UNIVERSITY OF GENOVA

Department of Communication Computer and Systems Science
LIRA-Lab Laboratory for Integrated Advanced Robotics

Toward Predictive Robotics: The Role of Vision and Prediction on the Development of Active Systems

Carlos Beltrán-González

Doctoral Degree Thesis
Electronics and Informatics, XVI^o Course

Thesis Supervisor: Prof. Giulio Sandini

Acknowledgments

I want to thank my advisor, Prof. Giulio Sandini, for his invaluable support during these years. Besides the gratitude for his guidance, I would like to express my admiration for his eternal enthusiasm for research and his capacity to transmit it to those who, like me, have been privileged to work with him. Special thanks go to Giorgio Metta for his help and patience, and particularly, for his friendship during these years. Thanks also to Lorenzo Natale who after a year has been able to satisfy his desire of *revenge* reviewing this thesis. I have to admit that he has done an excellent work, much better than the review I did for his thesis.

I am eternally in debt with my wife Cristina for her infinite support and constant help. The most difficult part of this thesis has been to spend uncountable hours in front of the computer and not in her company.

Finally, I would like to thank all my colleagues at the LIRA-Lab for their friendship and the professional and stimulating environment in which my thesis evolved.

Abstract

The research presented in this thesis is the result of a study on prediction and active vision. These two issues are approached from an interdisciplinary perspective ranging from brain sciences to developmental robotics. The key questions I try to answer are:

1. How does prediction affect the development of active artificial systems?
2. What are the biological and computational basis of prediction?
3. What is the role of prediction and expectation in perceptual processes?

In order to respond to these questions I have tried to understand what prediction is in its biological origin and how it may affect the current behavior and development of an active robot. In this respect, I have found that the neurological basis of prediction is not yet well understood, even though there is an important body of evidence suggesting that prediction plays a fundamental role in many processes, such as development, learning, behavior, motor control, perception, expectations, crossmodal perception and many others. The other important issue I have approached is that prediction is inseparable from the *active* nature of living systems. In this respect, I have built an upper torso humanoid robot and I have studied prediction in the context of motor control and active vision systems. The main contribution this thesis purports to make involves: (1) the understanding of prediction as fundamental to the development of an agent, (2) the study of prediction and expectation in perceptual processes, and (3) the use of prediction in the control of complex robotic systems.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

On Saturday morning you receive a phone call. It is a friend who invites you to play tennis later that morning. You accept, so you get up from bed and take a shower. In the shower you regulate accurately the temperature of the water; it is long ago that you have learned how to set it to get the right temperature. After the shower, you have breakfast taking care not to eat too much as you want to keep your stomach light in order to avoid any trouble during the match. Then you look out of the window and see that it is a cloudy and windy day so you decide to wear something warm. Of course, you put an umbrella in your sports bag just in case it rains. Then you get into your car and drive for half an hour to the sports club where you are meeting your friend. As you drive you stop at traffic lights, follow more or less closely the traffic signs' recommendations, and avoid accidents. You feel relief because you have just avoided a collision with a distracted driver who did not respect a stop sign. Once at the club, you meet your friend and then you play. You spend two hours hitting a small yellow ball (that flies at a certain speed) and trying to send it back into a rectangular field placing it where your friend cannot reach it. You win, of course, and secretly you thank the tennis lessons you have been taking over the last months. Then you take another shower, pay, drive back home, avoid other accidents ...

The quantity and complexity of actions that a human being can do is impressive. We have a very limited idea of how humans deal with all the complexity associated with the actions described in the little story described above. Yet, a high percentage of individuals can resolve such complexity in a *natural* way, without even thinking about it. While we are far from

understanding completely the neural mechanisms guiding you through your Saturday morning we propose the following hypothesis: you are using prediction!

If there is something that unites your activities that morning it is the fact that you have used prediction in performing each and every one of them. Why may this be so? Let us imagine that you wake up after the phone call and you have lost your predictive capabilities (maybe you hit your head during the night). In this case, things become really complex. To begin with, you will not know how your arm will look like after a voluntary movement. Indeed, in a normal situation, you are constantly predicting the *sensory consequences of your movements*. But anyway, let us imagine you manage somehow to go ahead, survive the shower and now you are in front of your coffee cup. Well, surely your brain will not anticipate the grip force necessary to pick up the cup filled with coffee and the cup will slip from your hand; indeed, probably you will never be able to drink your coffee. Anyhow you go ahead, get dressed (of course, you put on light clothing because you cannot anticipate that it is a cold day based in your visual information), and manage to get into your car. Now, recall, you have to drive for half an hour. The first problem arises from not being able to anticipate the behavior of your car. You cannot use the internal *dynamic model* of your car that you have learned patiently in the course of months (probably without even noticing it). Problems keep coming up, but let us suppose you somehow manage to arrive to the crossroads where a distracted driver does not respect a stop sing. Though he was distracted, he immediately sees you and he honks. In a normal situation the honk will make you expect a car coming, so you will immediately start to break (without looking at all); but in this case, you have lost your predictive ability and hence you form no sensorial or causal expectation based on the car horn. May be, by chance, you see the car coming, but you cannot predict its trajectory and consequently you do not know when to brake or steer the car in order to avoid the collision.

I hope these examples suffice to agree on one point: prediction *is* important. We use it constantly in our daily lives and indeed it is central both to our individual survival and to our success as a species. Therefore, I am interested in understanding how this complex capability is used by humans and how this knowledge can be used to build better robots. Figuring out how prediction works in biological systems can help us create better robotic systems, i.e. more helpful for humans, as well as understand how the human body works in general.

In this thesis I address three questions:

1. How does prediction affect the development of an active artificial system?

2. What are the biological and computational basis of prediction?

3. What is the role of prediction and expectation in perceptual processes?

With question (1) I try to understand the factors involved in development and how they are affected by prediction. I specifically discuss learning, perception and movement. Question (2) inquires into the biological and computational structures used for prediction. I describe the parts of the brain involved in prediction —particularly the cerebellum— and the main neural groups participating in prediction mechanisms. Subsequently, I study the principal computational paradigms related to prediction, particularly the models of the cerebellum. Traditionally, these models have been applied to motor control, so the study of prediction in this area acquires a particular relevance in this thesis. Question (3) addresses the interesting issue of the role of prediction and expectation in the processes involved in perception. In this respect, I present three experiments. In the first one, I address the particular relevance of prediction in humans for the task of tracking visual targets (i.e. smooth pursuit) and I present a practical implementation of a control strategy for an active robotic head based on predictive algorithms. In the second experiment, I confront the problem of crossmodal expectations. This is how an artificial system can learn and exploit audio-visual associations to create visual expectations. Finally, in the third experiment, I present a research that combines previous knowledge of a given environment (in the form of a CAD memory) with active vision for controlling a complex mobile robot.

But, what is prediction? And what is active vision? In the following two sections, I try to introduce both concepts. After that I present the outline of this thesis.

1.1 Prediction

Prediction is a statement made about the future. Humans can create predictions in both space and time. A typical example is the act of predicting a trajectory such in the case of a flying object. However, humans can do other type of predictions. For example, we can predict who is in another room hearing his voice coming through the door, or we can predict how a flying ball will bounce off the ground because we have previous experience about that fact. In other words, we need to take into account the influence of learning and cognition in the generation of predictions.

Yet, let's us consider another interesting point. How the knowledge of the future can affect the system in the present? When a system—biological or artificial—has information about the future, this knowledge can be exploited in the present for many things: (i) determine current behavior (Butz et al., 2003), (ii) determine current and future processing of sensory information, (iii) improve perception, (iv) affect development, (v) affect learning. This effect of prediction in the current state of the system is one of the main issues

discussed in this thesis, particularly, the improvement of perception through expectation mechanisms.

It is also necessary to clarify the meaning given to some terms in this thesis that in common language are used as synonyms of prediction. *Anticipation* refers to taking prior actions on the basis of information about the future. These actions can be directed to avoid or propitiate a particular future. The word *expectation* refers to the prediction of an event considered to be probable or certain in the future. This word is widely used in this thesis to refer to *sensorial* expectations.

1.2 Active Vision

Active vision is the *science* dedicated to the construction and use of robotic heads (Blake and Yuille, 1992). Usually these systems try to imitate and simulate the human vision system by physically implementing its dimensions and degrees of freedom (DoF). The more interesting aspect is the use of stereovision in an active manner; this means that the system is able to interact with the environment by altering its viewpoint. This can be done by controlling the pan-tilt-vergence angles of the cameras so that the range of field of vision is not restricted to a static point of view. The ability to actively control the camera parameters is vital for the robot to achieve robust and real-time perception in a complex and dynamic environment. The close association between perception and action proposed in the active vision paradigm is not limited to camera movements. The processing of visual input is tied closely to the activities it supports (navigation, manipulation, signaling danger or opportunity, etc.) allowing simplified control algorithms.

The first contribution to the field of active vision was made in the late eighties by Krotkow (1987), who proposed a new strategy for observation arguing that camera movements and focusing procedures could improve the robustness of depth estimation algorithms. This approach was later named *active* by Aloimonos in 1988 and *animate* by Ballard in 1991. This paradigm was demonstrated by Coombs (1992) who showed that by using dynamic vergence his binocular system was able to maintain an object within a narrow range of disparity, enabling segmentation with a simple disparity-filtering algorithm (Rougeaux et al., 1994, IROS'94). The result is a robust tracking without the necessity of any information about the object being followed.

It is worth stressing that the active vision system enhances the capabilities of foveated vision by allowing the system to put the object of interest in the center of the visual field (the fovea). This is precisely the way in which the oculomotor system works in many animals.

1.3 Dissertation outline

The thesis is organized as follows. Chapter 2 presents the upper torso humanoid built for this research and the software architecture used to control the robot. Chapter 3 addresses several issues in an attempt to provide a general overview of arguments related to prediction. Some of the topics discussed are: perception, the cerebellum, learning, neural plasticity, development, and motor control. In Chapter 4, I review some of the most relevant models of the cerebellum used in motor control. Chapter 5 addresses the problem of smooth pursuit and presents a short experiment with an active vision head. In Chapter 6, I present an experiment on crossmodal expectation based on audio-visual associations. Finally, in Chapter 7, I present the results of the Robvision project that addressed the complexity of localizing a mobile robot using a geometric memory and an active vision head.

Chapter 2

Eurobot software and hardware architecture

Contents

2.1 Hardware Architecture

A guiding hypothesis followed in this thesis is that cognition and behavior processes emerge from an interaction between brain, body and environment (Lungarella et al., 2004). This is in line with the concept of ontogenetic development that assumes that the system starts with minimal configuration and evolves into more complex configuration by interacting with the environment. Naturally, this concept is correlated with the idea that *action* and *movement* are fundamental elements in this development. In other words, the system needs a *body* with which it can explore and interact with the external world (i.e. embodiment). In this line of research and for the purpose of this thesis I have constructed an upper torso humanoid robot called Eurobot (depicted in figure 2.1). Eurobot has nine degrees of freedom and its aspect and kinematics are very similar to those of Babybot (Metta, 1999). From a human perspective, both robots could be considered as "brothers".

There are, however, some differences between both robots:

- The Eurohead: it is a four degrees stereo head; its most important characteristics are a simplified kinematics, light design and high precision. This head has been widely used during this thesis.

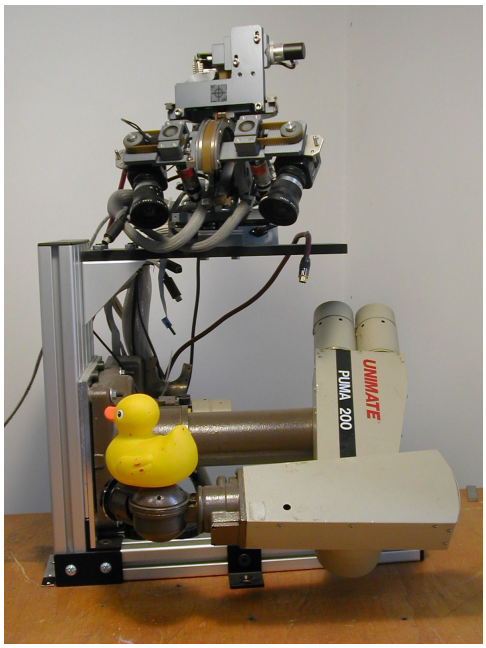


Figure 2.1: The Eurobot

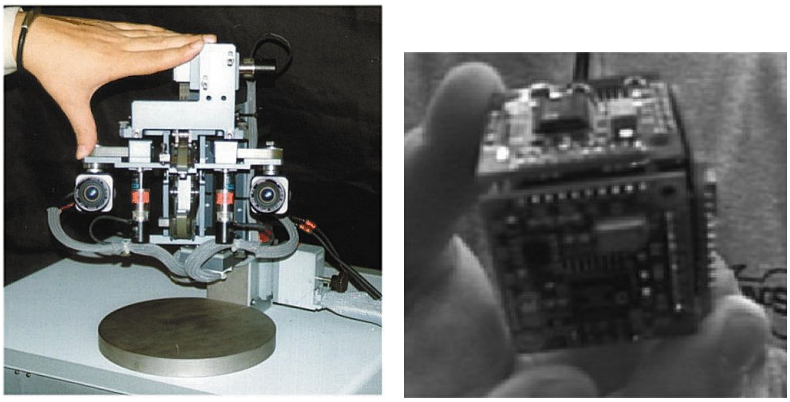
- The body structure: Eurobot has been built with prefabricated materials. This has reduced the construction time and it allows easy modifications of the robot structure (kinematics).
- The control cards: Eurobot is controlled with Galil[1] control cards. The device driver for these cards has been developed.
- The operating system: Eurobot uses QNX[2] a professional hard real-time operating system.

Despite of these differences, both robots run with a very similar software called YARP (to be described in section 2.2). The other minor differences between the robots have been solved through software modularization (e.g. the value of the gains and the reduction gears).

2.1.1 The Eurohead

The Eurohead is shown in figure 2.2(a). It has been designed and implemented to be an accurate vision-based measuring device. For the control of its four degrees of freedom, i.e. the pan, the tilt and the two cameras pan (vergence), four DC motors with harmonic drive reduction gear are used.

[1] www.galilmc.com
[2] www.qnx.com



(a) (b)

Figure 2.2: (a) the Eurohead, (b) the inertial sensors system

	Range(^{o})	Velocity($^o/s$)	Acceleration($^o/s^2$)	Resolution(^{o})
Pan	± 45	≥ 73	≥ 1600	0.007
Tilt	± 60	≥ 73	≥ 2100	0.007
Tilt	± 45	≥ 330	≥ 5100	0.03

Table 2.1: Synopsis of the stereo head characteristics

These actuators have been chosen according to their mechanical characteristics. Due to their harmonic drive gearing, they provide high reduction ratios in a single stage, zero backlash and high precision. Teeth belts have been used for the movement transmission from the actuator to the joints. This gives better results in term of accuracy than usual gearing transmission.

The specifications of the head are summarized in table 2.1. The head was carefully designed in order to be compact, portable and low weight. Its dimensions are $209mm$ x $222mm$ x $185mm$ and its weight is about $3Kg$. It carries standard CCD cameras of 752 x 582 resolution and $4.8mm$ lenses. Moreover, the head employs three piezoelectric gyroscopes (Panerai et al., 2000). Each sensor along with the driving and filtering electronics is mounted on a card of about $3.5cm$ in size. Three cards are arranged so that they form a small modular cube as the one shown in figure 2.2(b). In this manner, the sensing elements are able to measure motion along three orthogonal axes. The cube is mounted on top of the head, so that it monitors the external disturbances that are subjected to the head. The signals end up to the ADCs of the axis control board and are used to compensate the head movements due to such external disturbances.

2.2 Software Architecture

The operating system QNX 6.2 was used as a platform for the development of the software architecture of the robot. QNX 6.2 is a hard real-time operating system with a micro-kernel architecture design. The software development consisted basically on the adaptation of YARP (discussed in section 2.2.2) to work under QNX. Moreover, it was necessary to develop various device drivers and some special control modules.

2.2.1 BTTVX and the Galil device driver

Among the device drivers developed it is worth mention two of them. The first, called BTTVX, is a driver for controlling framegrabber cards based on a microchip commercialized by Conexant called BT848. This chip has evolved into many different versions but its internal architecture has remained as in the original version and therefore the driver can run framegrabbers based in all the available chips. The BT848 chip is a very popular device and many framegrabbers and computer television cards use it as their main video signal digitalization device. It can digitalize the most popular analog video signals, e.g. S-Video, PAL, NTSC.

The driver has been developed with an *open source* philosophy in mind and is available on a public licence. Some research groups are using it for their vision systems and some companies have reported internal use mainly for video streaming applications.

We refer to the second devide driver as the *Galil driver*. It is an adaptation of a driver originally developed for QNX 4.0 —an old version of the QNX operating system. Two problems appeared when developing this software; first, there are a lot of differences between the versions 4.0 and 6.0 of the QNX operating system thus the porting was not easy, and second, it was necessary to *emulate* some control strategies used in the Babybot control cards that simply were not available in the Galil control cards.

2.2.2 YARP: Yet Another Robotic Platform

YARP is a software library for research in robotics. It has been developed as an open source project involving several researchers and laboratories. It is similar to other robotic platforms but with the specific goal to control humanoid robots and to do so in a multi-operating system and multi-computer architecture.

The main components of YARP can be broken down into:

- **libYARP_OS** – interfacing with the operating system(s) to support easy streaming of data across different threads and machines. YARP is written to be OS neutral, and explicitly supports Linux, Microsoft Windows, and the QNX realtime operating system.

- **libYARP_dev** – interfacing with common devices used in robotics: framegrabbers, digital cameras, motor control boards, etc.
- **libYARP_sig** – performing common signal processing tasks (visual, auditory).

2.2.3 Requirements

YARP runs on Windows (2000/XP), Linux (Debian/SuSE), and QNX6. It is based on the open-source ACE (ADAPTIVE Communication Environment) library, which is portable across a very broad range of environments, as a consequence, YARP inherits that portability.

For real-time operation, network overhead has to be minimized, so YARP is designed to operate on an isolated network or behind a firewall. To interface to the hardware YARP relies on the operating system; this means that for each board (frame grabbers, control boards, to mention a few) it needs the appropriate device driver. The **libYARP_dev** library is structured to interface easily with vendor-supplied code creating a software multi-layer abstraction that facilitates hardware replacements. In this way YARP improves software maintenance and provides a robust structure that makes future changes easier.

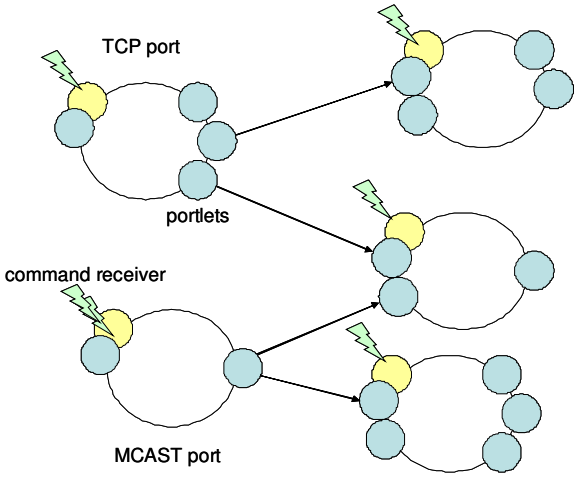


Figure 2.3: Communications model. Every process or thread can own any number of ports.

2.2.4 Communications

One of the key aspects of YARP is the support for communications. The main abstraction for inter-process communication is called a port. A port template class can be specialized to send any data type across an IP-network relying on a set of different protocols. Depending on the protocol different behaviors can be obtained. The implemented protocols include TCP, UDP, MCAST, QNET1, and shared memory. A port can either send to many target ports or receive simultaneously from many other ports. A port is an active object: a thread continuously services the port object. Being an active object it allows responding to external events at run time, and for example it is possible to send commands to port objects to change their behavior. Commands include connecting to another remote port or receiving an incoming request for connection and since all this can be done at run-time it naturally enables connecting/disconnecting parts of the control system on the fly.

Figure 2.3 shows the structure of the port abstraction. Each port is, in practice, a complex object managing many communication channels of the same data type. Each port is potentially both an input and output device although for simplicity of use only one modality is actually allowed in practice. This is enforced by the class definition and the C++ type check. Each communication channel is managed by a portlet object within the main port. Different situations are illustrated in Figure 2.3: for example an MCAST port relies on the protocol itself to send to multiple targets while on the contrary a TCP port has to instantiate multiple portlets to connect to multiple targets. In cases where the code detects that two ports are running on the same machine the IP protocol is replaced by a shared memory connection. Ports can run independently without blocking the calling process (if desired) or they can wake up the calling process on the occurrence of new data. In some cases synchronous communication is allowed (TCP protocol).

Protocols can be intermixed following certain rules. Different operating systems can of course communicate to each other. QNET protocol is an exception and it is only valid within a QNX network. YARP communication code leads to a componentization of the control architecture into many cooperating modules. The data sent through port can range from simple integral types to complex objects such as arrays of data (images) or vectors. Thus controlling a robot becomes something like writing a distributed network of such modules.

In addition, YARP contains supporting libraries for mathematics and robot type computation (kinematics, matrices, vectors, etc.), image processing (compatible with the Intel IPL library), and general purpose utility classes. We also designed a few modules based on existing Microsoft technology to allow remote controlling Windows machines (this support comes naturally on QNX). In short, these scriptable modules complete seamlessly the archi-

tecture allowing the design of scripts to bring up the whole control structure and connect many modules together.

A Matlab interface to ports has been implemented. This allows building Matlab modules (e.g. .m files) that connect to the robot to read/write data. There are basically two advantages: i) complex algorithms can be quickly implemented and tested relying on Matlab existing toolboxes, ii) an additional level of scripting can be realized within Matlab. Matlab provides a relatively efficient and easy to use display library that can be used to visualize the functioning and performance of an ongoing experiment.

2.2.5 Robot independent code

One of the goals in writing our control architecture has been that of simplifying the programming of a complex robotic structure such as a humanoid robot. Control cards come in many different flavors and programming them is usually painful. It would be much better if a standardized interface were provided. It would be even better if a suitable abstraction were available.

To solve the first problem we defined a virtual device driver interface into YARP. To solve the second, we encapsulated the control of parts of the robot (head, arm, frame grabbers, etc.) into a standardized template class hierarchy.

In short, the virtual device drivers bear much of their structure from the UNIX device drivers. Each cards driver class contains three main methods: Open, Close, and IOCtl. The latter is the core of the interface. Each device accepts a set of messages (with parameters) through the IOCtl call. Each message accomplishes a specific function. Two different control cards supporting roughly the same commands can be easily (as it was done in our setup) mapped into exactly the same virtual device driver structure, although clearly the implementation might differ.

The next layer is a C++ hierarchy of classes which through templates includes both the specification of the controlling device driver (e.g. the head is controlled through a certain control card) and the idiosyncrasies of the particular setup (e.g. wiring of the robot might differ, or initialization might require different calibration procedures).

2.2.6 Robot specific interface

The real communication with the robot is carried out through a set of binary modules that use a device driver structure. Module customization is at this stage accomplished through configuration files. In the YARP language these modules are called daemons (a term borrowed from UNIX). The daemons directly interact with the remainder of the robot software through YARP ports and in general they export very specialized communication channels. For example the frame grabber has an output port of type image and the

head control daemon an input port that accepts velocity commands. There are no specific restrictions on the type of ports exported by a daemon since any type of state information about the modules might be required.

Further, some of the daemons accept or send commands of a special type that are generally used to communicate status information. A bus structure based on the MCAST protocol has been implemented to transmit and receive these special messages (called bottles). YARP bottles may contain any type of data or even a group of heterogeneous elements of different types. The structure contains identifiers to properly decode messages and interpret the data. YARP bottles create a network within the network of behaviors to realize a high-level control and coordinate a large number of modules.

Chapter 3

An essay on prediction

Contents

3.1 Introduction

In this chapter I attempt a multidisciplinary review of several issues related to prediction. My purpose is to create a road map of the main areas of research that need to be mastered to understand prediction. This is a vast and complex task, which this essay can only fulfill in a limited fashion. Yet, I think the reader will obtain an adequate overview of the principal issues.

Prediction does not have a formal theory, it is an argument used by many, but it still does not have a formal *research line* to follow. Indeed, we need to survey several research areas to understand what prediction is and how it can be used.

The key idea for understanding the following sections is that prediction is not only the act of extrapolating a trajectory. Prediction is much more than that; and I have chosen to view it as a fundamental capacity of the brain that is used in perception, learning, control and many other basic brain functions.

This chapter is structured as follows. In the section *The paradigm of perception* I discuss the evolution of the concept of perception and how expectations —a form of prediction— and crossmodal information may assist and even dominate the act of perceiving. In the section *The cerebellum* I review this well studied brain organ with its contributions to motor control, sensorial expectations and its fundamental role as a predictive organ. I discuss how the cerebellum works and its more important cellular components. In the section *Learning* I discuss how learning is intimately related to prediction. Indeed, if one wants to make reliable predictions it is necessary to acquire experience about the facts to predict, which is to say, it implies learning. The link between learning and prediction is so intimate that learning can be seen as the acquisition of reliable predictions to the point to consider prediction errors as fundamental signals guiding learning, development and behavior. In the section *Neural plasticity* I explore even more in detail into the neural mechanisms involved in learning. These are related to the modification of neural synapses and the way neural pathways synchronize temporally. In the section *The development of prediction* I analyze how development may affect prediction. I put forward some conceptual ideas about the factors that should be taken into account when studying the development of prediction and review some of the results and conclusions reported by developmental psychology studies. Finally, in the section *Prediction and computational motor control* I review the research area of computational motor control and the main problems affecting control, particularly the effect of delays in the control loop.

3.2 The Paradigm of Perception

Traditionally perception has been considered as a process that has no direct relation to behavior. According to Marr, visual perception is nothing but the transformation of sensory information into a sensory representation, sorting out irrelevant information (Möller, 1997). Slowly, this traditional understanding of perception is being abandoned and a more flexible approach is gaining acceptance where perception is closely tied to action and is considered an active, generative process and not a mere projection (Möller, 1997).

In a more general sense, perception can be considered as a comparison between an expected and an actual state (Berthoz, 1997) and is, therefore, intimately related to prediction. The idea that the use of prediction and expectations in the perceptual cycle improves performance is gaining acceptance in the research community, e.g. Datteri et al. (2003) suggest that perception and action improve in speed and accuracy by expectation mechanisms. In some cases expectations can dominate the perceptual cycle modifying the perception of reality. Berthoz (1997) maintains that, in extreme cases, anticipation can become a prison for perception and a trap for action.

Another important idea is that of considering perception as intimately linked to action. In this context, perception of space and shape can be assumed to be a process of *anticipating* the sensory consequences of actions (Möller, 1997).

3.2.1 Perception as a multisensorial experience

Perception may not depend only on one sense. It is clear from daily experience that humans use several senses to perceive the world around them. Therefore, *crossmodal perception* plays a fundamental role. But, how does the brain manage to associate information from different sensor modalities? Möller (1997) suggest that this association may be based on the detection of statistical interrelations within afferent data. This idea is explored further in chapter 6 where I present an experiment on the creation and exploitation of audio-visual associations. Though not studied in this thesis, it is worth mentioning that the majority of stable easily detectable interrelations are presumably found between actions performed by an agent and their perceptual consequences (Möller, 1997). In other words, actions contribute to the development of the system by actively creating events that elicit multisensorial correlations that contribute on the development of multisensorial *contingency*[1].

3.2.2 Memory: A tool for predicting

What tools does the human brain use to generate expectations? Berthoz (1997) suggests that memory plays a fundamental role in this process. He argues that the information stored in memory is used to predict the consequences of actions and, he hence considers memory as a tool for predicting the future. This idea raises immediately a doubt about how memory stores and organize information. I cannot do justice to the complexity of this question in the limited space of this section allows; I will hence only mention the notion of *schemata* borrowed from cognitive psycology (Neisser, 1976).

The concept of schemata does not give us any idea about how memory is actually organized but it provides an intuitive idea about how memory may work and which is its contribution to the process of perception. Neisser (1976) suggests that schemata are anticipations, the means by which the past affects the future. Moreover, schemata provide anticipation about both temporal and spatial patterns.

3.3 The Cerebellum

The cerebellum is probably the part of the brain that has been studied in most detail. Its particular architecture has attracted the attention of many

[1] something liable to happen as an adjunct to or result of something else

researchers and has facilitated the understanding of its functions (Houk et al., 1996). The cerebellum is believed to have an important role in the predictive capacities of humans (Miall et al., 1996). Smagt (2000) claims the cerebellum performs the following functions:

- The cerebellum is responsible for coordinating movements.
- The cerebellum learns models of the skelotomuscular system.
- The learning process in the cerebellum is influenced by the simultaneous activation of parallel and the climbing fibers at the Purkinje cells.

The participation of the cerebellum in movement is supported by many researchers. This hypothesis has been formulated more than a century ago (Houk and Miller, 2001). In this line of research, Doya et al. (2001) argue that the cerebellum is involved in limb movements and the adaptation of quick eye movements. They discuss the hypothesis that the major role of the cerebellum extends to both the temporal and spatial coordination of movements. Barto et al. (1999) hypothesize that the role of the cerebellum is to eliminate the need for corrective movements by learning to adequately regulate the initial movement. There is also a strong conviction that the cerebellum is a sensory-motor system in which sensory and motor information are integrated (Parkins, 1997). Miall and Reckess (2002) suggest that one of the fundamental roles of the cerebellum is to act as a *sensory predictor* with the particular task of generating predictions about the sensory consequences of motor acts. This capability implies a precise timing, so it is not a surprise that many theories of cerebellar function claim that it is related to temporal processing (Miall and Reckess, 2002).

From a functional point of view, Ito (see Parkins, 1997) suggests that the cerebellum works as an adaptive control system. It is assumed that the cerebellum is functionally an adaptive controller with a comparator for detecting control errors through a comparison of intended and effective controls. Parkins continues analyzing Ito's work saying that, initially, the performance of the whole system relies on the cerebral feedback control, but the cerebellar feed-forward system takes over as soon as it is adapted to the control situation at hand. Schultz and Dickinson (see 2000, pp. 494) call this feedforward based control *predictive mode* and suggest that humans and animals switch to predictive modes as frequently as possible for optimizing behavioral reactions. In other words, once the cerebellum has learned all the dynamics related to a movement, the external feedback control is substituted by an internal feedforward control that uses a *model* of the world.

However, the cerebellum may not be alone in doing all this work; Parkins (1997) suggests that cerebrum and cerebellum complement each other in the process of information representation and processing. On one hand, the

cerebellum represents and processes sensorimotor information in a parallel manner. On the other hand, the cerebrum represents and processes abstract information essentially in a serial manner. In conclusion, Parkins suggests that the information representation and processing characteristics of the cerebellum and cerebrum are complementary and should be able to reciprocally evaluate and correct each other.

More recently researchers have started considering the role the cerebellum plays in cognitive processes (for a review see Parkins, 1997). Parkins (1997) reports that recent publications have studied cerebellar involvement in mental imagery. Houk and Miller (2001) report that recent studies on the cerebellar hemispheres make it clear that the cerebellum does much more than regulate movement. Particularly, the projections from the cerebellum into frontal lobes (considered to be the regions where the cognitive functions reside) supports the case for a significant participation of the cerebellum in cognitive processes.

3.3.1 Cerebellum Structure

The cerebellum presents a peculiar structure where there is clear orthogonal relationship between parallel and climbing fibers and the dendritic trees of Purkinje cells (Houk et al., 1996). Anatomical studies show that the cerebellar cortex is a phylogenetically ancient structure found in all vertebrates, and that it is especially large in primates (Miall and Reckess, 2002). Humans have a particularly complex cerebellum suggesting more powerful predictive capabilities (e.g. generation of prediction further into the future) (Miall and Reckess, 2002).

According to Houk and Miller (2001), in the case of movement regulation, three parts of the cerebellum are active. The portion closest to the midline of the brain is called vermis and phylogenetically it is the oldest part of the cerebellum. This part takes care of the accuracy of some basic movements (e.g. torso, legs, head and eye movements). The middle portion of the cerebellum is the one that regulates the accuracy of voluntary movements (e.g. reaching and grasping). In addition, the cerebellum has lateral parts called hemispheres that may regulate higher aspects of behavior. These hemispheres are particularly large in humans.

From a more operational point of view, Houk and Miller (2001) explain that the cerebellum can be divided in two main areas: the cerebellar cortex and the cerebellar deep nuclei. The cortex specializes in processing large amounts of information regarding body parts and objects. This information is received from the input of the mossy fibers. The mossy fibers project onto the granulle cells whose axons ascend to form the parallel fibers. The parallel fibers are then connected to the Purkinje cells (PC). More that 100,000 parallel fibers connect to each PC. The Purkinje cells specialize in detecting particular patterns appearing in the parallel fibers. They project into the

deep nucleus carrying an inhibitory signal that cannot initiate neural activity by itself. Yet, these inhibitory signals control the spatial and temporal activity of nuclear cells (Houk and Miller, 2001).

The structure and organization of the cerebellum suggest that this organ is a massive information processing structure. But, how does the cerebellum process all this information? Parkins (1997) argues that cerebellar representation and processing have essentially a parallel distribution and analogical characteristics. The cerebellum is considered able to represent and process both spatial and temporal information and to do so in such a manner that space and time are unified.

Nevertheless, if we want to understand how the processing takes place in the cerebellum it is necessary to study the type of neurons present in the cerebellum and the way they are connected. The human cerebellum is composed of about 10 million Purkinje cells, each receiving about 150,000 excitatory synapses via the parallel fibers. Parallel fibers are excited by the mossy fibers coming from the spinal cord, the cerebrum, and the brainstem (Smagt, 2000). Although the cerebellum is formed by several different types of cells I review here only those more interesting from the point of view of prediction: i.e. the *climbing fibers* and the *Purkinje cells*.

Cerebellar climbing fibers. Probably the most intriguing cells in the cerebellum are the climbing fibers (CF). These fibers originate in mammals in the inferior olivary nuclei in the brain stem and are unique to the cerebellum (Peters and Smagt, 2002). According to Schultz and Dickinson (2000) they have a role in activities such as *movement*, *aversive conditioning* and *the generation of prediction errors*. The nature of the signal transported by the climbing fibers is, however, strongly controversial. Different interpretations of the nature of this signal have produced different computational models of the cerebellum. For example, it is not clear whether the CF carry a sensory or a motor signal. Yet, it seems to be almost generally accepted that these neurons transport some kind of a training signal. In concomitance with other factors, the discharge of a CF produces a long-term depression (LTD) in the action of parallel fibers to Purkinje cells synapses as postulated by Albus (Barto et al., 1999).

The paper by Houk et al. (1996, sec. 3.3) contains an interesting discussion about the nature of climbing fibers. Houk, Buckingham, and Barto discuss how various combinations of sensory fibers and collaterals of motor fibers converge in the inferior olive (the origin of the climbing fibers). Accordingly, the inferior olive computes the error signal that is then sent through the climbing fibers. The intriguing point is to understand the nature of the combination between somatosensory and efference copies in the inferior olive. On the one hand, it is possible that motor signals *dominate* the formation of the signal generated on the climbing fibers. On the other hand, this sig-

nal could be dominated by somatosensory information. Another possibility is that the signal carried by the climbing fiber is a weighted combination of both somatosensory and motor information with complex temporal and intensity patterns. The problem is that the nature of the signal transmitted by the climbing fibers has not been established with any certainty yet. Thus, probably, as stated by Miall et al. (1996), the key to a definitive understanding of the function of the cerebellum is given precisely by understanding the functioning of climbing fibers.

Another study that is worth mentioning in this context was done by Keeler (1990, sec. 6.5) and contains an interesting discussion of the timing properties of the climbing fibers. He suggests that part of the role of the climbing fibers is that of transmitting a *synchronization signal*. This idea seems intimately connected to the concept of *eligibility traces* extensively studied during the nineties and that will be discussed in section 3.5.

Purkinje Cells. The Purkinje cells are the sole neural exit from the cerebellum and they project to several structures important in regulating and coordinating movement (Berthoz, 1997). They are large neurons measuring between 21 and 40 μm. Indeed, they are the largest cells of the cerebellum and also its main processing units (Keeler, 1990). Each Purkinje cell contacts about 35 nuclear cells. Purkinje cells generate complex spikes when activated by a a climbing fiber and a number of parallel fibers. The spikes are simpler when the cell is activated only by parallel fibers, basket and stellate cells. The Purkinje cell has only one single climbing fiber which branches out into its dentritic structure (Peters and Smagt, 2002; Keeler, 1990). Ito was the first to discover the inhibitory nature of Purkinje cells. It is believed that neural inhibition is actually one of the basic mechanisms in the production of movement and the main mechanism of sensorimotor training (see Berthoz, 1997, chap.10).

3.4 Learning

If one wants to understand the predictive capabilities of biological systems it is necessary to study the mechanism by which they are acquired. I will hence discuss some aspects of learning.

Learning is important for prediction. Intuitively we can affirm that; if one wants to predict something, one needs a previous experience about the fact in question. Learning is a huge research area and my intention is only to make a short survey of some interesting concepts that may help the understanding of prediction.

What is learning? Looking into a dictionary, the following definitions can be found:

1. Knowledge or skill acquired by instruction or study

2. Modification of a behavioral tendency by experience

From these definitions two main ideas emerge. First, learning implies the acquisition of knowledge; there is a strong body of evidence that in humans and other animals, knowledge arise from the variation of the neural synapses —an issue that I will discuss in the next section. Second, this acquired knowledge can affect current and future behavior. In this sense, Wolpert, Ghahramani, and Flanagan (2001) state that learning involves changes in behavior that result from interaction with the environment and is distinct from maturation. For Wolpert et al. (2001) the goal of this behavioral change is, in general, to improve performance.

3.4.1 Motor learning

Probably, an adequate framework for the understanding of many of the problems associated with learning and prediction is that of *motor learning* (Jordan, 1996). Motor learning has received a lot of attention in recent decades and there is an impressive quantity of findings that cannot be fully reviewed here. Nevertheless, I will expand on the problems associated with motor control in section 3.7 and here I will discuss some research results in motor learning that can be relevant to other kinds of learning.

The first question that comes to mind is: do we learn everything from scratch or do we have some kind of learning bias? Wolpert et al. (2001) explain that humans start with innate patterns of behavior, that conform to a set of hardwired motor skills. They argue that these innate patterns are the result of evolutionary pressures that have predetermined neural connections we have at birth. To understand motor learning, they suggest, we must approach learning as a process taking place not only during one's live, but also during previous generations.

Motor learning can also provide some clues about how time may affect learning. Indeed, it is important to realize that learning may develop over different time scales. For example, Flanagan et al. (2003) reports experiments showing that in humans predictor and controller learning occur at different speeds.

The time difference between learning and unlearning can shed light on about the mechanism of motor learning at work in humans. In works related to human motor control, Wolpert and Kawato (1998) show that there is a temporal asymmetry between learning and unlearning. This makes them hypothesize that learning implies the adaptation of a new module whereas unlearning represents only a switching operation between modules.

Another critical point is to understand in which *space* learning may take place. In the context of an active vision system, Coombs (1992) states that it is important to make the predictions in a coordinate system not perturbed by the control system, so that the target signal can be stable in that

space. A target trajectory may be relatively smooth in head-centered coordinates but can create problems in the visual system if a retinotopic space is used (Coombs, 1992). In a related work, Brown (1990) states that non-retinal representations are more robust and therefore it is better to use head or laboratory centric reference frames to predict retinal images.

In another work related to the study of human gripping, Witney and Wolpert (2003) cite studies that show that anticipatory grip force modulation is scaled to object weight, texture, shape, center of mass and previous experience. They suggest that this prediction is hence not hard-wired but learned through development.

3.4.2 Learning through prediction errors

I will discuss now an interesting work in neuroscience by Schultz and Dickinson (2000) regarding learning and the neural coding of prediction errors. For Schultz and Dickinson (2000), associative learning enables animals to anticipate the occurrence of important outcomes. They suggest that learning may occur when the actual outcome differs from the *predicted outcome*. This difference may produce a neural signal known as *prediction error*. They argue that several brain structures seem to code prediction errors regarding several external events.

Thus, for Schultz and Dickinson learning consist in the acquisition of reliable predictions about future outcomes. Prediction errors lead to the acquisition or modification of behavioral responses. In general terms, Schultz and Dickinson continue, learning can be viewed as the acquisition of the capacity to predict outcomes. In other words, prediction errors are considered as the leading factor that directs learning, although it is not clear whether prediction errors are directly used to create associations ,or rather, they affect quantitatively the attention allocated to the stimuli (Schultz and Dickinson, 2000).

Returning to the issue of the time scale of learning, Schultz and Dickinson show that prediction errors can affect learning after some days (as in the case of the coding of dopamine neurons) or in the course of a few seconds or minutes.

In the previous section, I presented the structure of the cerebellum and how prediction errors seem to be coded in the climbing fibers. Here, I continue the discussion commenting some other neural groups involved in carrying signals related to prediction errors.

Dopamine neurons. According to Schultz and Dickinson dopamine neurons show homogeneous, short latency responses to two classes of events:

- Certain attention-inducing stimuli
- Reward related stimuli

In the case of reward, dopamine neurons code an error in the prediction of the reward (Schultz and Dickinson, 2000). Furthermore, the role of dopamine as a behavioral reinforcer is in line with the role of the Time Difference error in the reinforcement of learning algorithms (Schultz and Dickinson, 2000; Doya et al., 2001).

Superior Culliculus. Schultz and Dickinson report an important implication of this group of neurons for saccadic movements and eye position. In the context of cognitive neuroscience, Berthoz (1997, pag.77) argues that the colliculus is involved in anticipation and motor prediction. In particular, he suggests that the colliculus is a key structure for understanding how the brain handles the problem of spatial and temporal coherence of signals coming from different senses —learning based on different sensorial modalities.

From a more general perspective, it seems that there is a wide agreement that, at least, these two types of learning may exist (Schultz and Dickinson, 2000):

- *Classical conditioning (Pavlovian)*
  Here, the reactions elicited by a signal are controlled solely by the predictive relationship between the signal and the reinforcer.

- *Instrumental conditioning*
  In this case, signals elicit changes in behavior that allow the person to control the occurrence of events. In other words, there is an active interaction that allows to acquire through experience the causal relationship between the reaction and the reinforcer.

Whatever, the particular form of learning, Schultz and Dickinson (2000) suggest that the learning cycle may be formed by the following steps:

1. Generation of predictions of an event.
2. Processing of the event.
3. Computation of the event and its prediction.
4. Use of the prediction error to modify both subsequent predictions and performance (learning).

3.4.3 Computational paradigms of learning

I have discussed so far some of the conceptual aspects of learning, but it is interesting to see how learning mechanisms are practically implemented in computer systems. I hereby consider three computational paradigms for learning which correspond to the three principal ways in which a biological learning system can interact with the environment:

- *Supervised learning;* In supervised learning the environment provides, for each input, an explicit desired output. The goal of the learning system is to learn the mapping from input to outputs (Wolpert et al., 2001). The difference between the learned mapping and the teaching signal, i.e. the error, can be used to modify the mapping itself by using an appropriate learning rule. These learning rules adjust the synaptic weights of the system and take the form of delta rule of backpropagation (Wolpert et al., 2001).

- *Reinforcement learning;* Reinforcement learning is the problem faced by an agent that must learn through trial and error interactions with a dynamic environment. Generally, a reinforcement signal from this environment is used to guide learning. Doya et al. (2001) affirm that, in general, reinforcement learning is notoriously slow for nonlinear, high-dimensional control tasks. It is much more time consuming and harder to apply to large-scale learning tasks (Mehta and Schaal, 2002).

- *Unsupervised learning;* A form of unsupervised learning is Hebbian learning. Donald O. Hebb, a canadian neuropsychologist, postulated that learning is based on the modification of synaptic connections between neurons. He proposed that the repeated stimulation of specific receptors leads slowly to a metabolic change in the synaptic connections of the cell involved.

3.4.4 Exploration versus exploitation

Probably, the most important conclusion is that learning is *context driven* (Smagt, 1998). The learning agent learns through interaction with the environment, but it must also take into account that this learning may be based on incomplete information and, most importantly, that it must be acquired quickly enough to keep up with the changes in the environment (Metta, 1999). Therefore, time has a fundamental role in the learning process. The agent has to decide when to stop exploring and to start using what has been learned (exploitation).

3.5 Neural plasticity

This section continues the discussion of learning. Here I summarize some of the main findings in *neural plasticity* in the cerebellum. Neural plasticity is the ability of neural circuits to undergo changes due to previous activity. From a neural point of view, as addressed by Houk et al. (1996), any model of the cerebellum needs to adopt a rule for modifying synaptic efficacy. This is to say that, if we intend to create a model of the cerebellum, we need to simulate the learning process happening in the cerebellum that presents

itself in the form of neural plasticity. Cellular and chemical mechanisms involved in neural plasticity, however, are extraordinarily complex. Indeed, research about these mechanisms is still a largely uncharted area of investigation. Here, I just discuss some ideas and concepts used in the computational modeling of neural plasticity.

The publications of Marrs (1969) and Albus (1971) about models of the cerebellum encouraged experimentalist to search for a cellular mechanism of synaptic plasticity (Houk et al., 1996). Marr proposed a learning rule similar to the rule used in Hebbian learning and analogous to the *long term potenciation* (LTP). Albus, on the other hand, suggested a mechanism by which the synaptic weight is decreased in the presence of a coincidence signal in three cells: a climbing fiber, a Purkinje cell and a parallel fiber. This learning rule is known as *long-term depression* (LTD) and it has been demonstrated experimentally that it is actually taking place in the cerebellum.

Yet, if one attempts to create a computational model of neural plasticity other problems appear. Houk et al. (1996) address the problem of the *credit assignment*, which is the difficulty of directing training signals to the appropriate sites in the network and at the appropriate moment. Smagt (2000) explains that this problem requires keeping track of which signal from which unit causes an error in the output. Consequently, the credit assignment problem can be divided into two different parts (Houk et al., 1996; Smagt, 2000):

- Structural credit assignment problem
- Temporal credit assignment problem

Considerable research efforts have been made toward understanding these processes of neural adaptation in the cerebellum. In particular, research by Houk et al. (1996), Barto et al. (1999) and Houk and Miller (2001) has been concentrated in creating biologically plausible computational models of cerebellar learning. For example, in Barto et al. (1999) the authors propose a simplified model of neural learning exploiting the concept of *eligibility traces* borrowed from Klopf (1972,1982). This hypothesis suggests that appropriate activity at a synapse creates a synaptically local memory trace. This local memory makes the synapse *eligible* for modification if and when the appropriate training information arrives within a short period of time. Barto et al. explain that this allows the learning rule to modify synaptic weights based on the synaptic actions that ocurred before the relevant climbing fiber information is available. In a work dealing with the creation of predictive models of smooth pursuit based on eligibility traces, Kettner et al. (1997) report that the model lost its capacity to learn predictive tracking when not using eligibility traces. Eligibility traces appear as a mechanism of neural synchronization, such that signals with different time delays contribute to synaptic adaptation.

Therefore, *synchronization* may have a fundamental role in the learning processes in the cerebellum and probably everywhere in the brain. Furthermore, the signals to be synchronized do not need to be of the same modality. In this context, Berthoz explains that one problem the brain has to solve in order to enable the fusion of multisensory information is that of time shift. He indicates that the solution adopted by the nervous system is called *temporal windows*. This is, in the words of Berthoz, "a memory developed by the neural network of the colliculus that maintains the sensitivity of the multimodal neurons during certain time."

The concepts of eligibility traces and temporal windows seem to overlap to a large extent. Or at least, both ideas try to solve the same problem —how to put together asynchronous signals. In addition, one can think of temporal synchronization as taking place at different levels, from a more synaptic based synchronization to a more cognitive based one.

3.6 The development of prediction

To understand prediction it is necessary to study also its evolution during development. In this case, we may formulate several questions:

- Is prediction a phylogenetic or a ontogenetic process?
- How do humans develop prediction?
- When do humans start using predictions?
- Do humans use the same neural structures for all types of predictions?
- How do humans develop the neural structures necessary for prediction?

These and many other questions arise when thinking about the development of prediction. Unfortunately, we have not yet found the answer for them. Based on current research results, we can provide only some partial answers.

As we do not have yet discovered the genes involved in prediction, it is difficult to answer the question of whether prediction is the consequence of a phylogenetic or a ontogenetic process; however, we already have some clues. It is known that at birth the human cerebellar cortex has a well established architecture. It is also believed that the cerebellum has an important role in predictive capabilities, particularly in those related to movement and sensorial expectations. Thus, we can make the hypothesis that some of the neural mechanisms involved in prediction are phylogenetic structures and are available at birth. The way these mechanisms adapt to particular types of prediction would be an ontogenetic process.

Yet, the latter hypothesis is incomplete because it considers development as starting only at the moment of birth. For a complete understanding of

the problem it would be necessary to analyze the development of the fetus. Prenatal movement is present in all animals and is believed to begin long before the twelfth week of gestation (van Heijst, 1998). It is possible that these prenatal movements have an influence on the ontogenesis of predictive mechanisms (e.g. in motor control). In this early ontogenic development, the consequences of living in an *aquatic* environment should be taken into account. Many animals spend the first weeks of their lives in an aquatic environment, to be later *born* in a gaseous environment. The main consequence of living in a fluid is the reduction of the gravity force that could help both the development of muscles and the acquisition of rough models of body dynamics. Moreover, the contact with the internal wall of the uterus provides the fetus with a clear feedback signal when moving the limbs, and this could help in the development of early control and predictive neural structures.

Moreover, the pathways variation of feedback signals during development should also be taken into consideration. The change in neural pathways has as consequence modifications in the nature of the feedback signals during development, that is, feedback signals could suffer significant changes in timing, composition and intensity.

Unfortunately, the hypotheses discussed above are difficult to test. Technical difficulties associated with prenatal research have limited our knowledge about development before birth. We are only now starting to understand plasticity changes in neural connections in adult subjects. To my knowledge, prenatal studies are mainly concerned with the quantitative aspect of fetal movement, though perhaps in the near future, using non intrusive three dimensional visualization tools, it will be possible to perform more complete experiments to analyze the dynamics of the fetal movements and their relation to prediction.

Much more information is available when we study development after birth. Indeed, this is the main research field of developmental psychology. From this field, we know that the predictive capabilities of humans increase considerably with age; probably this is due to both maturity and the adaptation of the neural mechanisms involved in prediction.

To understand this process we can refer to some works on the development of smooth pursuit. Smooth pursuit (SP) has attracted a lot of attention from the research community and has been studied in detail during the last half century. This makes SP an excellent starting point for understanding the developmental processes that take place in the acquisition of prediction capabilities.

Several important findings have been reported in the study of SP in young infants. Infants one month old present a delay of 180 microseconds when following an object with the eyes (Von Hofsten and Rosander, 1997); therefore, the earliest expressions of smooth pursuit do not predict the target very well and the performance of this prediction depends on the size of the stimulus (Von Hofsten and Rosander, 1996). Moreover, SP in these

early months of life is step-like (i.e. based on saccades) (Johnson, 1997). It seems that a predictive strategy appears between two and three months of age, when infants start following the target smoothly often staying on the target or little ahead of it (Von Hofsten and Rosander, 1996, citing Aslin). At this age infants are also able to anticipate the appearance of a picture by moving the eyes before the image appears (Von Hofsten and Rosander, 1996, citing Haith). Because of the internal lags of the perception-action cycle, this control strategy can only be explained in the context of predictive control. Indeed, this predictive strategy works well for sinusoidal trajectories but it fails for triangular ones. This could be explained with the fact that the sinusoidal trajectory can be easily predicted using a local velocity extrapolation, whilst the triangular trajectory requires a more complex prediction mechanism based on expectations or learning of the periodicity of the signal (Von Hofsten and Rosander, 1997).

These results suggest that a combination of maturation, adaptation and learning may take place during a long period of time thus allowing the subject to increase the complexity of predictions. First the subject is able to make simple predictions based on local extrapolations of the trajectory. Then, in a second stage, these extrapolation mechanisms generalize factoring in predictions based on global trajectory features such as periodicity or pattern probabilities. Third, the subject learns intersensorial associations and learns crossmodal expectations. After that, the subject starts developing even more complex predictions based on cognitive processes. All these steps are developed in parallel with the capacity to predict more and more into the future, due to the performance improvement of the neural mechanisms involved in prediction.

3.7 Prediction and computational motor control

In this section I discuss some of the problems encountered by computational motor control. Interestingly, these problems are also the ones faced by the human brain and it is likely that the brain uses prediction mechanisms to solve many of them. The study of motor control is fundamentally the study of sensorimotor transformations and includes also the study of dynamics (Jordan, 1999). Just as a terminology note, it is worth noting that in control theory the word *state* is used to define the variables that specify the configuration of the body; for example, these variables can be joint angles or positions (Wolpert et al., 2003). Other variables with slower and discrete change rates are considered to be the *context* (Wolpert et al., 2003).

3.7.1 Problems in control

Therefore, some of these problems are:

State estimation. To obtain accurate control, the system has to know with precision both the state and context of the body (Wolpert et al., 2003). The motor command depends on the state of the system (Wolpert and Kawato, 1998). However, the internal state and the context are not directly available to the system and they need to be inferred from the sensory feedback available. Moreover, in biological systems (and also in some robots) these signals can be of different nature: auditory, visual, tactile and propioceptive. In general, it is assumed that the state and the context can be estimated using a function of these signals. After this, the estimation can be used to control the system.

More generally, Wolpert (1997) identifies three cues available to estimate the state:

- Sensory inflow (e.g. propioception)
- Motor commands (efference copies)
- A combination of both the above

Delays. When dealing with sensor signals one has to address the problem of delay. Indeed, sensory signals can have a processing delay of more than 150 milliseconds in the case of the human visual system and of several milliseconds in robotic systems depending on the type of sensor used. The main difficulty is that these delays can create stability problems in using sensory information as control signals, particularly in the case of feedback control (Brown, 1990; Robinson, 1987; Coombs, 1992). Delays can be due to several causes:

- Processing
- Transduction (Wolpert et al., 2003)
- Transport of sensory signals
- Switching between control modes

Humans succeed in controlling a complex system such as the human body despite the delays. Indeed, several strategies can be used to deal with this problem. Three of them are:

- *Slow down system gains* This case is not very optimal because the performance of the system suffers (Brown, 1990).
- *Intermittency* It alternates movement and rest, waiting for a sensory validation after each movement (Wolpert, 1997).

- *Prediction* In general prediction refers to estimating future states of a system; but it can include many other functions (Wolpert and Flanagan, 2001). This strategy seems to be the one used by humans and other animals.

Synchronization. Another problem that may arise with different sensor and control modalities is the problem of synchrony. Sensor modalities can have different sampling frequencies which can make sensor integration difficult. The signals can occur up to several hundred milliseconds apart (Wolpert et al., 2003). In humans, this problem seems to be solved by a mechanism similar to the mechanism of eligibility traces explained in section 3.5. But it is likely that the nervous system is also able to regulate the velocity of conduction of nerve fibers and neurons to make signals arrive at the same time (see Berthoz, 1997, pg. 82).

Neural noise. Neural noise is definitely a problem in biological systems, but also in artificial ones. In artificial systems we deal with noise by using stochastic system analysis (discussed in section 5.3.5). In biological systems, however, it is not so clear how this phenomenon is treated . It has been suggested that noise may play a fundamental role in development. Neural noise can be caused by many factors, as for example immature neural pattern innervations or lack of myelination (Metta, 1999). Moreover, it is likely that noise affects not only the sensorial signals but also the motor commands (Wolpert et al., 2003).

Switching between control submodules. Another problem a control system has to face is that of the *interaction of subcontrol components* (Brown, 1990). This is specially the case when control is exercised through an adaptive system. The switching time between components can be a problem. Particularly, in the case where one of the control modules is a memory, the control architecture may face problems accessing and storing data.

Non-linear properties. This is one of the biggest problems faced by artificial systems that try to imitate biological control methods. The skeletomuscular system in humans is highly non-linear. Non-linearity has important computational consequences and can make difficult mathematical modeling.

High dimensional state of the motor system. The other problem associated with the control of such a complex structure as the human skeletomuscular system is its high dimensionality. This can lead to the known problem of the course of dimensionality (Bellman, 1957).

3.7.2 Computational representations for motor learning

The theoretical framework of *motor learning* is a possible alternative in dealing with the problems discussed above. Motor learning can be seen as the learning of the necessary *mappings* representing the sensorimotor transformations needed to perform a desired control. In artificial agents, these mappings take the form of computer based data representations acquired through some of the learning methodologies explained in section 3.4. These data structures can take the form of some of the representations listed below:

- Lookup tables
- Basis functions
- Parametric representations

3.7.3 Motor planning

Another problem faced by biological and artificial systems is that of motor planning. Motor planning can be viewed as the computational process of selecting a single solution in the motor hierarchy (Wolpert, 1997). In the case of artificial systems, a computational framework used to solve such a problem is called *optimal control theory*, where a cost function is minimized to find a solution (Wolpert, 1997). Indeed, one of the main goals of research in motor control has been to understand and model the cost function that the human brain uses to solve the problem of motor planning. Although much effort has gone into reverse engineering this cost function, the control method used by humans is still unknown (Wolpert, 1997). Yet, this research has developed various methods of trajectory planning that can be divided into four main groups:

- Kinematic
- Dynamic
- Force fields (Mussa-Ivaldi, 1997)
- Minimum variance

The first two try to minimize some particular variables like the jerk or the rate of change of the torques. The main difference between them consists in the parameters used to calculate the cost function and the degree of separation between planning and execution (Wolpert, 1997). The potential field approach is based on the experimental observation that any position of the arm configuration space can be obtained by a linear combination of a small number of motor primitives each represented by a torque field (Metta, 1999). Finally, the minimum variance theory assumes that noise has great

importance in motor commands, assuming that the noise increases proportionally to the *intensity* or volume of the motor command. Thus, when a movement is done very quickly the noise increases. Consequently, the error at the end of the movement is bigger than in the case when the movement is slower.

3.8 Chapter summary

This chapter deserves a summary of the salient points discussed so far. The first thing to remember is that prediction does not follow a distinct research line in the traditional sense. If one wants to create a robot with predictive capabilities similar to those of humans, one has to combine several research lines in a multidisciplinary approach, and even in this case, research results have not yet produced enough information to completely understand prediction. The second thing is the fact that prediction may affect many other ongoing processes in humans and other animals. I address the particular case where perception is interpreted as the comparison between an expected and an actual state. Perception must be understood as a multisensorial experience, and consequently, the synchronization and integration of different sensory signals acquire a special relevance. One should also consider the possibility that, crossmodal expectations may guide perception as well. In the third place, I have offered a description of the cerebellum. This brain organ has a particular relevance in movement, prediction, sensorial expectation and dynamics modeling. The fourth issue has been learning and how it seems to be encoded in the brain through variations in neural synapses —neural plasticity. Learning can be seen as the acquisition of reliable predictions. Prediction errors may be fundamental signals used throughout the brain to guide learning. Finally I have discussed some details related to the development of prediction and some problems that appear in motor control which, in all likelihood, humans and other animals solve by means of predictive mechanisms.

Chapter 4

Computational models of the cerebellum

Contents

In recent years, there has been considerable efforts to use cerebellar based controllers in robotic systems. The appearance of light-weight robot manipulators has recently attracted the interest of research groups working in this area (see Smagt and Bullock, 1997). There is a good reason for that. The traditional control methods have made an enormous progress in automation. Industries have successfully adopted robotized systems in manufacturing (e.g. car manufacturing industry). Why then are robots not used everywhere? The reason is that currently robots can successfully operate in constrained, well defined, environments. In these applications usually the interaction between the robot and the world is limited and controlled. One would like to create robots, however, that are more similar to humans. In this case, one encounters several problems that traditional methods are unable to address. Two of these problems are: *flexible links* and *flexible joints*. If one wants a light-weight robot, this usually means that the rigid body assumption used in classical robotics needs to be abandoned. The low weight of the links usually correlates with weaker materials that consequently have more flexibility. Moreover, the use of artificial muscles or high-ratio gear boxes allow for a certain degree of compliance at the joints. As already explained in section 3.7, two more complications must be taken into account: *times delays* and *high-dimensionality*. The time delays can be of many types

and can be particularly important in sensory feedback. Although computer power is constantly increasing, very often algorithms are still too complex to allow real time performance (this is especially true for image processing or when multiple sensorial sources are involved).

Consequently, one needs to study alternatives to current methods and consider architectures that will be better suited to the complex control required of the new generation of robots. The computational models of the cerebellum is one of the options. These models deal with aspects that have not been considered in traditional methods such as learning and adaptation.

So, can artificial cerebellar models compete to control robots? (Smagt and Bullock, 1997; Smagt, 2000). Probably, the question can be partially answered by saying that it *depends on the robot to control*. However, for the moment, there is still a big gap between existing applications and cerebellar based control. Usually, the latter is limited to the control of 2-link simulated robots arms (Smagt, 2000).

There have been two main lines of research attempting to develop plausible biological models of the cerebellum. There are many differences between these two approaches, but probably, the most controversial issue is whether the cerebellum creates *internal models* or not. Mehta and Schaal (2002) suggest that the two approaches can be classified in two conceptual groups of control: *direct control* and *indirect control*.

The differences between the two approaches are not confined to the philosophy of control, but extend to the way of understanding brain mechanisms. Direct control is dominated by a bottom-up approach where researchers try to model the morphology of the cerebellum as well as its ability to adapt at the cellular level. In contrast, indirect control is strongly based on the idea that the cerebellum implements internal models. This line of research has approached the problem from the point of view of control theory thus trying to simulate the cerebellum at a functional level.

In the following sections, I review the more relevant models of the cerebellum. The review of each model is organized in four subsections. First I present its historical development and theoretical concepts. Second, I discuss its range of capabilities. Third, I talk about the learning methods used. Fourth, I discuss the principal problems. And finally, I review the main improvements suggested in the literature for each given model.

4.1 Cerebellar Model Articulation Controller (CMAC)

4.1.1 Origins and concepts

Albus was the creator of the CMAC model. This model was the computational implementation of some ideas about how the cerebellum works. Albus improved and modified some ideas first proposed by Marr. This is why this model is popularly known as the Marr-Albus theory model.

The CMAC model seems to have been based on an earlier model called BOXES (by Michie and Chambers). BOXES essentially implements a lookup table, thus the extension proposed in CMAC resolves the generalization problems associated with this kind of representation (Smagt, 1998).

4.1.2 Capabilities

CMAC considers the cerebellum to be a perceptron-based associative memory that controls elemental movements (Smagt and Bullock, 1997). The cerebellum is considered a context-driven pattern recognition system (Doya et al., 2001). In the model, granule cells work as combinatorial encoders of sensory and motor variables and the Pukinje cells work as pattern classifiers. The climbing fiber inputs are used as teaching signals (Doya et al., 2001).

One of the most important features of CMAC is the discretization of input signals through the input sensors. Basically, the overlap of the receptive fields produces input generalization, while the offset of the adjacent layers of the receptive fields produces input quantization. In other words, the operation of the CMAC can be described in terms of a large set of overlapping, multidimensional receptive fields with finite boundaries (Peters and Smagt, 2002).

Smith (1998) states some of the advantages of CMAC:

- The mapping and training operations are extremely fast.
- The algorithms used by CMAC are easy to implement.
- Local generalization prevents over-training in one area of the input space from degrading the mapping in another.
- CMAC is useful in real-time adaptive control because of its speed.

4.1.3 Learning

CMAC has often been considered only a function approximation model (Smagt, 1998). The network output is linearly related to the weights of the respective signals thus the learning of a pattern is instantaneous and does not require an iterative procedure. The weights are modified using a delta rule.

4.1.4 Problems

Yet, CMAC is only a crude approximation of the cerebellum. An important shortcoming is that it ignores the inhibitory nature of the Purkinje cells (Smagt, 1998). Moreover, the original CMAC was not able to compute partial derivatives information (see Smagt and Bullock, 1997, chap.2). Indeed, this problem becomes highly relevant when the system is asked to create predictions in areas of the state space where the training data is

sparse (Smagt and Bullock, 1997). Another problem may be that CMAC does not take into account time (Smagt and Bullock, 1997).

According to Peters and Smagt (2002) the Marr-Albus theory focuses on the cerebellar cortex and associates each Purkinje cell with elementary movement. The cerebellar cortex is considered to be an isolated system, without taking in consideration the fact that the cerebellum is integrated in a more complex system (Peters and Smagt, 2002).

4.1.5 Improvements

A recent improvement was suggested by Kraft (see Smagt, 1998, pag.12); this improvement is called smoothed CMAC because the gradient of the network output can be calculated. This gradient is useful for dynamic control in robotic systems.

Another recent improvement on CMAC is the Fairly Obvious Extension (FOX) by Russel Smith (Smith, 1998). The major difference of FOX with respect to CMAC is that the climbing fiber carries error information which is filtered and used for weight modifications using eligibility traces (similar to APG –see bellow). Moreover, elegibilities are in vectorial form instead of a scalar form (Peters and Smagt, 2002).

4.2 Adjustable Pattern Generator(APG)

4.2.1 Origins and concepts

The adjustable pattern generator was introduced by Houk and colleagues. The term *adjustable* is used since the model is able to generate a burst command with adjustable intensity and duration (Houk et al., 1996; Peters and Smagt, 2002).

The APG is based on the same understanding of the cerebellum structure as CMAC. Basically, it contains the same *state encoder* that the CMAC and the improvement used in CMAC for the state encoder can be used for APG (Peters and Smagt, 2002).

4.2.2 Capabilities

The APG has only been used to control simple systems (i.e. single muscle or two-link robot arm having six muscles) (Peters and Smagt, 2002).

According to Miall et al. (1996), in the APG model, the key role for the cerebellar cortex is to modulate and terminate motor commands; however, this models ignores that the cerebellum may function also as a sensory predictor or state estimator (Miall et al., 1993).

Each module in APG includes a positive feedback loop between a cerebellar nucleus cell and a motor cortical cell. Each nucleus cell receives inhibitory

input from a private set of Purkinje cells. Each set of Purkinje cells receives a private climbing fiber training signal. A key assumption is that climbing fibers train Purkinje cells to recognize particular patterns of parallel fibers activity that indicate when desired endpoints are about to be reached (see Houk et al., 1996).

4.2.3 Learning

Although CMAC and APG have the same conceptual bases, they differ strongly in the learning mechanisms. The synapses of the Purkinje cells learn by the reinforcement given by one climbing fiber for each APG. The climbing fiber gives an external error signal (Peters and Smagt, 2002).

APG is continuously learning while controlling the system, thus it is to be considered an on-line learning controller (Peters and Smagt, 2002).

The most relevant characteristic of APG learning is that it uses eligibility traces. The learning rule has two components: a strong one named long-term depression (LTD) and a much weaker one named long-term potentiation (LTP) (Peters and Smagt, 2002).

4.2.4 Problems

Some authors complain about the positive feedback of the APG model. Positive feedback can be difficult to control and, consequently, can provoke excessive activation (Miall et al., 1996).

4.2.5 Improvements

Barto et al. (1999) present an extensive work about a simplified version of the APG model. This model was used to investigate the timing and predictive processes taking place in the cerebellum in the control of movement. According to the authors, the model does not make explicit predictions and does not use a forward model. The system simply learns to generate a motor command in a manner that elicits desired future behavior.

4.3 Internal models

4.3.1 Origins and concepts

According to Houk et al. (1996) internal models have fascinated control theorists with their potentialities in system control. The underlying idea is that the cerebellum creates internal models that predict responses when supplied with sample commands. Once learned these models become an integral part of the controller (Houk et al., 1996).

It seems that Ito was among the firsts to hypothesize that the cerebellum provides models of the body and the physical environment and that these

models may be used to achieve accurate control despite the time delays in sensory feedback (Doya et al., 2001).

Indeed, the necessity to create a model of the plant (e.g. the arm) is a requirement that emerges also from control theory (Doya et al., 2001). Skilled motor behavior requires both inverse and forward internal models. Any good controller can be thought of as implicitly implementing an inverse model of the system (Wolpert et al., 2001).

4.3.2 Capabilities

One of the leading ideas in this field is that the cerebellum provides an estimate of the current state of the motor system by employing a forward model of the plant (Miall et al., 1996; Wolpert et al., 1998).

In the case of the control of the arm, Wolpert et al. (1998) state that internal models can permit a control with reduced stiffness.

We can identify two types of internal models (Wolpert and Kawato, 1998):

- *Forward models*

  Forward models are directly involved in prediction. By using an efference copy of the motor commands, they are able to anticipate the sensory effect of movement (Jordan, 1999). Forward models can provide a fast internal feedback loop that contributes to the stability of the system. Moreover, this internal loop can be used to assist an imperfect controller (Miall and Reckess, 2002).

  In general, according to Wolpert et al. (2001), the prediction generated by the forward models can be used for: *state estimation*, *sensory confirmation and cancellation*, *context estimation*, *mental practice, imitation and social cognition.*

  Several research results support the hypothesis that the cerebellum provides a forward model of the motor system (Miall and Reckess, 2002). Evidence of the existence of forward models in the cerebellum, however, is difficult to find, because the output of the forward model is not used directly as a visible output (i.e. measurable experimentally). Instead, it is used indirectly to facilitate control processes (Mehta and Schaal, 2002).

  There are two kinds of forward models

  - *Forward dynamic model* These models capture the forward relationship between inputs and outputs. They predict the next state of the system (Wolpert, 1997). Their output, adequately delayed, can be used to create a teaching signal to improve control.

  - *Forward output model* Forward output models produce an estimate of the *sensory output* (Wolpert and Kawato, 1998). In other words, given the estimated state of the system (computed by the forward dynamic model) they are able to predict the sensory feedback (Wolpert, 1997). This fast feedback signal can be used to cancel the predictable part of the real sensory feedback.

- *Inverse models* They provide the motor input that will cause a desired change in the state of the plant. Therefore, they are well suited to act as controllers (Wolpert and Kawato, 1998). They can transform sensory variables into motor variables, that is, they transform desired sensory consequences into motor commands that yield these consequences. Internal models are the basic module in open-loop control systems.

Smith Predictor. A feedback control system that uses forward models both for mimicking the plant (forward dynamic model) and for canceling predictable feedback (forward output model) is known as *Smith Predictor* (Miall et al., 1993). It is believed that the cerebellum can act as such a controller. The Smith Predictor has been criticized because it needs to learn two models: one for forward dynamics and another one for feedback delays. However, these models present different learning rates, so it is plausible that they are learned in parallel (Wolpert et al., 1998). Experiments show that the learning process is slower when the delays of the feedback signal are artificially modified (Wolpert et al., 1998). This suggests a strong correlation between learning and the timing of the feedback signals.

4.3.3 Learning

Both forward and inverse models can be combined to create plausible learning structures. The models do not necessary need to be accurate; they can be coarse and adapt to new situations by using learning signals to restructure their internal neural synapses. Some learning architectures have been proposed and studied from a computational perspective, which suggests that biological systems use similar learning mechanisms.

Learning the forward model. Learning the forward model is easier than learning the inverse model. The basic idea is that forward models are not fixed entities but must be learned and updated through experience (Wolpert et al., 2003). Forward models can be easily learned by using supervised or self-supervised techniques comparing the predicted and the actual outcome of a motor command (Wolpert and Kawato, 1998).

Learning the inverse model. The learning of an inverse model can be more complicated. Usually, the signal used to teach the inverse model is the *motor error*. Unfortunately, this signal is not directly available to the Central Nervous System(CNS) but only in the form of an error in sensorial space (e.g. in visual retinal coordinates). Therefore, a transformation is necessary from sensory coordinates into motor coordinates (Wolpert et al., 1998).

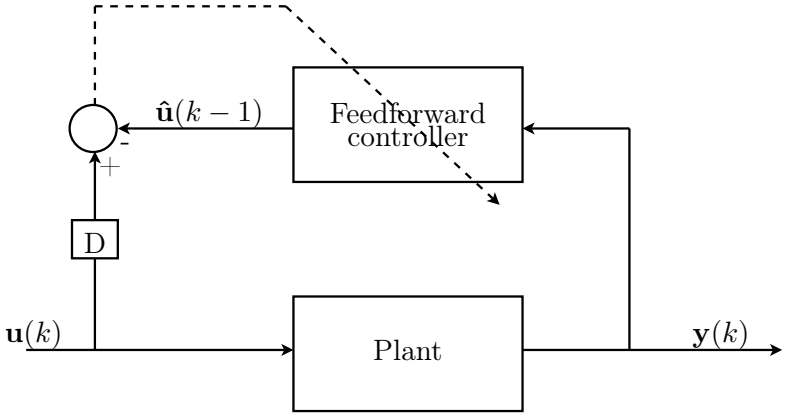


Figure 4.1: Direct inverse modeling architecture

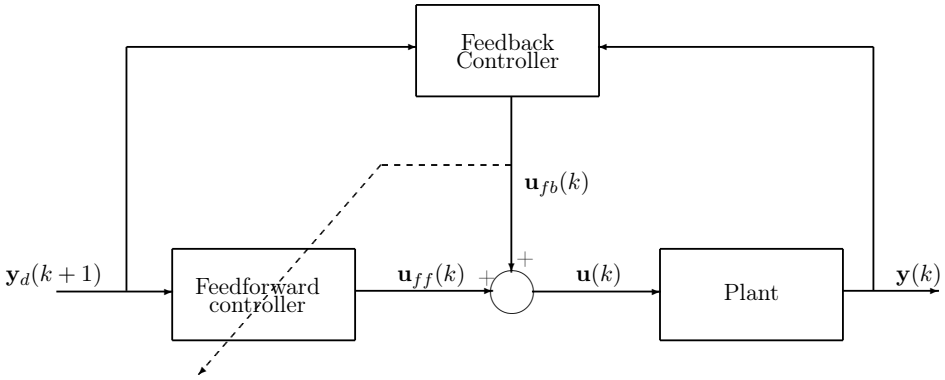


Figure 4.2: Feedback error learning

Different solutions have been proposed in the literature to solve this sensorimotor transformation problem:

- *Direct inverse modeling* In this type of learning, which architecture is depicted in figure 4.1, the output of the plant is provided as an input to the learning controller which in turn is required to produce

as output the corresponding plant input (Jordan, 1999). Plant input and controller output are compared to produce a learning signal that is used to regulate the controller parameters. Consequently, an inverse model of the plant can be obtained by supervised learning.

- *Feedback error learning* Figure 4.2 depicts the architecture of a feedback error learning system. A feedback controller is used to guide the learning of the feedforward controller. This architecture, originally proposed by Kawato (Kawato and Gomi, 1992), is inspired by the side-path model (Peters and Smagt, 2002). The control architecure has two main components: a fixed linear feedback controller and an adaptive nonlinear feedforward controller (Doya et al., 2001). The control signal that governs the plant is the sum of the output of both controllers; the learning signal for the feedforward controller is the output of the feedback. The desired plant output is used for both, control and training purposes; therefore, the feedback controller is trained on-line (i.e. control and learning happen at the same time). Goal directed control is used as the desired plant output is directly used for control. This model takes its biological clues from the assumption that the activity of the inferior olive essentially reflects a motor error (Houk et al., 1996).

- *Distal supervised learning* In distal supervised learning, the controller is learned indirectly, through the intermediary of a forward model of the plant; the forward model must be also learned observing the inputs and outputs of the plant. Both controller and forward model form a composite learning system, i.e. a single computational unit from an architectural point of view. If the controller is an inverse model then the composite learning system should be an identity transformation; therefore, using this identity transformation as a learning constraint, the controller can be indirectly trained by fixing the parameters of the forward model (Jordan, 1999).

4.3.4 Problems

In direct inverse modeling, the learning approach is not adequate for non-linear systems. The existence of multiple solutions could lead the learning algorithm (e.g. a last squares cost function) to produce a result that is the average of these solutions, thus producing an incorrect controller. Moreover, direct inverse modeling presents two different phases for learning and control, and requires necessarily a switching mechanism between them. This strategy is known as off-line learning.

In the case of feedback error learning, some researchers express their doubts that it may have some convergence problems, particularly in the control of non-linear systems. For example, Peters and Smagt (2002) state

that although feedback error learning can be proven to be globally stable, this is not always the case locally. Due to internal dynamics it can become locally unstable. Moreover, feedback error learning has been criticized as it lacks mechanisms to take into account time delays in the control loop. Another point of criticism is that feedback error learning needs an accurate and continuous trajectory to learn from (Wolpert et al., 2001). However, the question of whether the brain does or does not generate a desired trajectory is still an open debate in the research community.

4.3.5 Improvements

One of the principal improvements regarding internal models is probably that of considering architectures where multiple internal models are used together. This is the subject of the next section.

4.4 Multiple Internal Models

4.4.1 Origins and concepts

Modularity can simplify the control problem. In this sense, multiple internal models can be regarded conceptually as motor primitives. The general idea is that an architecture with different internal models can manage successfully many dynamically different situations (e.g. different object, different contexts) (Wolpert and Kawato, 1998).

Research results present more and more evidence that the cerebellum is a good candidate to contain multiple paired forward-inverse models (Wolpert and Kawato, 1998).

4.4.2 Capabilities

According to Doya et al. (2001), the basic question in using multiple controllers is how to select an appropriate controller under a given condition. A solution is to feed all controllers and select the one that gives the best performance. This solution, however, can be time consuming when there are a large number of candidate controllers (Doya et al., 2001).

According to Wolpert and Kawato (1998) a fundamental aspect in this type of control is that switching should be based on prediction errors rather than on performance errors. This way by using prediction the system can select *a priori* the correct action. In computational terms, the sensory prediction error from a given forward model is represented as a probability (Wolpert et al., 2003). This probability is used to create a *responsibility signal* that in turn is used to select the adequate module. The responsibility signal is derived from the combination of two processes:

- Forward model predictions

- Sensory contextual cues

A system that implements the latter control technique is called *MOSAIC* (Wolpert et al., 2003). MOSAIC runs multiple forward models that predict the behavior of the motor system; then these predictions are compared with the actual feedback. Each predictor is paired with a controller, so when the forward model achieves a good prediction its correspondent controller is selected to control the current action.

Another possibility is that the multiple internal models handle a nonlinear control task by integrating the control output of multiple linear controllers (Doya et al., 2001).

4.4.3 Learning

According to Wolpert and Kawato (1998), the problem of learning and control is best solved by using multiple controllers. Interestingly, a key question in experimental psychology research is whether humans use one internal model that must be tuned each time or more internal models that specialize in particular parts of the motor space. Motor adaptation experiments suggest that humans do not just tune the parameters of a single controller but retain multiple controllers and switch them on the fly (Doya et al., 2001).

Another interesting question is whether, inside a module, the forward models will be learned before the inverse model or vice versa.

4.4.4 Problems

The concatenation of optimal local control policies is not guaranteed to produce a globally optimal policy (Doya et al., 2001). The switching between different modules can be time consuming and create timing problems in the control. Lastly, the feedback prediction can be wrong, so in this case a fast switch based on real sensory information is necessary to perform the correct control action.

4.4.5 Improvements

An improvement on MOSAIC is called Hierarchical MOSAIC (HMOSAIC), because it consists of several layers of MOSAIC. According to Wolpert et al. (2003) the hierarchical architecture embodies a way to reconciling top-down plans and button-up constraints.

Chapter 5

Smooth pursuit

Contents

5.1 Introduction

In this chapter I make a review of the history of smooth pursuit and then I present a short experiment on smooth pursuit implemented on an active vision robotic head. An excellent review about smooth pursuit can be found in Pavel (1990). Although a little out of date, the paper by Pavel (1990) is the most complete and exhaustive review about smooth pursuit I have found. Therefore, to a certain extent, I follow his structure.

The oculomotor system has attracted a lot of attention in the research community and there exists an enormous body of literature. In this section I concentrate on a particular branch of research dedicated to the modeling of the oculomotor system using techniques of control theory. In particular, I will discuss works related to a particular function of the oculomotor system called *smooth pursuit*.

Smooth pursuit refers to the capacity of the oculomotor system of tracking a moving target on the fovea (Shibata and Schaal, 2001, IROS01). This task is not an easy one because the oculomotor system has delays that can affect the control of the system. Interestingly, it has been observed that individuals can not make smooth pursuit in the absence of a moving visual stimulus (Barnes, 1993).

Probably, the most intriguing question that has interested researchers is the fact that the oculomotor system tracks objects so well in the presence of

delays. These delays have been measured to be between 150-200 milliseconds. It is generally accepted that the human oculomotor system uses some kind of prediction mechanism to deal with the delays problem (Wexler and Klam, 2001; Klam et al., 2000; Pavel, 1990; Brown, 1990; Barnes, 1993; Von Hofsten and Rosander, 1996).

I have already discussed in section 3.6 some details about the development of smooth pursuit. The main idea is that humans seem to improve their predictive capabilities with time. This could happen because both the neural mechanisms involved in prediction are maturating and other neural structures are collaborating to create more complex predictions. Probably, the development of prediction and in particular of smooth pursuit involves a combination of both. But it must be also taken into account that humans can *learn* to predict (Pavel, 1990).

Many researchers agree that the oculomotor system uses a trajectory extrapolation mechanism, however, as noticed by Barnes (1993), the oculomotor system should include also other mechanisms like: periodicity estimator, sample and hold mechanism, intermediate storage system, conflict monitor and gain regulator. We should add also the saccades system and the VOR mechanism.

The saccadic mechanisms is out of the scope of this thesis, however, it is worth noticing that it also presents a predictive behavior. Moreover, saccades play a fundamental role in the initial *catch up* of the object of interest and help in correcting the smooth pursuit when errors are detected (Klam et al., 2000).

5.2 Models of smooth pursuit

Control theory has been used by many researchers as a theoretical framework to understand the complexity of the oculomotor system. Although the oculomotor system is not linear, linear models have been used extensively to study the eyes movement; in many cases, this has been done assuming that the oculomotor system can be approximated by a linear system when dealing with small signals (Pavel, 1990).

When modeling the oculomotor system one faces two technical problems: the delays and the coordination of subcomponents (Brown, 1990). Much of the classical work has been devoted to study how a mechanical system can use information to predict future paths (Pavel, 1990). The basic idea guiding this research is that the oculomotor system has evolved to minimize the output error (Pavel, 1990). In other words, when pursuing a moving object it is necessary to minimize the difference between the target position and the eye position. This difference is called the tracking error (when using the position) or retinal slip (when using the velocity) (Pavel, 1990).

5.2.1 Classical control theory approaches

I first analyze the *classical* control theory techniques used to model the oculomotor system; these techniques use various negative and positive feedback loops that are used to control the system. The general idea of such control architectures is to have a system with the maximum performance while, at the same time, avoiding instability.

Smooth pursuit control has traditionally been described as a negative feedback system. According to Leigh and Zee (1999) this kind of control offers some advantages:

- A prompt and accurate response to stimuli
- Relative insensitivity to changes in internal parameters

Negative feedback control offers many other advantages, and indeed, it is used extensively in robotics (e.g. PID control). However, in the presence of delays feedback control presents oscillations and instability (Leigh and Zee, 1999; Coombs, 1992). Instability occurs when a control system combines feedback, delays and large forward gains (Robinson, 1987). Therefore, a pure feedback system seems not adequate to model the smooth pursuit if we consider that the human oculomotor system has good stability and performance in the presence of large delays in the sensorial feedback loop.

Many of the early models of the oculomotor system use the retinal slip — the derivative of the tracking error— as a feedback signal to control smooth pursuit. However, in this case, the control signal approximates zero when accurate tracking is being achieved (Brown, 1990) and consequently, the system lags with respect to the target. This does not occur in the human oculomotor system that is able to maintain the fovea on the tracked object and, in some cases, anticipate it. This discrepancies between modeling and the real performance of the human oculomotor system lead Young and colleagues to suggest that the signal controlling smooth pursuit may not be the retinal slip (i.e. the velocity error) but an internal representation of the target in space (Brown, 1990; Coombs, 1992; Leigh and Zee, 1999).

Robinson proposed a way to implement the idea of Young (see Brown, 1990, for a review); his suggestion was to combine an efference copy of the eye velocity signal with the retinal error. This combination produces an estimation of the real target velocity because takes into account the motion of the eyes in the orbit during the tracking (Brown, 1990; Leigh and Zee, 1999).

In practical terms, Robinson's idea is to feedback this efference copy through a positive feedback loop to transform the system in an open loop control system, and avoid instability problems. However, using this technique the nice properties of negative feedback control are lost and it is necessary to have an accurate model of the delays of the system to precisely

combine the negative and positive feedback loops (Coombs, 1992). This can be difficult in a biological system taking into account that the real delays are due to the physics of the system whereas the estimation of the delays depends on neural modeling (Leigh and Zee, 1999).

5.2.2 Optimal control

Although classical control theory approaches have been useful for understanding the problems associated with the oculomotor control they are still very limited as models of the oculomotor system. The main success of these approaches has been that of obtaining a phase lag reduction in the smooth pursuit; however, this is not enough to understand completely the anticipatory behaviors of the oculomotor system. Indeed, one needs to take into account issues like learning, anticipation based on internally generated signals, and other contextual cues (Pavel, 1990).

Optimal control theory overcomes some of the problems associated to the classical approach. In this framework a control problem can be characterized by specifying three things (Pavel, 1990):

- The dynamic of the system
- An objective function
- Additional constrains

Pavel (1990) contains an exhaustive discussion about the use of optimal control theory to model the oculomotor system. Using a deterministic approach, Pavel concludes that the controller of such a model must be, in fact, a predictor. This predictor needs to be able to anticipate the position of the target τ seconds in advance and, for this reason, must have precise information about its trajectory.

Other problems arise in the Pavel's deterministic control system, but the main idea to take into account is that an estimator of the future target trajectory is needed. Moreover, in a real environment this estimator should deal with the randomness and signal noise associated to biological processes. The latter can be taken into account using the framework of stochastic system analysis (Pavel, 1990). Maybeck (1979) contains an excellent introduction about why stochastic models should be used. He presents three reasons: (1) no mathematical model is perfect, (2) there are disturbances which we can neither control nor model deterministically, and (3) sensors do not provide perfect and complete data about the system.

Consequently, it is necessary to take into account that noise can be present in the dynamical system but also in the sensorial observations (i.e. the measurements).

5.2.3 The Kalman filter

If we consider prediction as the extrapolation of a trajectory then it seems natural to use previous information about that trajectory to make the prediction. The use of trajectory past history to generate predictions seems reasonable because there is little else on which to base predictions (see Landy et al., 1996, chap. 2). However, if one wants to generate the prediction based in all the previous trajectory history the problem becomes rapidly computational intractable. Therefore, it is interesting to use *adaptive techniques* that allow the system to constantly upgrade the predictions using information from prior observations. In other words, "new information is acquired in small steps and at each stage it is added to what is already known" (see Pavel, 1990).

A good example of an adaptive method (also known as recursive techniques) is the Kalman filter. The Kalman filter has been used in several applications since it was first published by R.E.Kalman (Kalman, 1960). This filter generates predictions about the future state of the system and it does so tacking into account the noise in both the system and the measurements. The formulation of the Kalman filter can be found in many papers but I include it here for completeness.

The Kalman deals with the problem of obtaining an estimate of the state $x \in R^n$ of the system that can be represented with the next difference equation:

$$x_k = A \cdot x_{k-1} + B \cdot u_k + w_{k-1} \tag{5.1}$$

where k represents the discrete time step, x_k is the estimated state at time t_k, x_{k-1} is the state at time t_{k-1}, u_k is the measurement at time t_k and w_{k-1} represents a random noise. A and B are matrixes that relate both the previous state and the measurement with the estimate of the state.

In this system, the measurement $z \in R^m$ is calculated as:

$$z_k = H \cdot x_k + v_k \tag{5.2}$$

where the matrix H relates the state x_k to the measurement z_k.

The noise is represented by the random variables w_k and v_k. These random variables model respectively the noise in the system and in the measurement. The variables w_k and v_k are supposed to be white, independent, and with a normal probability distribution. That is:

$$p(w) \sim N(0, Q) \tag{5.3}$$

$$p(v) \sim N(0, R) \tag{5.4}$$

where Q is the noise covariance and R is the measurement noise covariance.

The Kalman filter performs two steps: a *predition* and a *update*. The equations for these two steps can be seen in table 5.1.

Prediction
$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$ $P_k^- = AP_{k-1}A^T + Q$
Update
$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$ $\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$ $P_k = (I - K_k H)P_k^-$

Table 5.1: Kalman filter equations

5.3 A short experiment on smooth pursuit

To experimentally test some of the results found in the literature about smooth pursuit we implemented a smooth pursuit experiment on an active vision system (Baroni, 2002).

5.3.1 Experimental Setup

The experiment was done using the active vision system Eurohead (described in section 2.1) and a linear sliding track (see figure 5.1(a)). The sliding track has a computer controllable cart that can be programmed to move with different velocities and accelerations. The programming is done through a serial cable connected to the control unit shown in figure 5.1(b). By putting a colored object on the sliding cart we were able to perform smooth pursuing experiments using the color segmentation method described in the next section.

(a) (b)

Figure 5.1: (a) is a detail of the sliding cart, (b) is the control box of the sliding track

The complete experimental setup can be seen in figures 5.2 and 5.3. The sliding cart was controlled using a computer based on the Windows operating system. The binocular head control and all the other processing

were implemented on a rack of computers running QNX 6.2 operating system, actually this was an early version of the Eurobot setup described in secion 2.1. For the acquisition of the images in the QNX operating system we developed a device driver (discussed in section 2.2) to access the framegrabber cards based in the chip Bt848 commercialized by Conexant (Beltrán-González, 2002). For the control, we used eight axis control cards developed by Galil, and also in this case we developed the device driver necessary to control the cards under the QNX 6.2 operating system.

Figure 5.2: The complete smooth pursuit experimental setup

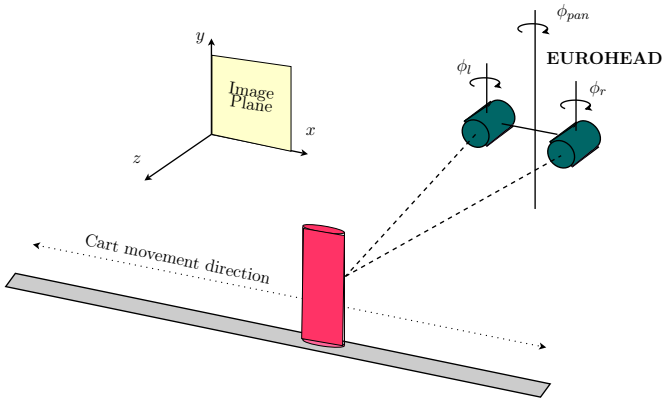


Figure 5.3: A schema of the complete system

5.3.2 Color segmentation

To track the colored object mounted on the sliding cart we implemented a color segmentation algorithm. We used the same technique presented in (Metta et al., 2004). The first step of this technique is to transform the

RGB color information arriving from the cameras to a Hue Saturation Value (HSV) space. The transformation of the RGB values into HSV space is performed according to the following equations:

$$H(R,G,B) = \arctan\left(\sqrt{3}(G-B),(R-G)+(R-B)\right) \quad (5.5)$$

$$S(R,G,B) = 1-\frac{\min\left(R,G,B\right)}{R+G+B} \quad (5.6)$$

Complex techniques can be used to segment an object automatically (Metta et al., 2004), however, for simplicity, we performed our color segmentation using a threshold value calculated empirically (in the HS space). This technique was sufficient to segment the object of interest. This object was a red bar ($15cm$ long). Figure 5.4 shows the object and the result of the color segmentation.

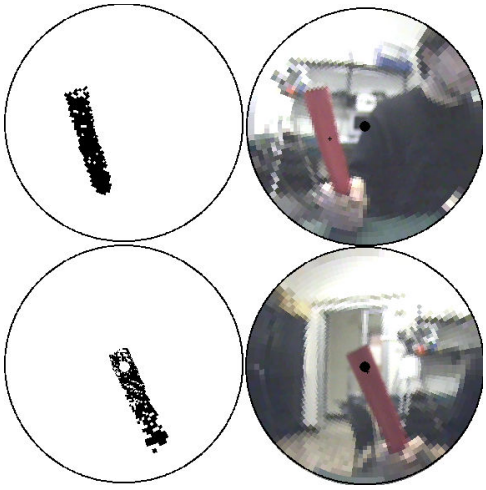


Figure 5.4: The segmentation result. On the right are the log-polar images as acquired by the robot. The experimenter is presenting the red bar in front of the robot. On the left, the result of the color segmentation process.

Once the object is segmented, the center of mass (x_c, y_c) of the object can be computed as:

$$x_c = \frac{1}{A}\sum_x\sum_y xI(x,y) \quad (5.7)$$

$$y_c = \frac{1}{A}\sum_y\sum_x yI(x,y) \quad (5.8)$$

where $I(x,y)$ is a binary pixel value obtained from the segmented image (Metta et al., 2004). The object centroid is a good measurement of the object position and can be used to calculate the retinal distance from the object to the center of the image. This distance is the *retinal error* that can be used to control the binocular head through a feedback loop.

5.3.3 The control method

However, the retinal error can not be directly used for control. It is necessary to perform a transformation from the sensorial space (retinal) into the motor space (for a review see Jordan, 1999) (i.e. a sensorimotor transformation). According to Metta et al. (2004) this problem can be formalized with the formula:

$$\mathbf{e} = \mathbf{C} \cdot \mathbf{s}(t) \tag{5.9}$$

where $\mathbf{s}(t)$ is the retinal error, $\mathbf{e}$ is the error expressed in motor coordinates and $\mathbf{C}$ is a convenient transformation matrix. A way to calculate this matrix is as follows:

$$\mathbf{C} = \left(\frac{\partial \mathbf{s}}{\partial \mathbf{q}} \right)^{-1} \tag{5.10}$$

Finally, the motor command can be calculated as:

$$\dot{\mathbf{q}} = -\lambda \cdot \mathbf{e} \tag{5.11}$$

where $\dot{q}$ is the joint speed and λ is a positive constant gain.

The matrix $\mathbf{C}$ can be calculated using an automatic least-square technique (Metta et al., 2004); however, for the purpose of this experiment its coefficients were calculated beforehand. As a result the head was able to follow a colored object moving in front of the robot.

However, as already discussed in the previous sections, in this way the head always lags behind the object being tracked because the visual feedback has a delay due to the visual processing. To overcome these limitations and improve the performance of the tracking, a Kalman filter was include in the control system.

5.3.4 Kalman filter implementation

For the experiment we used two types of Kalman filters. The first assumes that the object being tracked has constant velocity, whereas the second assumes constant acceleration.

The Kalman filter prediction was applied to the coordinates of the object center of mass in the image plane, thus making a trajectory prediction on a two dimensions plane (see figure 5.3). However, the sliding cart moved mainly in the x axis of the image plane so the variations in the y axis were mainly due to errors in the calculation of the center of mass of the moving object. These errors occurred because the color segmentation procedure suffers of noise and leads to errors in the localization of the object.

Figure 5.5 shows the reduction in the retinal error when using the Kalman filter. Figure 5.5(a) shows an instant in the tracking process where the fovea (the black spot in the center of the image) is clearly far away from the center of mass of the object (the cross in the left part of the image). In the

(a) (b)

Figure 5.5: (a) an snapshot of the moving bar without prediction, (b) a snapshot of the bar moving with prediction

other image, figure 5.5(b), it is shown how the system can obtain a better performance using the Kalman filter prediction. In this case, the system manage to put the fovea almost on the object center of mass. However, the snapshots presented in figure 5.5 are among the best results obtained in this experiment. The system obtained this performance when the characteristics of the real movement were close to the assumptions made in the models (i.e. constant velocity or constant acceleration). The interested reader can consult Baroni (2002) to get detailed information about the results obtained during the experiment.

5.3.5 Drawbacks and discussion

In the experiment the predictions were done in the image plane and consequently the performance decreased because we did not take into account the motion of the head. This is related to the problem already discussed in section 3.4 and in section 5.2.1 about the space used for trajectory representation. If one uses a representation based in head-centric reference frame the prediction of the trajectory will not be affected by the motion produced by the robot. In this case, however, a coordinates transformation would be necessary to calculate the position of the object. There is some evidence that humans develop the capacity to use extra-retinal coordinates during the first 6 months of life Johnson (1997). However, there is little understanding of the neural mechanisms involved in such transformation.

Nevertheless, from a robotics point of view many improvements can be made to the system. For example, it is worth mention implementations of the Kalman filter based in the Interacting Multiple Model (IMM) (Bradshaw et al., 1997) where a set of N Kalman filters modeling different types of motions is used to produce a better prediction of 3D movements. As I said before, we used two Kalman filters but these were interchanged beforehand. Therefore, our system lacks of an automatic filter selector as the one proposed in the IMM architecture.

Another implementation by which or system could be improved has been

called Signal Synthesis Adaptive Control (Coombs, 1992; Bahill and McDonald, 1983) where a group of already learned trajectories are stored in a memory. During smooth pursuit the system selects from the memory the trajectory that better fits the observed movement. It is however not clear in this architecture how the system recovers and accesses the information stored in the memory.

Chapter 6

Attention modulation based on crossmodal expectations

Contents

6.1 Introduction

6.1.1 Motivation

This chapter address the problem of crossmodal perception already discussed in section 3.2. The focus is on understanding how an artificial system can improve its perceptual capabilities by using multimodal cues. This work is divided in two main sections: (i) we study how a system can segment objects based in visual and sound cues, and (ii) we discuss how the system can use this object segmentation to create visual expectations that may guide attention based only on audio cues.

The motivation of this work is based on the hypothesis that perceptual and attentional mechanisms are fundamentally crossmodal processes. That is, different senses participate in the act of perception. This sensor collaboration, however, is not simply the act of sensor fusion. Indeed, we consider

the fact that a sensor modality can elicit an expectation on another sensor modality. For example, the smell of food can be sufficient to identify the components of a particular dish even before we can see it, the sound produced by another person when walking can be sufficient to identify him completely and create a visual expectation of the person you expect to see. These crossmodal influences may guide in an unconscious level many of the daily perceptual processes.

Let us introduce how this conceptual approach may fit in the history of computer vision.

6.1.2 Short history of machine vision

Though machine vision is a long and well established scientific discipline the several decades of intensive research were not enough to resolve a problem that humans seem to pass by almost intuitively. During the early days of computer vision, i.e. 1960's and 1970's, the research efforts were concentrated in the passive processing of single images. The visual cortex was believed to process all the information in the field of view and to do so in a sequential and increasingly complex process. This doctrine influenced the vision computational models of that time that tried to create general descriptions of the visible scene (Landy et al., 1996). This period culminated with the publication of the influential Marr's work *Vision* (Marr, 1982), where computer vision was formalized as a pure information processing task.

In the late 1980's and early 1990's a new approach to computer vision appeared : *Active Vision* (Aloimonos et al., 1987; Ballard, 1991). During the late 1990's the concept of active vision was exploited, improved and expanded. A strong emphasis was made in the simplification of the early stage vision problems (see (Vieville, 1997)) by exploiting the explorative capacities of vision systems. During this period, there was also an approach between "cognitive sciences" and robotics that yielded to epigenetic approaches to robotics and the investigation of the perception-action paradigm where the artificial system is able to act in the world and modify it (see (Metta, 1999) for a remarkable example).

More recently, the perception-action paradigm has been explored further in the area of humanoid robotics. For example, Metta and Fitzpatrick (Metta and Fitzpatrick, 2003) have shown how to segment an ambiguous object from the background by active manipulation. Several researches stress that an agent could construct a self image by actively exploring and manipulating (Natale, 2004; Arsenio, 2004).

However, though the *learn by doing* approach (Fitzpatrick et al., 2003) is providing encouraging results, we think that further development is still necessary. Particularly, the exploitation of intersensorial relations for the improvement of perception has not been sufficiently explored, e.g. the interrelation between sound and vision. In this chapter this problem is addressed

studying audio-visual causal interrelations. In particular, it is studied how these interrelations may improve object perception and how they could be exploited to create intersensorial expectations.

6.1.3 Chapter outline

The chapter is organized as follows: first, we propose a conceptualization of crossmodal perception; second, we analyze the research in automatic sound recognition and show how traditional speech recognition techniques can be used to parametrize sounds produced by objects; third, we discuss how an approximation of a statistic called *mutual information* can be used to create a common intersensorial space for sound and vision; fourth, we present how the latter information can be used to segment an object from the background with the assistance of a color back-projection technique; and finally we show how the system: a) creates a sound-object associative memory, b) uses this memory to recognize sounds (through a dynamic time warping algorithm) and c) extracts from the memory a visual expectation associated with a sound auditory event.

6.2 Toward cross-modal perception

Neuroscience research is actively studying cross-modal relations in the human brain and several researchers suggest that perception is a multisensorial experience. However, still many questions remain unanswered, for example:

- How cognitive pathways may dominate perception (top-down approach)
- How different sensorial modalities are integrated.
- How these sensorial interrelations may guide development and learning.
- How sensorial modalities may influence each other.

There is little understanding on how these mechanisms may work in the human brain. However, some conclusions can be advanced: a) sensorial interrelations seem to be fundamental for the development of high level cognitive abilities, b) perception seems to depend strongly on multisensorial cues.

In the robotics research field, we can categorize these assumptions in the context of the crossmodal perception paradigm. We conceive crossmodal perception as an *extension* of the active-vision/perception-action paradigms. The crossmodal perceptual agent uses multisensorial cues to reinforce its explorative perception and creates actively synchronized multisensorial inputs (e.g. by hitting repeatedly an object on the ground producing a change in both the visual field and the auditive input).

We attempt the first steps toward this kind of perception by trying to solve, in the context of a humanoid robot architecture, two problems: a) *object segmentation using multisensorial cues*, and b) *sound classification for attentional priming*. More formally, we suggest that these two problems could be conceptualized into two distinct phases:

- *Syn**a**esthetic Phase*: From *syn* "co" and *aisthanesthai* to perceive. This yields to an etymological interpretation as *joint perception* or *to perceive simultaneously*. In this particular experiment, this phase corresponds to an object segmentation based on the integration of sound and visual cues.
- *Synesthetic Phase*: A concomitant sensation; especially, a subjective sensation or image of a sense (as of color) other than the one (as of sound) being stimulated (from Merriam-Webster Online). In this experiment, this phase is formed by a sound classification algorithm that can *remember* the visual aspect of an object.

Notice that syn**a**esthesic and synesthetic are very similar (there is only a letter "a" difference), they have a common etymological origin, however, the meaning is slightly different in our interpretation. Moreover, it is worth noting that these words have been used interchangeably in the literature; particularly, *synaesthetic* is used in cognitive neuroscience to address an *unusual mixing of the senses* that affects certain people (see (Rich and B.Mattingley, 2002) for an extensive review). This unusual mixing of senses interpretation stress the "strength" how senses interrelate, and also the *non relation with the real world*. For example, patients experience visual hallucinations (i.e. they *see* colors) when hearing a particular noise or they have smell hallucinations when they see a particular number.

We adopt a slightly different interpretation; we believe that most humans have *subjective* sensations activated by crossmodal interrelations. The differences with respect to the cognitive neuroscience point of view are: a) the "intensity" of the interrelations, and b) that they correspond to *real* sensorial experiences. In our view, the interrelations are related with sensorial *expectations* and not with sensorial hallucinations.

Thus, paraphrasing Fermüller and Aloimonos (see (Landy et al., 1996) chap.9), we may say that:

> Now, it has become clear that image understanding should also include the process of selective acquisition of data in space and time **from multisensorial cues**.

6.3 System architecture and experimental setup

Sound could be considered as important as vision. However, comparatively, little research has been done in the field of sound recognition. The research

has been mainly concentrated in the recognition of speech and music and in the study of orienting behaviors (Natale et al., 2002). More recently, the sound research has included works in scene analysis (Stäger et al., 2003), detection of talking faces (Hershey and Movellan, 2000) and rhythm detection (Arsenio and Fitzpatrick, 2003).

In a work related with the segmentation of objects by a humanoid robot, Arsenio and Fitzpatrick (Arsenio and Fitzpatrick, 2003) address the problem of object detection based in the rhythm properties of movements, both in sound and vision streams. They address the recognition of toys designed for infants. We use a similar approach, but we do not exploit the rhythmic characteristics of movement but the intrinsic common information created in both sensorial streams when the toy is squeezed or shaked by the experimenter in front of the robot.

We will show that a combination of speech recognition techniques and statistics can be used to create a crossmodal perceptual architecture that can create associations between the images of toys and the sounds the toys produce; and, in a second stage, evocate the toy's visual image by recognizing the sound associated to the toy, and consequently, have the potential to exploit this visual expectation in explorative movements.

In Figure 6.1 we present the architecture of the crossmodal perceptual system. This system was implemented in YARP (Yet Another Robotic Platform)(Metta et al.). YARP is a framework for humanoid robotics development that provides support, among other things, for distributed computation and multi-operating system communications.

The proposed system was running in a rack of standard PC's, either with Microsoft Windows or QNX installed on the PC's. The system received its inputs from the environment through a PAL camera and two microphones. A standard PCI framegrabber, based on the Conexant Bt848 chip (Beltrán-González, 2002), digitalizes the images which are converted utilizing a log-polar mapping by a software conversion algorithm (Sandini and Metta, 2002). A standard audio PCI card digitalizes the sound signal obtained by the microphones. Both cards use a Direct Memory Access (DMA) mechanism to transfer the data streams into the computer main memory.

A fundamental problem was how to synchronize in time the video and sound streams. The standard acquisition cards do not employ any hardware synchronization line, so we developed a special device driver in the Windows operating system that controls the acquisition of both cards. The driver initializes the acquisition cards in a sequential manner using a software *critical region* (i.e. a code execution flow that is not interrupted). Consequently, the acquisition software needs to run in a single process being executed by a single computer.

However, the previous mechanism does not guarantee a perfect synchronization, and for this reason, the computer internal clock (*timestamps*) was employed to monitor the time alignment between the data streams. Using

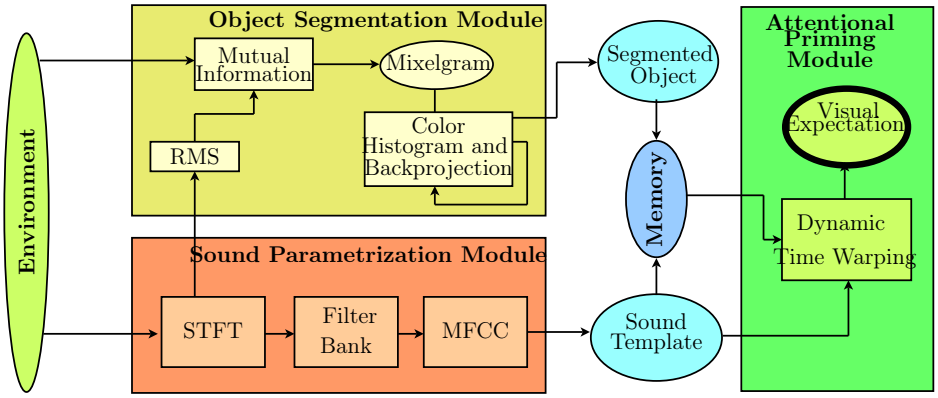


Figure 6.1: The architecture of the crossmodal perceptual system

this technique we measured a time difference that in average of several tests was less than one millisecond with different CPU loads.

The sampling frequency for the sound was 44100 Hz, whilst the precision was 16 bits and the transfer memory was 1764 samples/frame. This produced a sound framerate of 25 frames per second at a rate of one frame each 40 milliseconds, exactly the same as the frame sequence rate of the PAL video images.

(a) (b) (c)

Figure 6.2: Experiment objects as perceived by Eurobot: (a) A deformable plastic yellow duck, (b) a hollow hard plastic blue pig filled with plastic bottle caps, and (c) a hollow hard plastic red pig filled with chickpeas.

For the experiment, we used an upper torso humanoid robot called Eurobot and a set of three baby toys acquired in commercial stores. Figure 6.2 shows the group of toys as seen by the robot. Figure 6.2(a) is a deformable yellow plastic duck; it produces a high frequency sound when squeezed with the hand. The hollow hard plastic toy pigs shown in Figures 6.2(b) 6.2(c) are the same toy; the differences are: they have different colors and we have filled them with different materials. Therefore, the sound produced by each

toy pig was slightly different.

6.4 Sound Parametrization

The goal of the sound parametrization module was to obtain a low dimensional representation of sound. In the speech recognition literature this module is known as the signal-processing front-end. The idea is to have a sequence of measurements of the input signal, usually the output of some type of spectral analysis technique, that yields a "pattern" that represents the sound; though we prefer the term *sound template* for this representation. This sound template is a sequence of spectral vectors. Each of these vectors represents the frequency transformation of the sound in a short period of time; in our system, this period of time has a duration of 40 milliseconds. Therefore, the sound template is a representation of the sound both in time and in frequency.

To implement this sound parametrization module, we reviewed the most popular techniques used in speech recognition and, based on several research reports, we chose a technique called mel-frequency cepstral coefficients (MFCC). The MFCC algorithm can create a compact representation of sound into a vector of few parameters. We tested the MFCC algorithm in the Matlab environment using the auditory toolbox developed by Malcolm Slaney (Slaney, 1998) and then we implemented a C++ version based in his algorithm for the YARP environment.

Algorithm 1 Calculate MFCC

loop
  Window the data with Hamming window
  Apply Fast Fourier Transform
  Compute the magnitude of the FFT
  Convert the magnitude into filter bank outputs
  Find the log_{10}
  Find the cosine transform to reduce dimensionality
end loop

Algorithm 1 shows the steps suggested by (Slaney, 1998) to compute the MFCC transformation. In the next sections we explain in detail the parts of the algorithm.

6.4.1 Short-Time Fourier Transform (STFT)

The traditional approach to spectral analysis of the sound signal consists in applying a set of filter-banks (see (Rabiner and Juang, 1993)). Figure 6.3 shows graficaly how these filters may be applied to the sound data flow.

Mathematically this operation is expressed with the formula

$$\begin{aligned} s_i(n) &= s(n) * h_i(n), 1 \leq i \leq Q \\ &= \sum_{m=0}^{M_i-1} h_i(m)s(n-m) \end{aligned} \tag{6.1}$$

where $(*)$ is a convolution, $s(n)$ is the sound stream at time n, $h_i(n)$ is the response of the i bandpass filter and Q is the number of filters applied.

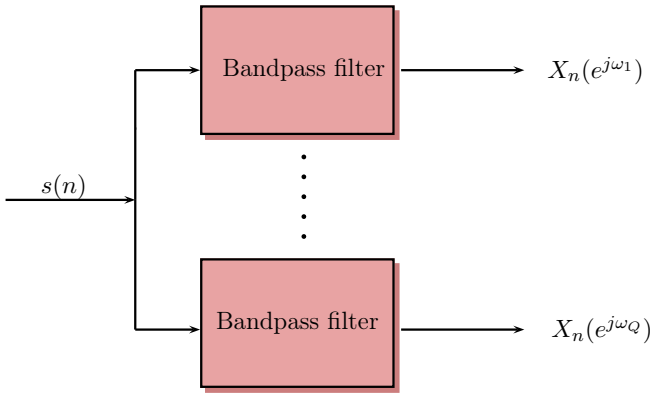


Figure 6.3: Processing of the sound frame through a set of bandpass filters

According to (Rabiner and Juang, 1993), the filter bank computation can be conveniently implemented by applying first a short-time fourier transform (STFT) to the incoming data. By substituting $h_i(n) = w(n)e^{j\omega_i n}$ and considering $x_i(n)$ to be the discrete version of $s_i(n)$ the equation (6.1) can be extended to

$$\begin{aligned} x_i(n) &= \sum_m w(m)e^{j\omega_i m}s(n-m) \\ &= \sum_m s(m)w(n-m)e^{jw_i(n-m)} \\ &= e^{j\omega_i n}\sum_m s(m)w(n-m)e^{-j\omega_i m} \\ &= e^{j\omega_i n}S_n(e^{j\omega_i}) \end{aligned} \tag{6.2}$$

where $S_n(e^{j\omega_i})$ is the short-time Fourier transform of $s(n)$ at frequency $\omega_i = 2\pi f_i$.

Consequently, from equation (6.2) we can extract the form of the STFT and rewrite it as:

$$S_n(e^{jw_i}) = \sum_m s(m)w(n-m)e^{-jw_i m} \tag{6.3}$$

The form of $w(n)$ can be conveniently chosen for the application. In our case, we chose a Hamming window. The mathematical form of a Hamming window is presented in the next formula:

$$w(k) = 0.54 - 0.46\cos\left(2\pi\frac{k}{n-1}\right), \quad k = 0, \ldots, n-1 \tag{6.4}$$

The shape of the window, which is produced by the above equation, is depicted in figure 6.4.

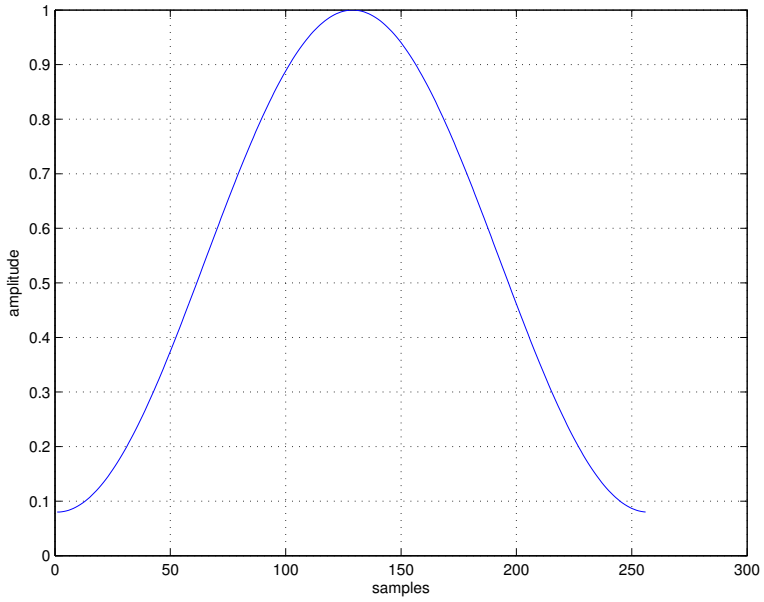


Figure 6.4: The hamming function

We calculated the magnitude of the STFT as $mag_i = \sqrt{(a_i)^2 + (b_i)^2}$, where a and b are the real and imaginary components of the fast Fourier transform for a given discrete frequency i.

The STFT produces a representation of the sound stream both in time and frequency domains that facilitates the application of the filter-bank in the frequency domain. Rabiner (Rabiner and Juang, 1993) proposes that the filter-bank can be implemented by varying adequately the frequency in the exponential term of (6.3); in the simplest case, this frequency has an uniform distribution choosing $f_i = i(F_s/N)$, where F_s is the sampling frequency. However, non-uniform frequency distribution can be used; in particular, neurophysiological studies propose numerous models of the human auditory system. One of those is the mel-frequency scale where the filter-banks are distributed linearly in the low frequencies and then decrease logarithmically in the higher frequencies. As suggested in (Slaney, 1998),

we constructed the filter-bank using 13 linearly-spaced filters (133.33 Hz between center frequencies) followed by 27 log-spaced filters (separated by a factor of 1.0711703 in frequency).

6.4.2 Mel-Frequency cepstral coefficients (MFCC)

The formula for the mel-frequency cepstral transform is as follows:

$$c_i = \frac{2}{N} \sum_{k=1}^{N} Y_k \cos[i(k+0.5)\frac{\pi}{N}], \quad i = 1, 2, \ldots, M \tag{6.5}$$

where c_i is the cepstral coefficient, and Y_k are the outputs of the filter-bank discussed in the previous section.

In our system, the MFCC transform reduces the dimensionality by transforming the output of 40 filter-banks into a compact representation of 13 cepstral coefficients. Figure 6.5 shows a graphical 3D representation of a MFCC transform applied to the sound produced by toy 6.2(a).

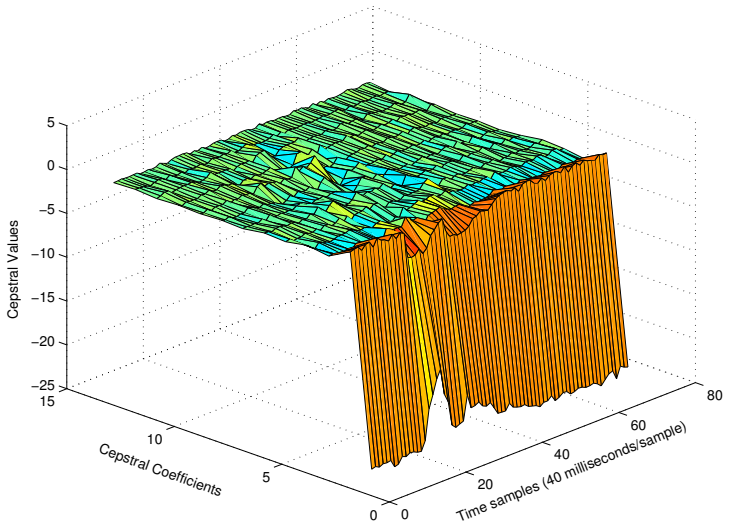


Figure 6.5: Three dimensional representation of a MFCC Transform

After applying equation (6.5) we packed the cepstral coefficient in the sound template data structure. This template contains the ceptral coefficients associated to a sound produced by a toy. To detect the presence of an object producing a sound, we measure empirically the background sound level and we use it as a threshold to activate the template recording procedure.

6.4.3 Delta-Delta Mel-Frequency coefficients

The simple MFCC transform does not include time evolution information. However, it would be interesting to obtain dynamic information from the sound signal. This information could be included in the feature set by cepstral derivatives. This information is not used in the current system but probably it is a way by which we could improve the robustness of the system. The first order derivative of cepstral coefficients is called Delta coefficients, and consequently the second order derivative of cepstral coefficients is called Delta-Delta coefficients. Delta coefficients tell us somehow the sound rate, and Delta-Delta coefficients describes the acceleration of the sound signal.

$$\Delta c_l(n;m) = \frac{1}{2}(c_l(n:m+1) - c_l(n;m-1)) \tag{6.6}$$

Delta-Delta coefficients can be calculated applying (6.6) to the Delta coefficients.

6.5 Multisensory object segmentation (Synaesthesis)

Once the sound is parameterized, the level of *synchrony* of the sound and visual data streams need to be measured. For this purpose, we use the method suggested by Hershey and Movellan based in the *mutual information* (Hershey and Movellan, 2000).

6.5.1 Mutual Information

Hershey and Movellan define the temporal synchronization of a video and sound channels as an estimate of the mutual information between both streams. Their algorithm was originally applied to the problem of finding a vocalizing person in a video sequence (Hershey and Movellan, 2000). They consider that $a(t) \in R^n$ is a vector describing the acoustic signal at time t and that $v(x,y,t) \in R^m$ is a vector describing the video signal at the same time instant. They assume that these vectors form a set S of audio-visual vectors and that these vectors are independent samples from a joint multivariate Gaussian process. Under these assumptions, Hershey and Movellan affirm that an estimate of the mutual information can be calculated as

$$\begin{aligned} I(A(t_k);V(x,y,t_k)) &= H(A(t_k)) + H(V(x,y,t_k)) - H(A(t_k),V(x,y,t_k)) \\ &= \frac{1}{2}\log(2\pi e)^n|\Sigma_A(t_k)| + \frac{1}{2}\log(2\pi e)^m|\Sigma_V(x,y,t_k)| \\ &- \frac{1}{2}\log(2\pi e)^{n+m}|\Sigma_{A,V}(x,y,t_k)| \\ &= \frac{1}{2}\log_2\frac{|\Sigma_A(t_k)||\Sigma_V(x,y,t_k)|}{|\Sigma_{A,V}(x,y,t_k)|} \end{aligned} \tag{6.7}$$

where $|\Sigma_A(t_k)|$ is the determinant of the covariance matrix of the audio stream, $|\Sigma_V(x, y, t_k)|$ is the determinant of the covariance matrix of a pixel of the image (e.g. the RGB values), and $|\Sigma_{A,V}(x, y, t_k)|$ is the joint covariance of both the audio and visual signals .

The calculation of (6.7) is based on the Shannon's statement (Shannon, 1948) that the entropy of a one-dimensional Gaussian distribution whose standard deviation is σ is given by $H(x) = \log\sqrt{2\pi e}\sigma$. This is explained in (Shannon, 1948) as follows. Given the gaussian probability distribution

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(\frac{x^2}{2\sigma^2})}$$

taking the logarithm on both sites of the equation we get

$$-\log p(x) = \log\sqrt{2\pi}\sigma + \frac{x^2}{2\sigma^2}.$$

Then, given the formula of the entropy

$$H(x) = -\int p(x)\log p(x)dx$$

and substituting $\log p(x)$ by the expression calculated above we get

$$\begin{aligned} H(x) &= \int p(x)\log\sqrt{2\pi}\sigma dx + \int p(x)\frac{x^2}{2\sigma^2}dx \\ &= \log\sqrt{2\pi}\sigma + \frac{\sigma^2}{2\sigma^2} \\ &= \log\sqrt{2\pi}\sigma + \log\sqrt{e} \\ &= \log\sqrt{2\pi e}\sigma \end{aligned} \tag{6.8}$$

To compute equation (6.7) different sound and images parametrizations can be used. In a first experiment, we calculated equation (6.7) using 13 cepstral coefficients (the parameters of covariance matrix $\Sigma_A(t_k)$) and three RGB values of the pixel (the parameters of the covariance matrix $\Sigma_V(x, y, t_k)$) during 0.6 seconds ($S = 15$). Consequently, the combined audio-vision covariance matrix $\Sigma_{A,V}(x, y, t_k)$ comprises 15x15 elements. The computation of the determinants of these matrices exhibits a considerable computational cost, because the determinants are calculated for each pixel in the image. This produces a considerable degradation of the system performance. Although this algorithm can be improved by having a distributed computation, we decided to use a simplified version of the mutual information as suggested by (Hershey and Movellan, 2000). This is the special case when the data streams are in a one dimensional representation (i.e. $n = m = 1$). Then, the mutual information can be expressed as

$$I(A(t_k); V(x, y, t_k)) = -\frac{1}{2}(1 - \rho^2(x, y, t_k)) \tag{6.9}$$

where $\rho^2(x, y, t_k)$ is the Pearson correlation coefficient between $A(t_k)$ and $V(x, y, t_k)$ (see (Weisstein, 1999) and (Press et al., 1992)). To obtain this one dimensional representation, we used for the sound the root mean square (RMS) of the short-time Fourier transform coefficients (see the arrow connection between the STFT box and the RMS box in figure 6.1) and a gray level value of the color RGB components calculated as grayvalue = $0.299R + 0.587G + 0.114B$. Notice that the MFCC transform was still used to form the sound template representation.

One may argue why we did not use the root mean square of the MFCC coefficients. The reasons were that we could easily compute the RMS value in the STFT module with little computational cost and that this early RMS computation made possible a parallel implementation of the object segmentation and sound parametrization modules.

Considering the data streams to be pairs of quantities (x_i, y_i), $i = 1, \ldots, N$, we calculated the Pearson correlation coefficient according to the formula:

$$\rho = \frac{\sum xy - \frac{1}{N}\sum x \sum y}{\sqrt{(\sum x^2 - \frac{1}{N}(\sum x)^2)(\sum y^2 - \frac{1}{N}(\sum y)^2)}} \tag{6.10}$$

where $\sum$ this time stands for sum.

6.5.2 The Mixelgram

To conceptualize the output of the mutual information between sound and vision, Prince et al. (Prince et al., 2004) introduced the *mixel*; that is a combination of the words **m**utual and p**ixel**. They proposed that the mixels form a topographic representation called *mixelgram*. These can form shapes that are perceptually relevant for human observers (Prince et al., 2004). Therefore, the mixelgram is to be considered a common space representation for both visual and audio sensorial channels.

Figure 6.6 depicts an example of the mixelgram of the toy 6.2(a). It is possible to distinguish the shape of the duck.

Algorithm 2 shows the steps we followed to calculate the mixelgram.

6.5.3 Improved object segmentation

The original image and the mixelgram maintain a direct geometric correspondence, therefore the mixelgram can be used to segment the object by segmenting the pixel in the original image which position corresponds to an activated mixel. However, the segmentation obtained with this method has a low quality because many pixels of the object are not segmented at all.

To improve the object segmentation, we use a technique based on color segmentation. We assume that the activated mixels belongs to a uniformly colored object. Then, we use a back-projection technique to improve the

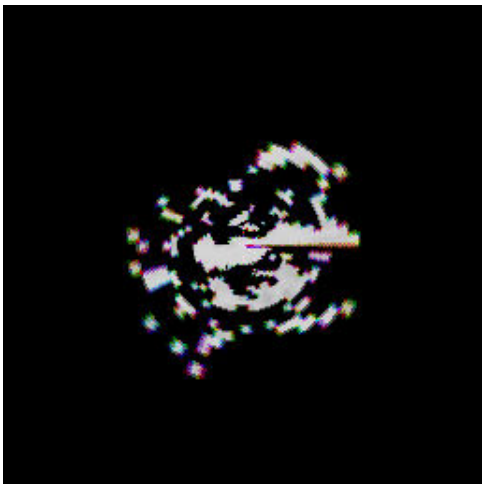


Figure 6.6: The mixelgram of the duck toy. Notice that the mixelgram inherits the same log-polar geometry used in the original image.

segmentation results. The pixels segmented with the mixelgram are used to create a HS (Hue-Saturation) histogram.

The HS histogram provides an idea of the object predominant color and this information can be used to segment by color the object in the original image using a back-projection algorithm. The pixels segmented using the back-projection technique are then combined with the pixels segmented by the mixelgram to create an improved segmentation of the object.

Our implementation is similar to that described in (Metta et al., 2004). However, in our system, the object is originally detected using the mutual information and we do not use a model of the background to segment the object. A detailed explanation of the technique has already been done in section 5.3.5.

As an example, figure 6.7 shows the HS histogram for the segmented object 6.2(b).

In figure 6.8 we present the results of the discussed segmentation process for the three toys used in the experiment. These were among the best segmentations obtained during the present experiment.

6.5.4 Associative memory

After an object is segmented, the segmentation results are stored in a dynamic lookup table. Each element in the lookup table contains the segmented image and the sound template associated to that object. To create the memory we appear the object in front of the robot several times squeezing or shaking the object with different speeds and strengths. This way, we produced slightly different sounds that were associated to the same object in the memory. This provided some robustness to the process of recognizing

Algorithm 2 Calculate mixelgram

```
loop
  Get Sound Stream
  Get Image
  Memorize sound and image streams
  for Each pixel do
    for Each recorded time stamp do
      Compute RMS sound stream value
      Transform RGB pixel into a grayscale pixel
    end for
    Compute Pearson correlation coefficient
    Compute mutual information
  end for
end loop
```

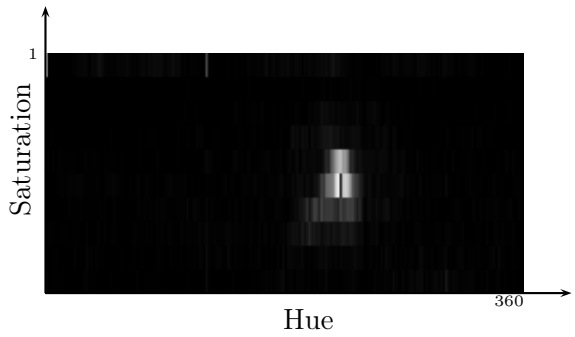


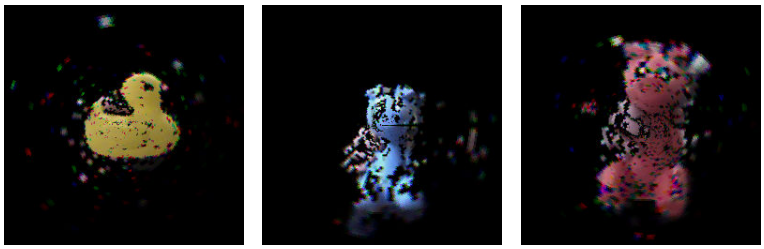
Figure 6.7: The resulting HS histogram of the segmented blue pig toy

the sound.

6.6 Attentional priming (Synesthesis)

This module performed basically a pattern classification for sound identification. When the system hears an unknown sound, the sound is parametrizated using the MFCC algorithm explained in section 6.4. Then, the sound template is compared with the memorized sound templates using a measure of similarity (distance).

To compare the sound templates it is necessary to compute both a local distance measure between the spectral vectors, and a global time alignment procedure (Rabiner and Juang, 1993). To compute the local distance, we used the truncated cepstral distance $d_c^2(L)$ (see (Rabiner and Juang, 1993)



(a) (b) (c)

Figure 6.8: The segmented toys

page 195) calculated as

$$\sum_{n=1}^{L}(c_n - c_n^{'})^2 \tag{6.11}$$

where c_n and $c_n^{'}$ are the cepstral coefficients of the stored and the heard sound templates respectively.

6.6.1 Dynamic Time Warping

The global time alignment procedure is necessary because the automatic sound recognition system has to take into account: a) time alignment and b) time normalization. This can be done using a Dynamic Time Warping (DTW) algorithm (Rabiner and Juang, 1993). We used the DTW algorithm to compare the heard sound to those stored in the associative memory discussed in previous section. During the experiment we produced these sounds outside the robot field of view. The system was able to recognize the sound and *remember* the object image associated with the sound. Then, the recovered toy image was presented to the experimenter for verification.

6.7 Results and Discussion

Table 6.1 presents the empirical results obtained during the presented experiment. In the case of the segmentation results, the table shows the percentage of segmentation trials with similar results of those presented in figure 6.8. Because a color segmentation is used, lighting conditions influence the segmentation. The results presented were obtained with good lighting conditions.

In the case of the sound recognition results, we did the experiment in a quiet laboratory environment with only some computer generating background noise. The results in both cases degraded significantly when we performed the experiment in noisy conditions, as for example, with people talking in the room.

Experiment	Duck	Blue pig	Red pig
Segmentation (Syn**a**esthesis)	64%	70%	75%
Classification (Synesthesis)	99%	88%	83%

Table 6.1: Segmentation and recognition results for the system

For the recognition module we used only the $c_1 \ldots c_{12}$ cepstral coefficients. The use of the c_0 cepstral coefficient degraded the capacity of the system to distinguish between similar objects. This was the case with the two pig toys that are made of the same material. This result make us suggest that the c_0 cepstral coefficient could be used to implement an algorithm to distinguish *classes* of objects. This may be convenient when the classification needs to be done among a big number of different sounds.

Chapter 7

Robvision: A research in mobile active robotics

Contents

7.1 Introduction

In this chapter I present the results of the more relevant project in which I have collaborated: Robvision(**ROB**ust **VI**sion for **S**ensing in **I**ndustrial **O**perations and **N**eeds). Robvision was a Esprit European Project with the ambitious goal of creating a mobile robot with an active vision system that could provide visual information about the environment. The visual information confronted with a Computer Assisted Design (CAD) model provided a precise position and orientation measurement for a mobile robot. The results of this project are relevant in several aspects for the present thesis: (i) the use of a memory (the CAD model) facilitates the process of perceiving the world providing robust information about the environment, (ii) the use of active vision provides fundamental information for the computation of distances and allows the concentration of attention on the most important features of the environment.

Though the CAD model data were filled by humans designers it is particularly relevant that the integration of this model—seen as a previous knowledge of the environment—and an active vision binocular head can solve a complex navigation and positional problem for a mobile robotic platform.

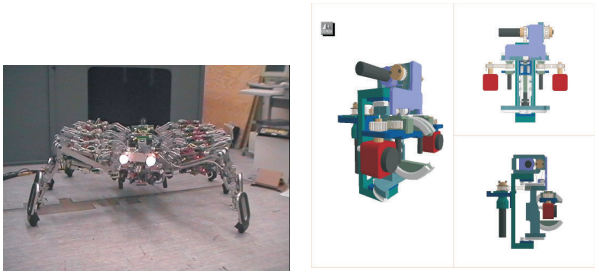
In certain extent, the CAD model can be seen as a memory that produces visual expectations. These expectations provide the robot with information about where to look at and the features that should be found corresponding to its current position and gaze direction. The errors detected between this *expected* visual information and the real scene are used by the robot to update the positional and orientation information.

This way of perceiving the world is coherent with the discussion presented in section 3.2. The way of generating the visual expectation is by using the memory (CAD model) and the belief about the actual position. The main conclusion of this chapter is that such perception increases the performance of the system.

The rest of the chapter is basically an outline of the two main publications related to the work done during the framework of Robvision (see (Gasteratos et al., 2002; Vincze et al., 2003)). The project involved the collaboration of several people from different research groups and industries, therefore the reader will find descriptions of the work developed by other teams. Though Robvision was a excellent result of team work with a hard integration phase, the results considered to be original from our research team can be found in section 7.3.2.

7.2 Robvision: An applied research project

Quality assurance and intelligent products are key roads to success in global competition. Supervising and automatically measuring the quality of parts in the production of large structures can reduce work costs. The inspection of these structures needs automated systems to position the tools necessary in a large environment (e.g. mobile robots). For both, quality measurement and positioning of inspection tools, the key is to generate the 3D pose of objects. To find this 3D pose a vision sensor system is developed. Since Industries are using CAD systems to design parts or working areas this model knowledge is applied to initialize the vision process. The CAD system provides features to the vision module that tries to find these features in the images. Two alternative image-processing techniques are implemented, a monocular and a binocular, to take advantage of the arising redundancy. The emphasis of using two different vision methods and CAD model data is to enhance the robustness of the vision process to make the overall system reliable. To ensure correct feature detection, model and image cues (e.g. geometrical order of features or feature intensity) are integrated (Vincze, 2001) (Vincze et al., 1999). After feature extraction, the pose estimation algorithm can use both 3D and 2D feature information found in the images to calculate the current pose and send it to the robot. One of the industrial project partner constructed the 8-legged walking robot used during the project (see Figure 7.1).



(a) (b)

Figure 7.1: (a) the 8-legged pneumatic walking and climbing robot; (b) the Stereo-head

7.3 Sytem Overview

Figure 7.2 depicts the entire system architecture. It consists of several modules colored in different greyscales. Each subsystem is provided by one of the project partners and fulfills all the functions drawn inside the according module. The next subsections give an overview of the modules and their respective functions.

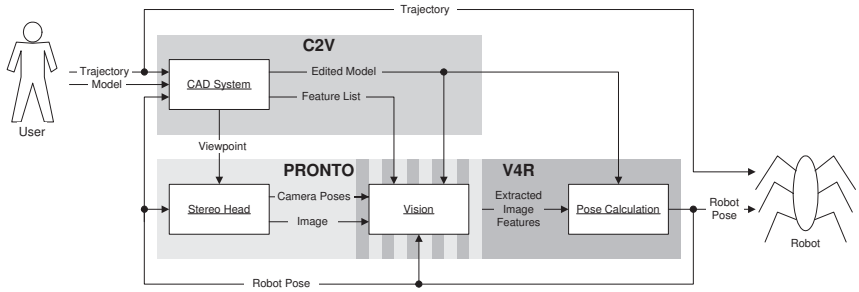


Figure 7.2: System Overview of the RobVision Subsystems

7.3.1 C2V

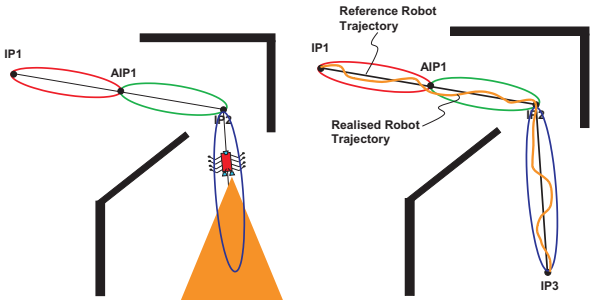
C2V (Cad to Vision), developed by the Department of Production, Aalborg University, provides the CAD system. The main task is to generate geometrical features for the vision system. These features specify the shape and location of geometrical entities such as lines, junctions and regions that the cameras can expect to see while the robot is moving inside the structure.

As it appears from Figure 7.2 the first input of the CAD system is a *CAD model* that is the geometric model of the structure inside which the robot has to move. As a second input, a *Reference Robot Trajectory* is delivered

containing a specification of the task or trajectory that the robot has to perform. The user generates this trajectory off-line by using a trajectory planner developed by OSS. The last input are *Robot Poses* computed by the pose calculation component using the features found by the vision system. The main output from the CAD system are model and view data. The *Model data* contain a specification of relevant information from the underlying geometrical model, such as the feature topology. The *Camera Viewpoint* specifies a 3D point (x,y,z) in world co-ordinates towards which the stereo head ought to point. The *Feature List* contains a set of robust features that the vision systems can expect to find when looking at the specified viewpoint.

The CAD-model and the reference robot trajectory are computed prior to the operation of the RobVision system. In this off-line phase of C2V (C2VoffLine), good viewpoints along the referece robot trajectory are determined and the features associated with these views are derived and generated. This approach divides the trajectory into areas where sets of features can be used to determinate the robot pose (Figure 7.3 a).

In the online phase of C2V (C2VonLine) the estimated robot pose generated by V4R is used to identify in which of the areas generated by the off-line system the robot is presently located (Figure 7.3 b). The according viewpoint and the features associated with this area are sent to the vision systems. This approach requires that the actual deviations of the robot trajectory are less than the areas generated by C2VoffLine. Simulations show that deviation of 500 mm or more, depending on the situation, are acceptable.



(a) (b)

Figure 7.3: The (a) Reference- and the (b) Realised- Robot Trajectory. The ellipses denote areas along the trajectory where a constant set of features is visible.

7.3.2 PRONTO

PRONTO developed by Laboratory for Integrated Advanced Robotics, University of Genoa, is the software/hardware module that is responsible for

the stereo head task which in brief consists of *Head control, Head stabilization, Head calibration* and the *acquisition of 3D feature data*. For each of the 3D edge junctions PRONTO applies a stereovision algorithm to measure the feature depths. A Hough technique is implemented to extract the lines belonging to the junction. A weighted LMS (Least Mean Square) method is used to relate them to the features provided by the CAD system (Gasteratos and Sandini, July 2000). Then a closed loop method is followed, so that by moving simultaneously the three degrees of freedom of the head the junction is put at the principal point of the image in both images. When this is the case the two cameras are verging on the certain junction and the direct kinematics of the head are applied, in order to determine the 3D position of the junction relative to the head.

After this high precision depth measurements, the vision algorithm based on tracking (V4R) is used and PRONTO switches to its second operation mode, where it concentrates its computational efforts to stabilize the head and to keep the gaze of the cameras fixated on the viewpoint recommended by C2V. The robot pose generated by V4R is used every 240 ms to calculate the required movements of the head. The stabilization is performed using angular accelerometers that react to the movements of the robot. This consents PRONTO to calculate the angular velocity by integrating the acceleration in a 40 ms cycle. This velocity is then multiplied by a predefined factor and directly introduced in the head motors producing a compensation movement. Both, stabilization and gaze orientation have the purpose to guarantee a good image quality to increase the robustness of the tracking algorithm described in the next section.

7.3.3 V4R

V4R (Vision for Robotics), developed by Institute of Flexible Automation, Vienna University of Technology consists of two functions. The first task of V4R is the monocular 2D feature search and tracking. The second unit of V4R is the pose calculation component to calculate the pose of the robot relative to a reference system.

The emphasis of the vision method is basically to provide robust features in real-time, i.e., keeping the processing time lower than the frame rate of the camera (40ms). To handle the real-time constraint a windowing method is applied to limit the processing time (Hager and Toyama, 1998). Robustness can be achieved by including image and model information - so-called cues - into the vision process. The method used is a combination of cue integration strategies (with cues like intensity values, color, texture) and the RANSAC-method for Edge finding (Fischler and Bolles, 1981) (Vincze et al., 1999). V4R is presently able to track edge features like lines, junctions, ellipses and arcs. In near future regions will be included.

For the first search of features the projection of the model into the image

indicates an approximate position of the features. Since the model contains the model of the features and their topological relations to each other, checking of the actual geometric relations like, e.g., connectedness of some features can assure that the right feature is found. Furthermore some other model cues like color information of the object can be used to eliminate wrong feature candidates. Once a feature is found for the first time, this feature is then tracked in the next cycle. Additional information of the features found in the image (image cues like intensity) can be stored to facilitate the search for the same feature in the next tracking cycle. An example for a tracking sequence is given in Figure 7.4.

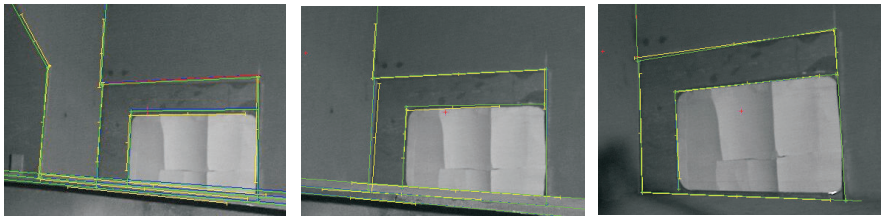


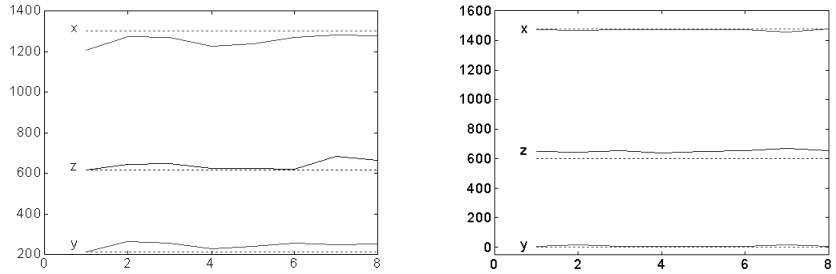
Figure 7.4: A tracking sequence along the robot trajectory.

The second unit of V4R is the pose calculation which computes the position and orientation of the robot related to the environment object, i.e. the ship. This is done by fitting the 2D and 3D image features extracted to the according model received from the CAD system. The algorithm employed is based on a method proposed by Wunsch (Wunsch, 1997) and was modified for providing accuracy estimation of the calculated pose as well as outlier detection (Kraus, 1997).

7.4 Results

This section gives an overview of the tests for evaluating the whole system. A mock-up was built with structures that can be found in a big ship environment. A big challenge is to find a large number of "good" views to enable a successful task completion, that is, views containing a large number of robust features. The importance hereby is not only to generate good views along the path but also good alternative views for one robot pose to be on the secure side if pose determination with the first view fails. Malfunction can happen since the mock-up is made of big metal plates welded together, which is a big defiance for the vision. Some welding causes irregular edge features, contrast is sometimes poor and additional features on the plate surfaces can affect pose calculation results negatively. That's why feature redundancy is a basic issue. The CAD system generates the features seen in the image to deliver it to the vision. The two vision methods are able to extract the 3D

coordinates of the junctions (Gasteratos et al., 2000) and the 2D positions of edges and junctions in the image. From all the features extracted a pose is calculated (Wunsch, 1997).



(a) (b)

Figure 7.5: Robot pose measurements in a fix pose (a) diagonal, (b) parallel

Two kind of tests were performed. The first group had the goal to *mesure the accuracy of the system at different fix poses* within the mock-up. With a ruler a reference position was measured. Figure 7.5 shows the results of different trials to calculate the position in two poses. The first one (Figure 7.5 a) corresponds to the robot standing diagonal in the mock-up and the second one (Figure 7.5 b) corresponds to the robot standing parallel. Table 7.1 and Table 7.2 summarise the measurements and give the standard deviations and the mean deviation from the reference measurements. The mean values are relatively high due to consistent off-set, which might be caused by the difficulty to access the origin of the measurement coordiante system. A further uncertainty is the actual geometry of the mock-up. The bending of the plates causes a few centimetres deviation over the model. As can be seen, the three dimensional standard deviation is in one case 35.72 mm and in the other case 4.64 mm, measurements that fulfil the specification for a ship building aplication.

	X[mm]	Y[mm]	Z[mm]	roll[deg]	pitch[deg]	yaw[deg]	3D pos[mm]
std	28.17	16.91	23.56	0.35	0	0	35.72
mean	45.87	33.87	25	2.13	0	0	71.91

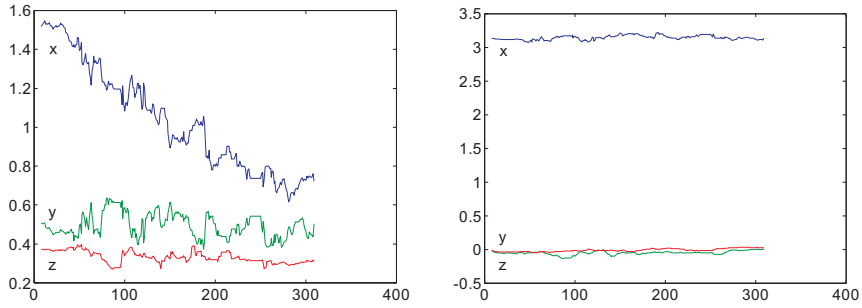
Table 7.1: The standard deviation (std) of the measurements and the mean distance from the reference.

The second group of tests had the goal to test the *performance and reliability* of the pose calculation with the robot walking along a trajectory. Figure 7.6 gives an overview of the pose calculation recorded along a path in the x-direction. As can be seen the pneumatic walking robot produces

	X[mm]	Y[mm]	Z[mm]	roll[deg]	pitch[deg]	yaw[deg]	3D pos[mm]
std	5.49	3.29	8.48	0	0.92	0.35	4.64
mean	4.88	8.62	51.25	0	0.62	0.13	52.49

Table 7.2: The standard deviation (std) of the measurements and the mean distance from the reference.

many jerks, in particular for the translational degrees of freedom. It can be observed that the body of the robot sometimes moves up 15 cm when releasing a leg. These fast changes in the orientation of the robot cause big deviations of the feature positions in the image. Because of the fast but spatially restricted windowing technique, used in one of the vision methods, such features are then lost. This fact reduces the reliability of the system. The camera head stabilization implemented lower the optical flow and increases the robustness of feature finding.



(a) (b)

Figure 7.6: Robot pose output along a trajectory, (a) position, (b) orientation

The tests executed show the system benefits from the redundancies implemented. As long as enough features can be tracked, the system is able to re-find lost features. Also wrong detected image features are filtered out from the pose calculation. The stabilization of the pose increases by the use of 3D features and the continuous update of new image features from the CAD database makes the overall system robust and reliable.

Chapter 8

Conclusions

The biological basis of prediction are not well know. I have shown, however, that the comparison between predictions and real information, that is the *prediction error*, seems to play a fundamental role in many brain processes. Moreover, prediction seems to play an important role in motor control.

I have addressed also the effect of prediction in the present, particularly how visual expectations and active vision can improve perceptual processes in complex robotic systems. In this respect, it is specially relevant the study about the effect of audio-visual crossmodal expectations and how they can affect attention.

Yet, prediction remains a complex issue that needs to be approached through multidisciplinary research. This thesis aims to be a contribution in understanding prediction and its fundamental role in robots that learn and develop. That is why I have entitled this thesis *Toward Predictive Robotics*, because I try to demonstrate that prediction deserves more attention and can be a key issue in future robotic technology. I hope that, at least, I have succeed in this goal.

Appendix A

QR Factorization for efficient covariance determinant computation

A.1 Introduction

In the chapter 6 I have addressed the problem of segmenting an object by combining both visual and audio cues. The focus is in detecting the temporal correlation in both sensorial cues by using a statistic called mutual information. The calculation of the mutual information implies the computation of the covariance matrix determinants for the audio and visual data streams and for the combinations of both streams—this is the joint covariance matrix. This operations must be done for each pixel in the image with a considerable computational cost in spite of the system using log-polar images with smaller dimension than standard (rectangular) images. In this appendix I discuss in more detail the techniques used to perform such operations.

A.2 The covariance matrix

Usually the variance, in the discrete case, is calculated as:

$$\sigma^2 = \sum_{i=1}^{n} P(x_i)(x_i - \mu)^2 \tag{A.1}$$

where $P(x_i)$ is a probability distribution, μ is the mean, and x_i are the samples of the random variable.

Frequently, in real data experiments, the probability distribution is not

known and therefore is acceptable to calculate the variance using the form:

$$S_n^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x}) \tag{A.2}$$

S_n^2 is known as the sample variance and $\bar{x}$ is known as the sample mean calculated as $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$. In order to obtain an unbiased estimator for σ^2 it is possible to use the next version of the sample variance:

$$S_{n-1}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x}) \tag{A.3}$$

As a practical example consider that you have to compute the covariance matrix of the RGB values of a pixel during a given period of time, lets say during six time samples. In such a case you can form the next matrix with the data vectors:

$$X = \begin{pmatrix} 200 & 200 & 200 \\ 123 & 234 & 56 \\ 43 & 232 & 20 \\ 33 & 200 & 50 \\ 100 & 140 & 100 \\ 140 & 200 & 160 \end{pmatrix}$$

Each row (e.g. 200 200 200) correspond to an observation of the RGB values of the pixel in a given instant of time. Therefore is trivial to calculate the means vectors that results in:

$$\bar{\mathbf{x}} = \begin{pmatrix} 106.5000 & 201.0000 & 97.6667 \end{pmatrix}$$

By considering the variances matrix as $V = (X - \bar{X})$ then it is possible to calculate the covariance matrix by applying the next formula:

$$S = \frac{1}{n-1}(V^T V)$$

A.3 Fast computation of the covariance matrix determinant

In many practical applications may be interesting, however, not to form the covariance matrix S directly particularly if it is necessary to calculate the determinant of the covariance matrix. Eventually we can represent the covariance matrix, S, as:

$$S = A^T A \tag{A.4}$$

where the matrix A can be calculated from the variances matrix V—seen in the previous section—by making $A = V \cdot \frac{1}{\sqrt{n-1}}$. Then, instead of computing

S, the matrix A can be decomposed with a QR factorization and it can be substituted in (A.4) as follows:

$$S = R^T Q^T Q R \tag{A.5}$$

Because Q is orthogonal, then $Q^T Q = 1$ and consequently, (A.5) can be reduced to:

$$S = R^T R \tag{A.6}$$

Then the determinant of the covariance matrix can be computed applying (A.7).

$$\det(S) = \det(R)^2 \tag{A.7}$$

The $\det(R)$ can be trivially calculated because R is an upper triangular matrix and therefore its determinant is equal to the product of the diagonal elements.

Appendix B

Log-polar images

Vision is the sense more used by humans. In this appendix we describe a mathematical model, known as log-polar representation, that approximates the way humans may perceive the *light* information.

B.1 Log-Polar Images

The photoreceptors within the human retina exhibit a space-variant distribution. The cones ,which are responsible for visual perception of light, have a higher density at the center of the visual field and are sparser in the periphery. The resolution has a radial symmetry, which can be approximated by a polar distribution. The projection of the photoreceptor array into the primary visual cortex can be well described by a logarithmic-polar (log-polar) distribution mapped onto a rectangular-like surface.

This particular sensor distribution has important consequences in the attentional processes. The fact that the fovea generates the most informative flow of information determines the importance of moving this fovea towards the object of interest. Indeed, this is the case in the animal kingdom where frequently foveal vision is associated with *active* capacities.

The other important fact is that with such geometric distribution of visual receptors it is possible to perceive a wider field of view, moreover, the photoreceptors of the periphery are more sensitive to changes in illumination, hence more appropriate for motion detection.

In summary, the main advantage of such visual perception architecture is the fact that it is possible to maintain both a good visual acuity and wide field of view maintaining a low size of the resulting image. This can have important consequences both from a computational point of view and for transmission and storage purposes.

Log-polar images find therefore inspiration in biology and have been used frequently in robotics. Particularly, in active systems. It is worth mention-

ing that artificial sensors with log-polar geometry have been developed in silicon C-MOS support (Sandini and Metta, 2002) and they are particularly useful in real-time robotic applications (Bernardino and Santos-Victor, 1996. IROS'96). This type of geometry has been using widely during this thesis though the log-polar images have been used through a software mapping from standard (rectangular image) cameras.

From the mathematical point of view, the log-polar mapping can be expressed as a transformation between a polar plane (ρ, θ) (retinal plane) and a cartesian plane (ξ, η) (log-polar or cortical plane), as follows:

$$\begin{cases} \eta & = q\theta \\ \xi & = K_\xi \ln_a \frac{\rho}{\rho_0} \end{cases} \tag{B.1}$$

where ρ_0 is the radius of the innermost circle, $1/q$ is the minimum angular resolution of the log-polar layout, and (ρ, θ) are the polar coordinates. K_ξ is a linear scaling parameter, this has been added to the original formulation in order to fit the mapping into a fixed size squared image (which is determined by the frame grabber characteristic). These are related to the conventional Cartesian reference system by:

$$\begin{cases} x = \rho \cos\theta \\ y = \rho \sin\theta \end{cases} \tag{B.2}$$

Graphically the mapping is represented in figure B.1.

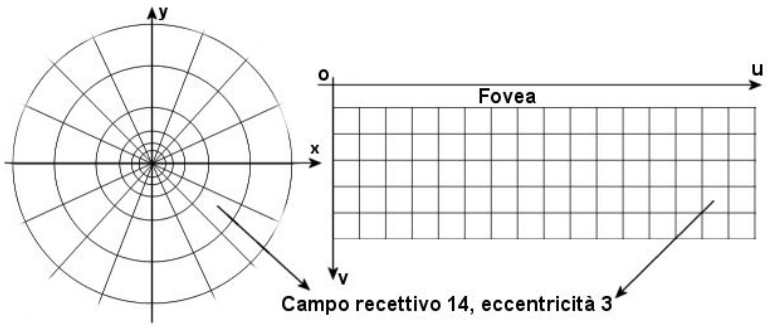
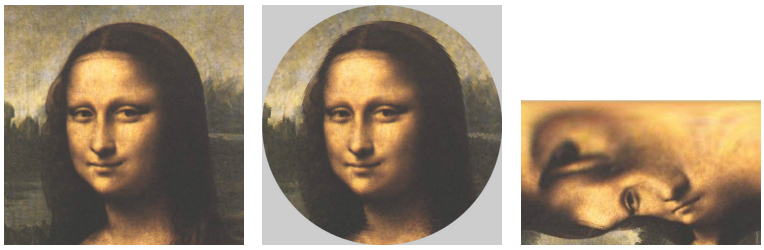


Figure B.1: The mapping can be explained as follows: The original image is divided in concentric circles which are uniformly sampled and arranged along the rows of the logpolar image. The outermost and innermost circles are placed in the last and first rows respectively.

The result of the log-polar mapping in a real image can be seen in figure B.2.



(a) (b) (c)

Figure B.2: (a) the original rectangular image, (b) the concentric circles samples, (c) the log-polar image

Bibliography

Aloimonos, J.Y., Weiss, I., and Bandopadhay, A. (1987). Active vision. *International Journal on Computer Vision*, pages pp. 333–356.

Arsenio, A. (2004). *Cognitive-Developmental Learning for a Humanoid Robot: A Caregiver's Gift*. Ph.D. thesis, Massachusetts Institute of Technology.

Arsenio, A. and Fitzpatrick, P. (2003). Exploiting cross-modal rhythm for robot perception of objects. In *2nd International Conference on Computational Intelligence, Robotics, and Autonomous Systems*, pages p 15–18. Singapore.

Bahill, A. Terry and McDonald, Jack D. (1983). Smooth pursuit eye movements in response to predictable target motions. *Vision Research*, 23(12):1573–1583.

Ballard, D. (1991). Animate vision. *Artificial Intelligence*, 48(1):pp. 1–27.

Barnes, G.R. (1993). Visual-vestibular interaction in the control of head and eye movement: The role of visual feedback and predictive mechanisms. *Progress in Neurobilogy*, 41:435 – 472.

Baroni, Alberto (2002). *Inseguimento adattativo binoculare in tempo reale*. Master's thesis, University of Genova.

Barto, Andrew G., Fagg, Andrew H., Stikoff, Nathan, and Houk, J.C. (1999). A cerebellar model of timing and prediction n the control of reaching. *Neural Computation*, 11:565–594.

Bellman, R. E. (1957). *Dynamic Programming*. Princeton University Press.

Beltrán-González, Carlos (2002). *Bttvx. A QNX driver for BT8xx based framegrabbers*.

Bernardino, A. and Santos-Victor, J. (1996. IROS'96). Vergence control for robotic heads using log-polar images. In *International Conference on Intelligent Robots and Systems*.

Berthoz, Alain (1997). *The Brain's Sense of Movement*. Perspectives in Cognitive Neuroscience. Hardward University Press.

Blake, A. and Yuille, A. (1992). *Active Vision*. Artificial intelligence. Massachusetts Institute of Technology.

Bradshaw, K.J., Reid, I.D., and Murray, D.W. (1997). The active recovery of 3d motion trajectories and their use in prediction. *IEEE Transactions on Pattern Analisys and Machine Intelligence*, 19(3):219 – 233.

Brown, C.H. (1990). Gaze control with interaction and delays. *IEEE Transactions on Systems, Man and Cybernetics*, 20(1):518–527.

Butz, Martin, Sigaud, Olivier, and Gerard, Pierre, editors (2003). *Anticipatory Behavior in Adaptive Learning Systems. Foundations, Theories, and Systems*. Springer.

Coombs, D. (1992). *Real-Time Gaze Holding in Binocular Robot Vision*. Phd, University of Ronchester.

Datteri, Edoardo, Teti, Giancarlo, Laschi, Cecilia, Guglielmo, Tamburri., Dario, P., and Gunglielmelli, E. (2003). Expected perception in robots: a biologically driven perception-action scheme. In *International Conference on Advanced Robotics*. Coimbra, Portugal.

Doya, K., Kimura, H., and Miyamura, A. (2001). Motor control: Neural models and system theory. *International Journal of Applied Mathematics and Computer Science*, 11:101–128.

Fischler, M.A. and Bolles, R.C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In *Communications of the ACM*, volume 24, pages pp.381–395.

Fitzpatrick, Paul, Metta, Giorgio, Natale, Lorenzo, Rao, Sajit, and Sandini, Giulio (2003). Learning about objects through action - initial steps towards artificial cognition. In *Proceedings of the 2003 IEEE International Conference on Robotics & Automation*.

Flanagan, J.Randall, Vetter, Phillipp, Johansson, Roland S., and Wolpert, D.M. (2003). Prediction precedes control in motor learning. *Current Biology*, 13:146–150.

Gasteratos, A., Martinotti, R., Metta, G., and Sandini, G. (2000). Precise 3d measurements with a high resolution stereo head. In *IWISPA 2000*, pages pp,171–176. Pula, Croatia.

Gasteratos, A. and Sandini, G. (July 2000). On the accuracy of the eurohead. Lira-lab technical report, LIRA - TR 2/00.

Gasteratos, Antonios, **Carlos Beltran**, Metta, Giorgio, and Sandini, Giulio (2002). Pronto: A system for mobile robot navigation via cad-model guidance. *Microprocessors and Microsystems*, 26.

Hager, G. and Toyama, K. (1998). The xvision-system: A portable substrate for real-time vision applications. In *Computer Vision and Image Understanding*, volume 69, pages 23–37.

Hershey, J. and Movellan, J. (2000). Audio-vision: Using audiovisual synchrony to locate sounds. *Advances in Neural Infromation Processing Systems*, 12.

Houk, J.C., Buckingham, J.T., and Barto, A.G. (1996). Models of the cerebellum and motor learning. *Behavioral and Brain Sciences*, 19(3):368–383.

Houk, J.C. and Miller, L.E. (2001). *Encyclopedia of Life Sciences*, chapter Cerebellum: Movement Regulation and Cognitive Functions.

Johnson, M.H. (1997). Vision, orienting, and attention. In *Developmental Cognitive Neuroscience*, Fundamentals of Cognitive Neuroscience, pages 69–231. Blackwell, Cambridge, Mass.

Jordan, Michael I. (1996). Computational aspects of motor control and motor learning. In H. Heuer and S. Keele, editors, *Handbook of Perception and Action: Motor Skills*. Academic Press, New York.

Jordan, Michael I. (1999). Computational motor control. In M Gazzaniga, editor, *The Cognitive Neurosciences*. MIT Press, Cambridge (MA).

Kalman, R.E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:35 – 45.

Kawato, M. and Gomi, H. (1992). The cerebellum and vor/okr learning models. *Trends in Neuroscience*, 15:445 – 452.

Keeler, J.D. (1990). A dynamical system view of cerebellar function. *Physica*, pages 396 – 410.

Kettner, R.E., Mahamud, S., Leung, H.C., Sitkoff, N., Houk, J.C., and Peterson, B.W. (1997). Prediction of complex two-dimensional trajectories by a cerebellar model of smooth pursuing eye movement. *The American Physiological Society*.

Klam, F., Petit, J., Grantyn, A., and Berthoz, A. (2000). Predictive elements in ocular interception and tracking of a moving target by untrained cats. *Experimental Brain Research*, 139:233–247.

Kraus, K. (1997). Advanced methods and applications. In *Photogrammetry Volume 2*, pages pp.205–215. Dümmler/Bonn.

Landy, M.S., Maloney, L.T., and Pavel, M. (1996). *Exploratory Vision: The Active Eye*. Springer Series in Perception Engineering. Springer.

Leigh, R. John and Zee, David S. (1999). *The neurology of eye movements*. Oxford University Press.

Lungarella, Max, Metta, Giorgio, Pfeifer, Rolf, and Sandini, Giulio (2004). Developmental robotics: a survey. *Connection Science*.

Marr, D. (1982). *Vision: a computational investigation into the human representation and processing of visual information*. W. H. Freeman, San Francisco.

Maybeck, P.S. (1979). Introduction(stochastic models, estimation, and control). In Academic Press, editor, *Stochastic models, estimation, and control*, volume 1. Academic Press.

Mehta, Biren and Schaal, Stefan (2002). Forward models in visuomotor control. *J. Neurophysiol*, 88:942–953.

Metta, Fitzpatrick, and al. (). *Yet Another Robotic Platform (YARP). http://yarp0.sourceforge.net/*.

Metta, G. and Fitzpatrick, P. (2003). Better vision through manipulation. *Adaptive Behavior*, 11(2):109–128.

Metta, Giorgio (1999). *Babyrobot: A Study on Sensori-motor development*. Ph.D. thesis, University of Genova.

Metta, Giorgio, Gasteratos, Antonios, and Sandini, Giulio (2004). Learning to track colored objects with log-polar vision. *Mechatronics*, 14:989–1006.

Miall, R.C., Malmus, M., and Roberson, E.M. (1996). Sensory prediction as a role for the cerebellum. *Behavioral and Brain Sciences*, 19:466.

Miall, R.C and Reckess, G.Z. (2002). The cerebellum and thetiming of coordinated eye and hand tracking. *Brain and Cognition*, 48:212–226.

Miall, R.C., Weir, D.J., Wolpert, D.M., and Stein, J.F. (1993). Is the cerebellum a smith predictor? *Journal of Motor Behavior*, 25(3):203 – 216.

Möller, R. (1997). Perception through anticipation - an approach to behaviour-based perception. In *Proc. of the New Trends in Cognitive Science*, pages pp. 184–190. Viena, Austria.

Mussa-Ivaldi, F.A. (1997). Nonlinear force fields: A distributed system of control primitives for representing and learning movementents. In IEEE, editor, *IEEE International Symposium on Computational Intelligence in Robotics and Automation*. IEEE.

Natale, L., Metta, G., and Sandini, G. (2002). Development of auditory-evoked reflexes: Visuo-acoustic cues integration in a binocular head. *Robotics and Autonomous Systems*, 39(2):pp. 87–106.

Natale, Lorenzo (2004). *Linking Action to Perception in a Humanoid Robot: A Developmental Approach to Grasping*. Ph.D. thesis, University of Genova.

Neisser, Ulric (1976). *Cognition and Reality. Principles and implications of cognitive psychology*. W.H. Freeman and Company.

Panerai, F., Metta, G., and Sandini, G. (2000). Learning VOR-iike stabilization reflexes in robots.

Parkins, E.J. (1997). Cerebellum and cerebrum in adaptive control and cognition: a review. *Biological Cybernetics*, 77:79–87.

Pavel, M. (1990). Predictive control of eye movements. In E Kowler, editor, *Eye movements and their role in visual and cognitive processes*, pages 71 – 113. Elsevier Science.

Peters, Jan and Smagt, Patrick (2002). Searching a scalable approach to cerebellar based control. *Applied Intelligent*, 17:11–33.

Press, William H., Flannery, Brian P., Teukolsky, Saul A., and Vetterling, William T. (1992). *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press, 2 edition edition.

Prince, Christopher G., Hollich, George J., Helder, Nathan A., Mislivec, Eric J., Reddy, Anoop, Salunke, Sampanna, and Memon, Naveed (2004). Taking synchrony seriously: A perceptual-level model of infant synchrony detection. In *Proceedings of the Fourth International Workshop on Epigenetic Robotics*.

Rabiner, Lawrence and Juang, Biing-Hwang (1993). *Fundamentals of Speech Recognition*. Prentice Hall Signal Processing Series. Prentice Hall.

Rich, Anina N. and B.Mattingley, Jason (2002). Anomalous perception in synaesthesia: A cognitive neuroscience perspective. *Nature Reviews | Neuroscience*.

Robinson, D.A. (1987). Why visuomotor systems don't like negative feedback and how thet avoid it? In M Arbib and A Hanson, editors, *Vision, Brain and Cooperative Computations*, pages 89–107. MIT Press.

Rougeaux, S., Kita, N., Kuniyoshi, Y., Sakane, S., and Chavand, F. (1994, IROS'94). Binocular tracking based on virtual horopter. In *International Conference on Intelligent Robots and Systems*. Munich, Germany.

Sandini, Giulio and Metta, Giorgio (2002). Retina- like sensors: motivations, technology and applications. *In Sensors and Sensing in Biology and Engineering*.

Schultz, Wolfram and Dickinson, Anthony (2000). Neuronal coding of prediction errors. *Annual Review of Neuroscience*, 23:473–500.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27:pp. 379–423, 623–656.

Shibata, T. and Schaal, S. (2001, IROS01). Biomimetic smooth pursuit based on fast learning of the target dynamics. In *International Conference on Intelligent Robots and Systems*. Outrigger Wailea Resort, Maui, Hawaii, USA.

Slaney, Malcolm (1998). Auditory toolbox. version 2. Technical report, Interval Research Corproation.

Smagt, P. (2000). Benchmarking cerebellar control. *Robotics and Autonomous Systems*, 32:237–251.

Smagt, Patric and Bullock, Daniel, editors (1997). *Can Artificial Cerebellar Models Compete to Control Robots*, volume 1. German Aerospace Center.

Smagt, Patrick (1998). Cerebellar control of robots arms. *Connection Science*, 10:301–320.

Smith, Russell L. (1998). *Intelligent Motion Control with an Artificial Cerebellum*. Ph.D. thesis, University of Auckland.

Stäger, Mathias, Lukowicz, Paul, Perera, Niroshan, von Büren, Thomas, Tröster, Gerhard, and Starner, Thad (2003). Soundbutton: Design of a low power wearable audio classification system. In *ISWC 2003: Proceedings of the 7th IEEE International Symposium on Wearable Computers*, pages 12–17.

van Heijst, J.J. (1998). *Self-organization in neural networks as models for the development of motor control*. Ph.D. thesis, University of Groningen.

Vieville, Thierry (1997). *A Few Steps Towards 3D Active Vision*. Springer.

Vincze, Ayromlou, Ponweiser, **Beltrán**, and Gasteratos (2003). A system to navigate a robot into a ship structure. *Machine Vision and Applications*, 14(1).

Vincze, M. (2001). Robust tracking of ellipses at frame rate. In *Pattern Recognition*, volume 34, pages 487–498.

Vincze, M., Ayromlou, M., and Kubinger, W. (1999). An integrating framework for robust real-time 3d object tracking. In *Int. Conf. on Vision Systems*, pages 135–150. Gran Canaria, Spain.

Von Hofsten, C. and Rosander, K. (1996). The development of gaze control and predictive tracking in young infants. *Vision Research*, 36(1):81 – 96.

Von Hofsten, C. and Rosander, K. (1997). Development of smooth pursuit tracking in young infants. *Vision Research*, 37(13):1799–1810.

Weisstein, Eric W. (1999). Correlation coefficient. From MathWorld–A Wolfram Web Resource.

Wexler, Mark and Klam, François (2001). Movement prediction and movement production. *Journal of Experimental Psychology: Human Perception and Performance*, 27(1):pp.48–64.

Witney, A. G. and Wolpert, D.M. (2003). Spatial representation of predictive motor learning. *Journal of Neurophysiology*, 89:1837–1843.

Wolpert, D.M. (1997). Computational approaches to motor control. *TRENDS in Cognitive Sciences*, 1(6):209–216.

Wolpert, D.M., Doya, Kenji, and Kawato, Mitsuo (2003). A unifying computational framework for motor control and social interaction. *Philosofical Transactions Royal Society of London*, 358:593–602.

Wolpert, D.M. and Flanagan, J.Randall (2001). Motor prediction. *Current Biology*, 11(18):729–732.

Wolpert, D.M., Ghahramani, Zoubin, and Flanagan, J.Randall (2001). Perspectives and problems in motor learning. *TRENDS in Cognitive Sciences*, 5(11):487–493.

Wolpert, D.M. and Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks*.

Wolpert, D.M., Miall, R.C., and Kawato, K. (1998). Internal models in the cerebellum. *TRENDS in Cognitive Sciences*, 2(9):338–347.

Wunsch, P. (1997). *Modellbasierte 3-D Objektlageschätzung für visuell geregelte Greifvorgänge in der Robotik*. Phd thesis, Munich University of Technology.