_n(\mathbf{v}_n))\ldots))$$

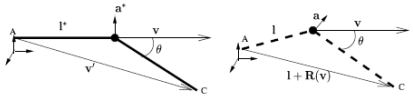
Static body schema learning

Observed Many examples of:
- the position of the end-effector in the FoR of the eyes
- the position of the end-effector in the FoR of the end-effector
-- its angular configuration

Inferred: All the rotation axes and translation vectors on the path from eyes to end-effector



Static learning – single joint example



Real (unknown) transform Estimated transform

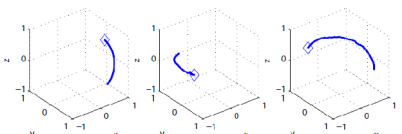
Problem Given an estimated transform parametrized by **a** and **l**, how to find **a*** and **l*** knowing many examples of θ , **v**, and **v'?**

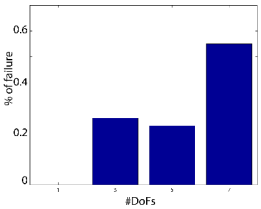
Solution Perform a simple gradient descent on the discrepancy between the observed and predicted transform of **v.**

$$\Delta \mathbf{a} = -\epsilon \frac{\partial}{\partial \mathbf{a}} \frac{1}{2} \left\| \mathbf{v}' - (\quad \mathcal{T}(\mathbf{v})) \right\|^2$$

$$\Delta \mathbf{l} = -\epsilon \frac{\partial}{\partial \mathbf{l}} \frac{1}{2} \left\| \mathbf{v}' - (\quad \mathcal{T}(\mathbf{v})) \right\|^2,$$

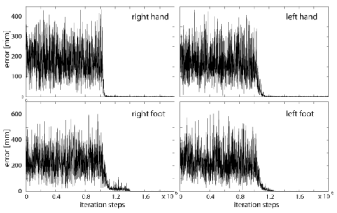
Convergence



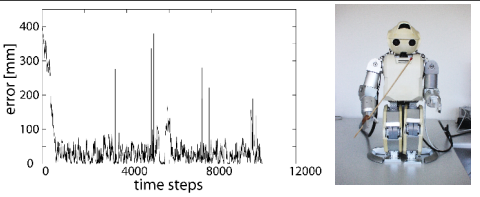


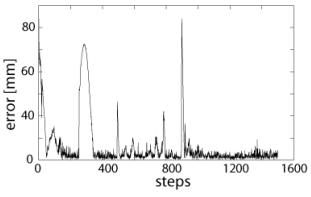
Experience - simulations

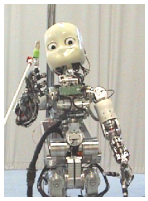
- Body schema learning – the robot "tracks" is hands and feet while randomly exploring its configuration space.



Result: static learning in real setting



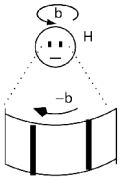




Rotation-based learning

Observed: Many examples of:

- an initial angular head-eye configuration
- a small angular head-eye displacement
- -- the resulting perceived small visual rotation of the world

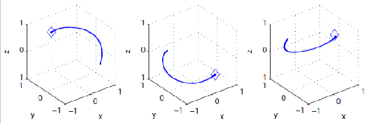


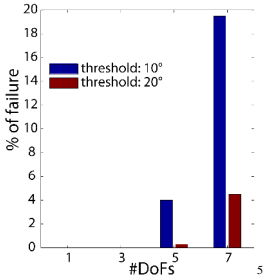
Inferred: All the rotation axes of the head and eyes.

Method: Gradient descent on the discrepancy between the observed and predicted visual rotation of the world $\Delta a_i = -\epsilon \frac{\partial}{\partial a_i} \|b' - J_R(\theta)\dot{\theta}\|^2$

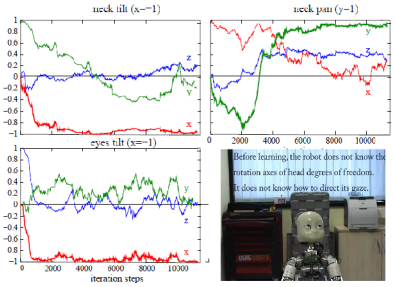
b': observed rotation of the world $J_R(\theta)$: orientation component of the Jacobian

Convergence

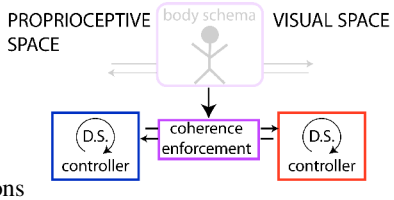




Results: rotation-based learning



Part II: controlling reaching movements



Main questions

- *control* How to implement a multi-modal controller using a dynamical system approach.
- *usefulness* What are the advantages of such a multi-modal dynamical system approach.

Hypothesis Redundancy is part of the solution, not the problem.

The control problem

The question From an initial arm configuration, how to reach a final point with the end-effector.

Sources of redundancy (where a « choice » has to be made)
- final arm configuration
- arm trajectory
- control variables

Possible approaches Optimize some global cost function (jerk, torque, energy).
Let a dynamical system choose (on a local basis).

Dynamical system motor control

- No explicit trajectory planning
- Movements are controlled through dynamical systems active on goal-relevant variables [Kelso 95] .

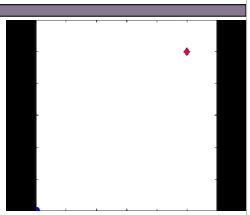
  $\dot{s} = f(s, i)$ (and not $s = f(t, i)$)
- This provides an implicit set of trajectories.
- Support for a dynamical system motor control in animals has been provided by neuro-biological studies [Bizzi et al. 91].
- This allows for higher robustness and adaptability.

The VITE dynamical system

Equation (simplified form)

$\ddot{\mathbf{r}} = \alpha(-\dot{\mathbf{r}} + \beta(\mathbf{r_T} - \mathbf{r}))$

$\alpha, \beta \in [0\ 1]$ $\mathbf{r_T}$: target



Properties

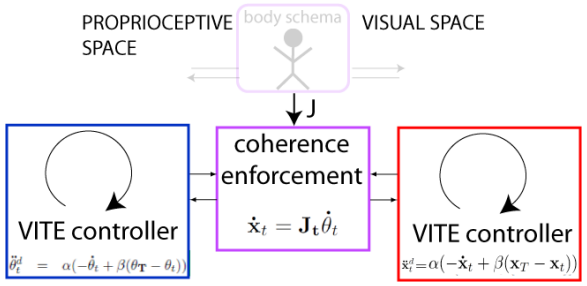
Straight-line trajectories

Bell-shaped velocity profile

Smooth convergence to the target

Put forth by [Bullock & Grossberg, 1988] and argued to underlie human reaching movements.

Overview



Coherence enforcement

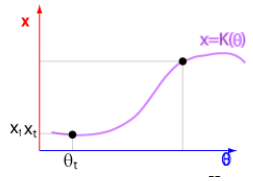
Principle: Find the position that is closest to the desired positions specified by each controller, while remaining consistent.

Modulation: Diagonal weight matrices W_x and W_θ modulate the influence of each controller.

$$\min_{\dot{\theta}_t, \dot{\mathbf{x}}_t} \frac{1}{2}((\dot{\theta}_t - \dot{\theta}_t^d)^T \mathbf{W}_\theta (\dot{\theta}_t - \dot{\theta}_t^d) + (\dot{\mathbf{x}}_t - \dot{\mathbf{x}}_t^d)^T \mathbf{W}_\mathbf{x} (\dot{\mathbf{x}}_t - \dot{\mathbf{x}}_t^d))$$

$$\text{u.c.} \quad \dot{\mathbf{x}}_t = \mathbf{J_t}\dot{\theta}_t,$$

$$\dot{\theta}_t = (\mathbf{W}_\theta + \mathbf{J_t}^T \mathbf{W}_\mathbf{x} \mathbf{J_t})^{-1} (\mathbf{J_t}^T \dot{\mathbf{x}}_t^d + \dot{\theta}_t^d)$$



Weight modulation

- Weight modulation can be used to smoothly switch from one sub-controller to the other.
- This way one can take advantage of the intrinsic properties of each sub-controller.

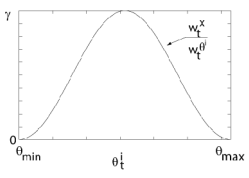
Joint angle controller

No singularity

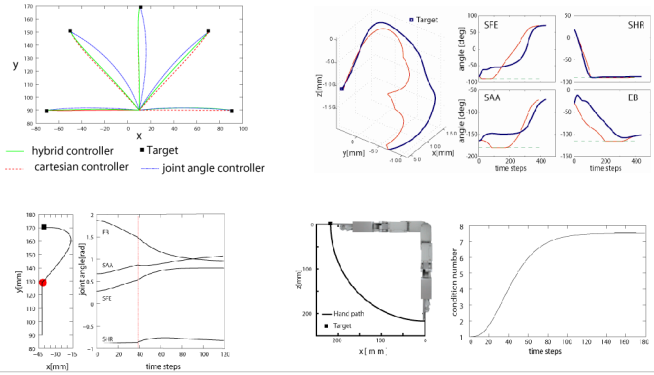
Convex workspace (avoids joint limits)

Cartesian controller

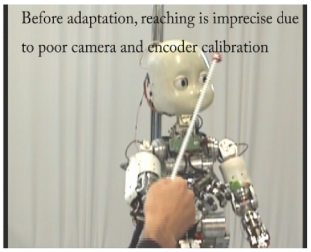
Straight hand paths



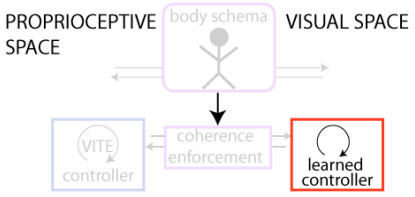
Results



Movies



Part III: learning motor primitives



Main question: How can this framework be extended to enable the robot to learn more sophisticated gestures.

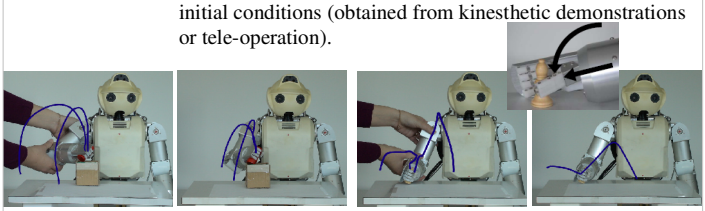
Idea: Replace the Cartesian VITE controller with a learned controller.

Setup

Tasks considered Simple goal directed motions, for which straight line reaching does not work.

Goal The robot should be able to learn to accomplish a task despite different initial conditions and perturbations.

Available data Examples of successful joint angle trajectories for various initial conditions (obtained from kinesthetic demonstrations or tele-operation).



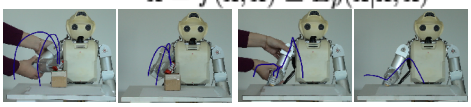
Dynamical system learning

- Gaussian Mixture Regression (GMR):
- Assume the following dynamical system: $\ddot{\mathbf{x}} = f(\mathbf{x}, \dot{\mathbf{x}})$
- Model the joint probability distribution of all variables with a Gaussian Mixture Model:

$$p(\ddot{x}, \dot{x}, x) = \sum_k \pi_k N(\mu_k, \Sigma_k)$$

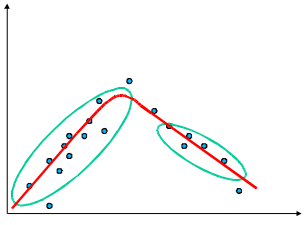
  - p can be learned using the demonstrations (E-M)
- The function f can be approximated as the conditional expectation of $\ddot{\mathbf{x}}$

$$\ddot{\mathbf{x}} = f(\mathbf{x}, \dot{\mathbf{x}}) \simeq E_p(\ddot{\mathbf{x}} | \dot{\mathbf{x}}, \mathbf{x})$$



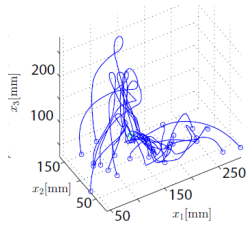
GMR – toy example

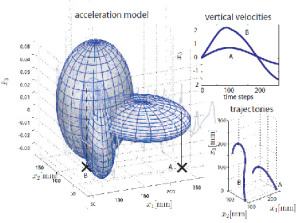
Learning $y = f(x)$

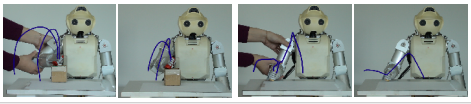


1. Model p(x,y) as a GMM trained on the data
2. f(x) is given by $E_p(y|x)$

Dynamical system learning –chess task

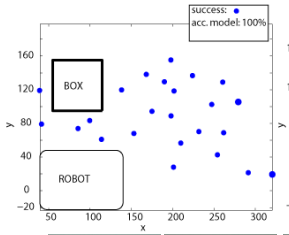




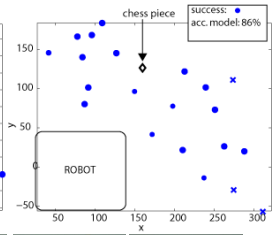


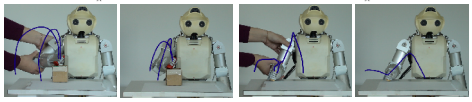
Results

Box task

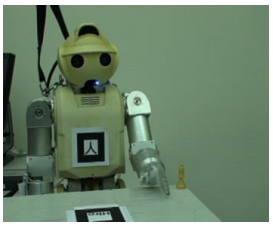


Chess task





Movies



Summary – contribution to robotics

Conceptual An investigation of the use of dynamical systemes for motor control learning, from low-level sensorimotor contingencies to simple task learning, with emphasis on adaptivity - at the sensori-motor level, at the control level, at the task level.

Algorithmic Algorithms for learning to control a humanoid upper body using three concurrent and interconnected dynamical systems - in the space of body schemata, in the proprioceptive space, in the visual end-effector position space.

Software It has been implemented on two robots (in a platform independent way) and the code for controlling the robot and learning the body schema is freely available on the web (through the RobotCub project).

Cognition

- The body schema is mainly about sensori-motor relationships. It is as much related to sensor properties as to objective body geometry.
- There is no need for an “objective” space representation for control. One can have many modality-specific representation of space and ways to relate them to each other.
- There is no need for a centralized control, it is possible to control simultaneously in different frames of reference and balance between them.
- Dynamical systems and attractor dynamics are powerful tools to model this kind of decentralized control.

Micha HERSCH

Msc in Computer Science and in Cognitive Science, PhD in robotics

Chemin des Fleurettes 32, CH – 1007 Lausanne
Phone : 021.311.25.90 (private) 021.693.54.63 (professional)
e-mail: micha.hersch@epfl.ch
http://lasa.epfl.ch/people/member.php?SCIPER=114245

Born May 6th, 1979
Swiss citizenship
Married since 2004
One child, born in 2006

Academic Degrees

2008	PhD in robotics for a thesis entitled « Adaptive peripersonal space representation and motor learning for a humanoid robot »	LASA – EPFL, Lausanne Advisor: Prof. Aude Billard
2004	M.Sc in cognitive science	Université P. et M. Curie, Paris
2003	M.Sc in computer science	EPFL, Lausanne
1998	One Year Program in Humanities	Hebrew University, Jerusalem
1997	Scientific “baccalauréat” (high school degree)	Gymnase du Bugnon, Lausanne

Awards and Scholarships

2008	Full grant for acting as EPFL student representative at the World Knowledge Dialogue
2003	Award for second-best EPFL computer science graduate
2002	Full scholarship for a six-month internship at the International Computer Science Institute, Berkeley, USA
2000	Erasmus scholarship for a one-year exchange at the Technische Universität Berlin
1997	Special prize in mathematics and physics at the “baccalauréat” (high school degree)

Skills

- *Science:* dynamical system theory, statistical learning theory, robot control, artificial neural networks, computational motor control, signal processing, cognitive science, artificial intelligence
- *Academics*: Scientific communication (papers and presentations), reporting for funding agencies, peer reviewing, student supervision, university-level teaching
- *Programming:* C/C++, matlab, java, perl, html, php, SQL
- *Languages:* Fluent: French (mother tongue), English, German, Hebrew
  Colloquial: Spanish

Scientific Publications

Peer-reviewed International Journals

- M. Hersch, F. Guenter, S. Calinon and A. Billard (2008), "Dynamical System Modulation for Robot Adaptive Learning via Kinesthetic Demonstrations", *IEEE Transactions on Robotics*, 24(6),1463-1467.
- M. Hersch and A. Billard (2008), "Online learning of the body schema", *International Journal of Humanoid Robotics*, 5, 161-181.
- M. Hersch and A. Billard (2008), "Reaching with multi-referential dynamical systems", *Autonomous Robots*, 25, 71-83.
- F. Guenter, M. Hersch, S. Calinon, and A. Billard, (2007) "Reinforcement learning for imitating constrained reaching movements", *RSJ Advanced Robotics*, 21(13), 1521-1544.

Peer-reviewed International Conference Proceedings

- M. Hersch, T. Reichert, and A. Billard.(2008) "Iterative rigid body transformation estimation for visual 3-D object tracking", in *Proceedings of the Third International Conference on Computer Vision Theory and Applications*, pp. 674-677.
- M. Hersch, F. Guenter, S. Calinon and A. Billard, (2006) "Learning Dynamical System Modulation for Constrained Reaching Tasks", in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots,* pp. 444-449.
- M. Hersch and A. Billard (2006) "A Biologically-Inspired Model of Reaching Movements", in *Proceedings of the 2006 IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, pp. 1067-1072.
- M. Hersch and A. Billard, (2006) "A Model for Imitating Human Reaching Movements", in *Proceedings of the ACM Human Robot Interaction Conference*, pp. 341-342.
- A. Maurer, M. Hersch and A. Billard, (2005) "Extended Hopfield network for sequence learning: application to gesture recognition", in *Proceedings of the 2005 International Conference on Artificial Neural Networks,* Lecture Notes in Computer science, pp. 493-498.

Reviewing

I provided reviews for the following international journals and conferences :

- IEEE Transactions on Robotics
- IEEE Transactions on Autonomous Mental Development
- Autonomous Robots
- RSJ Advanced Robotics,
- Neural Networks
- IEEE International Conference on Intelligent Robots and Systems (IROS)
- IEEE International Conference on Robotics and Automation (ICRA)
- IEEE International Conference on Robotics and Biomimetics (Robio)
- ACM Human Robot Interaction Conference (HRI)

Hobbies

Humanities (modern European history, philosophy), basketball