

Baby_{robot} A Study on Sensori-motor Development

Giorgio Metta

LIRA-Lab, DIST, University of Genova



This work has been carried out by Giorgio Metta, during his Ph.D. course in Computer Sciences, under the supervision of Prof. Giulio Sandini at LIRA-Lab, Department of Telecommunication, Computer and System Sciences, University of Genova, Italy. © 1996-1999 LIRA-Lab

The research described in this book has been supported by grants from the Italian Ministry of Research and University (MURST), the European Union (projects: VIRGO, SMART, SVAVISCIA, NARVAL, ROBVISION) and by the Italian Space Agency (ASI).

All rights reserved. No part of this book may be reproduced, in any form or by any means, without the permission in writing from the authors.

Printed in Italy.

Copyright notice:
LIRA-Lab, DIST, University of Genova, Italy, © 1996-1999 LIRA-Lab
URL: <http://www.lira.dist.unige.it>
URL: <http://pasa.lira.dist.unige.it>

Considerate la vostra semenza:
fatti non foste a viver come bruti,
ma per seguir virtute e canoscenza.

Dante, Inferno XXVI, vv. 118-120

Acknowledgements

I should have mentioned here all the people who made this work possible. I was rather afraid to forget somebody and, in any case, it would have been a long, long list. Consequently, whether you have been working at LIRA-Lab or we have been in touch between 1997 and 1999, you should feel as being part of this. Among all the people, I would like to thank, in particular, Prof. Giulio Sandini who, being of constant support, helped in creating the nice environment around the newborn Babybot, and Dr. Nick Barnes who helped in the revision of the manuscript.

Moreover, I wish to thank my wife Sabrina who has been so patient, considering the “extra” time I spent trying to make things work. I should not forget, my family and hers, for the constant “moral” support. Finally, I would like to dedicate this book to my grandmother ☺.

Giorgio Metta

Abstract

The research presented in this manuscript stems from an interdisciplinary approach ranging from “brain sciences” to robotics. The goal is to answer two main questions, namely:

- Is it possible to test hypotheses on the brain function involved in a particular task, by implementing biologically plausible models on a real physical system such as a robot?
- Is there any way to design more adaptable and potentially efficient robots?

In order to provide a sensible reply, we have studied “sensori-motor development” as a model of learning and adaptation from a neuroscience and robotics perspective. In this respect, we shall argue that development is not simply a mere summation of learning stages. We rather consider it as the process which governs the time-varying nature of the learning agent itself. Consequently, on one hand we believe that the analysis of developmental processes provides unique insights on how sensori-motor coordination arises in biological learners, on the other it could be the only feasible procedure to design highly complicated artificial systems. From the modeling point of view, we shall demonstrate how a twelve degrees of freedom “baby” humanoid robot (Babybot) acquires orienting and reaching behaviors, and investigate the advantages of the proposed framework over traditional learning paradigms. More specifically, artificial development is based on the observation that biological development of visuo-motor coordination follows several stages, starting from a “plant” mostly driven by reflexes, and steering through phases where the cortex begins to influence sub-cortical structures. In the artificial implementation, we shall show that reflexes can be seen as the building blocks, guiding the learning of more sophisticated behaviors, and acting as a bootstrap procedure for the whole system. Moreover, they can also serve to keep subsequent learning processes within feasible regions of the “state space”. It will be shown, through the implementation of an artificial system, that it is possible: i) to outline the mechanisms relevant for visuo-motor control, ii) to implement realistic models of sensori-motor learning iii) to demonstrate the important role of development in building adaptable systems operating in real environments.

Contents

Acknowledgements	5
Abstract	7
Contents.....	9
1 Introduction.....	11
1.1 A few words about this manuscript.....	14
1.2 Robotics and biology	15
1.3 The “baby” humanoid robot	17
1.4 Complexity and the “telephone switchboard model”	18
1.5 Development of gaze control and reaching	22
1.5.1 Development of eye-head coordination	23
1.5.2 Development of reaching	26
1.5.3 Relevant issues for artificial development	30
1.6 Seeing, reaching and touching.....	31
1.6.1 Phases and components	32
1.6.2 The big show after learning.....	33
1.7 Discussion	34
2 The learning problem.....	39
2.1 Bias and variance	42
2.2 Exploration versus exploitation.....	44
2.3 Stability and catastrophic forgetting	47
2.4 Dimensionality: the big curse	48
2.5 Is the environment interaction necessary?	49
2.6 Do we need to “perceive” in order to control our movements?...	51
2.7 Noise is everywhere.....	52
2.8 Discussion	53
3 A developing robot.....	55
3.1 The noisy initial configuration.....	56
3.2 The cortical take-over process	60
3.3 Controlling more degrees of freedom, eye-head coordination....	63
3.3.1 What the maps look like	75
3.4 Tuning the Vestibulo-Ocular Reflex.....	80
3.4.1 Visuo-inertial integration: the simple way.....	82
3.4.2 Tuning VOR: the hard way.....	85
3.5 Saccade and VOR interaction	90
3.6 Reaching.....	94

3.7	Reaching: the idea and beyond	97
3.7.1	Initialization of the motor-motor map	101
3.7.2	Trajectory generation.....	101
3.7.3	The learning procedure.....	103
3.7.4	Biologically plausible trajectory profiles	107
3.8	Improving reaching by employing more DOF.....	107
3.8.1	Reducing input size.....	108
3.8.2	Reducing output size.....	109
3.9	Discussion	113
4	Conclusions.....	117
5	Appendix.....	119
5.1	Log-polar images	119
5.2	Optical flow	121
5.2.1	An alternative approach based on the RF concept.....	123
5.3	Stabilization index.....	125
5.4	Color segmentation.....	125
5.5	The learning module: a growing neural network	126
5.6	The force field approach to motor control	131
5.6.1	From the EP to the activation values	135
5.7	The technological corner-shop	137
	Bibliography	141
	Table of illustrations.....	153
	Index	165

1 Introduction

The study presented in this manuscript stems from an interdisciplinary approach covering aspects of “brain sciences” and robotics. The goal is to answer two main questions, namely:

- Is it possible to test hypotheses on the brain functions involved in a particular task, by implementing biologically plausible models on a real physical system such as a robot?
- How can we design more adaptable and potentially efficient robots? Is it possible to build a truly human-like robot?

Of course, given these premises and the difficulties in answering such questions, not all the aspects will be equally detailed. The hope is, at least, to provide useful hints on the many topics involved.

In fact, we believe that there might be commonalities, sometimes due to the nature of the tasks, sometimes to the physics itself, which suggest that both artificial and biological agents could consistently employ the same solutions. The study of the biology – the modeling of brain functions – could suggest how to build more successful and adaptable “artificial beings”. On the other hand, the quest for adaptation raises the issue of learning; in other words, how can the learner acquire useful information in order to accomplish a given task? Which sensors does it need? Is learning always feasible? Until now, robotics and AI have failed to give a definitive answer (assuming it does exist) and indirectly they have also failed to produce truly autonomous and flexible agents. In spite of many successes in building robots of various shape, size, abilities, sensory types, etc. there seems to be something lacking in terms of “cognitive abilities”, as well as adaptability of the system to the dynamic of the environment. Moreover, even for successful robots, the integration of different behaviors and sensory modalities gave rise to a series of unexpected problems. The traditional artificial learning paradigm faced such difficulties, perhaps because of some wrong assumptions about the learning process itself, rather than the lack of proper models and algorithms.

In recent times “brain sciences” also face an increasingly and intricate picture, where it is hard to discover the general underlying principles,

which eventually will bear the real explanatory power. The panorama consists of a huge number of brain areas and intricate interconnections between them (the so-called “telephone switchboard” model). It is not the case that all parts are fully interconnected (i.e. the connection density is about 30%¹), but a general explanatory principle is still missing. Here lies the significance of modeling, many researchers have employed computational models to explain functions, to derive general rules, and to integrate data gathered by using different methodologies. Perhaps, not everything is suitable for representation in a model, and the world – the “external” environment – is probably something which is too complex to be replicated appropriately. For this role, a robot is a suitable tool to be employed in the field of computational neuroscience.

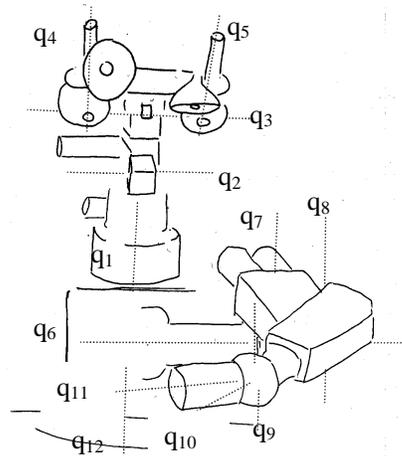
Eventually roboticists have realized that there is something to learn from biology – though the approach was often too pragmatic – and neuroscientists have approached robotics with growing interest. If AI researchers were less pragmatic, we might have realized that biological learners are characterized by a peculiar sequence of learning stages, where control structures radically change over time, and new abilities progressively arise. Moreover, pushing this further, there might be the chance to cast new light on the design of learning agents, specifically, by adopting a correct design paradigm and the correct assumptions.

In this context, “brain scientists” have studied, since a long time, the acquisition of behaviors and cognitive abilities, and nobody is surprised by the fact that newborns are not simply a sort of “reduced size human beings”. What is more surprising is that, from an early age, infants show a series of “innate” behaviors, basic control synergies, and reflexes. On this basis, more sophisticated behaviors develop, and this process progresses through stages, where the limited abilities already formed are efficiently exploited in order to simplify the learning process itself. On the contrary, the approach followed in robotics is mainly that of designing the “complete final product” (i.e. the adult robot). One might wonder: what is wrong with that? Perhaps, something was underestimated, and from a purely engineering point of view, this “something” was the whole process of design. Dennet (Dennet, 1997), for instance, thinks that the overall design process must be included in the specifications of the final product. This approach shifts the emphasis from the final product to the developmental process thus the goal of the designer becomes that of devising a suitable

¹ This figure concerns the visual areas of the macaque brain – (Van Essen & Deyoe, 1995).

initial state (at time t_0), and the appropriate developmental rules to get some close approximation of the desired final product.

Box 1 The experimental setup. The experimental setup consists of a five degrees of freedom robot head (designed and realized at LIRA-Lab), and an off-the-shelf six degrees of freedom robot manipulator (an Unimation Puma260), both mounted on a rotating base: i.e. the torso. The kinematics resembles that of the upper part of the human body although with less degrees of freedom. From the sensory point of view, the Babybot is equipped with two space-variant cameras (Sandini & Tagliasco, 1980), (Sandini, Braccini, Gambardella, & Tagliasco, 1981), microphones for acoustic localization, an inertial sensor simulating the vestibular system (Panerai & Sandini, 1998), and proprioceptive information through motor encoders. The robot is controlled by a set of PCs – ranging from Pentium II to Pentium III processors – each running Windows NT and connected by a fast Ethernet link. In order to provide the necessary interface with the hardware (i.e. sensors and motors) some machines are equipped with motion control boards, frame grabbers, AD converters, etc. In particular, one machine controls the robot arm and the torso, another one the head, and a third computer performs the visual processing. The software adheres to DCOM, a standard, which allows running objects among the various machines. The Babybot kinematics is shown on the right panel of the picture below. The dashed lines indicate joint's axes numbered from q_1 to q_{12} respectively.



However, posed in this way, our goal questions would cover a rather broad scope, ranging from cognition, motor control, sensory processing, etc. In order to narrow this range, our investigation was focused on sensori-motor coordination tasks, although we took care not oversimplify either the physics of the system or the problems under scrutiny. As a general principle, the “physical system” should have enough degrees of freedom (e.g. in some part it might even be redundant), and a proper sensorial stimulation. The important point, rather than having an exceptional sensory and motor system, is that of keeping all sub-parts reasonably balanced, so that they can possibly develop all together (see also Box 1).

1.1 A few words about this manuscript

The general organizing principle of the book is that of presenting things in a sort of reverse order. The first chapter partially covers some of the experimental results obtained after the robot learnt simple coordination tasks; namely i) orienting the gaze toward visually identified targets, ii) reaching the fixation point in order to touch the acquired visual target. The rest of the manuscript will try to explain, demonstrate, and show how and why the robot is able to learn and what simplification the aforementioned “developmental” approach provides.

Furthermore, the manuscript presents experimental results and related biological facts. They are inserted as boxes; the main concepts, on the other hand, are treated in standard text sections.

The remainder of this chapter, besides introducing our experimental setup in section 1.3, makes the point that we indeed deal with complex systems (section 1.4) and, in order to justify the “developmental approach”, presents some of the aspects related to development in biological systems (section 1.5). We shall comment briefly those relevant for the design of complex systems.

Chapter 2 introduces the learning problem, and points out that developing systems show a more effective adaptive behavior, compared to “traditional artificial learners”. We shall describe some well-known problems, such as, the bias-variance dilemma, or the “curse of dimensionality”, and we will conjecture that “developing systems” may be equipped to cope with these “theoretical pressures”. Through some examples, we will argue that this is actually the case, and we will make references to the corresponding “biological” solutions.

Chapter 3 describes the Babybot’s architecture in details; it also shows where biology meets robotics and where it does not. It presents many of the details concerning the robot design, its controllers, and the

experimental results. In particular, a full range of experimental results will cover the aspects related to gaze control and reaching – this is our reference task – in both biology and robotics.

Conclusions, future work, comments, etc. are contained in chapter 4. Finally, mathematical details and algorithms are presented in the appendix 5. We describe the visual algorithms (optic flow and color segmentation), the learning models, and finally a torque based motion control, known as “force fields” approach.

The next section presents a brief overview of research at the boundary between biology and robotics.

1.2 Robotics and biology

Research activity linking studies on artificial systems to “brain sciences” is certainly not new. Besides the studies on artificial neural networks, substantial efforts are devoted worldwide to build physical models of parts of biological systems, with the aim of suggesting novel solutions to robotics or processing problems, and to advance our understanding of human brain functions (Brooks, 1996), (Sandini, 1997). For example, the control of robot heads and visually guided manipulation tasks were studied with reference to psychophysical performance data of humans and animals (Aloimonos, Weiss, & Bandyopadhyay, 1988), (Bajcsy, 1985), (Ballard & Brown, 1992), (Crowley, Bobet, & Mesrabi, 1992), (Capurro, Panerai, Grosso, & Sandini, 1993), (Gandolfo, Sandini, & Tistarelli, 1991), (Grosso, Manzotti, Tiso, & Sandini, 1995), (Grosso, Metta, Oddera, & Sandini, 1996). In this respect, many levels of similarity to biological systems can be considered: from emulation to a vague resemblance. The important point is to grasp the relevant aspects of biological systems so that we can both address specific biological questions and propose new methodologies for robotics.

However, the main advantage of using robots rather than pure computer simulations, at least in the study of the motor system, is that the physics of the environment comes “for free” – a proper simulation would be very difficult, if not impossible. The concept of embodiment further supports this view or, at least, it makes more significant to situate the “brain” into a real physical body (such as a robot) (Pfeifer & Scheier, 1998), (Pfeifer & Scheier, 1997).

A very active area of research has been, for instance, the study of locomotion, either strict bipedal locomotion (Brooks, 1989), (Hirai, Hirose, Haikawa, & Takenaka, 1998) or swimming (Vaidyanathan, Chiel, & Quinn, 1997). Also insect-like robots have been proposed (for a review see

(Beer, Chiel, Quinn, & Ritzmann, 1998)). A few realizations used muscle-like actuators either pneumatic- or polymeric-based (Majarrad & Shahinpoor, 1997), (Caldwell, Medrano-Cerda, & Bowler, 1997).

The study of computational motor control, involves a series of techniques borrowed from traditional system theory. Some of these have been applied to the study of limb movements (e.g. optimization techniques), in order to fit psychophysical data to models (Jordan, 1996). An influential model, derived from biology, is the so-called “equilibrium point model” (EP), which has direct applicability to robotics (Mussa-Ivaldi & Giszter, 1992), (Mussa-Ivaldi, Giszter, & Bizzi, 1993), (Mussa-Ivaldi, 1997), (Mussa-Ivaldi & Bizzi, 1989), (Gomi & Kawato, 1997) – see also section 5.6 in this manuscript.

Williamson (Williamson, 1996) applied this approach for controlling the arm movements of COG – the humanoid robot being built at MIT. Metta and coworkers used a similar biologically inspired approach to control the movements of the LIRA-Lab Babybot (described in this book) (Metta, Sandini, & Konczak, 1999).

Another area which attracted considerable attention is the study of orienting behavior and ocular movements. Many implementations arose, using visual, acoustic, and inertial sensory systems (Crowley et al., 1992), (Capurro, Panerai, & Sandini, 1995), (Berthouze, Bakker, & Kuniyoshi, 1996), (Panerai, Metta, & Sandini, 2000). Rucci and colleagues (Rucci, Wray, Tononi, & Edelman, 1997) modeled the orienting behavior after that of the barn owl, with a particular emphasis on the visuo-auditory plasticity (Stein & Meredith, 1993). Also, Kuniyoshi is using a multi-cue approach in his humanoid robot controller (Kuniyoshi & Cheng, 1999). Panerai et al. used inertial sensors to simulate the vestibular organs, and employed both visual and inertial information inside the control loop of a binocular robot head. They demonstrated superior performance compared to a purely vision based controller (Panerai & Sandini, 1998).

On the other hand, in spite of all these efforts, few researchers addressed the problem of adaptive behavior from a developmental point of view (Pfeifer & Scheier, 1997), (Kuniyoshi & Cheng, 1999), (Berthouze & Kuniyoshi, 1998), (Sandini, Metta, & Konczak, 1997). In our view, in spite of such advances, the systems implemented are still far from achieving human-like performance levels and task flexibility. More importantly, the integration of different behaviors, such as manipulation and gaze control, proved to be more difficult than expected. This difficulty arises, at least in part, from the approach followed to construct complex systems: to make the problem more tractable, sensori-motor coordination is broken down

into a set of sub-problems defined by a specific sensory modality (e.g. vision, audition, touch, etc) or specific motor skills (e.g. manipulation, gaze control, navigation).

A different solution is used in humans and many other vertebrates, where flexible and efficient levels of performance are achieved through the simultaneous development of sensory, motor, and cognitive abilities. This process is not simply caused by the maturation of single components or by learning a progressively more sophisticated set of skills. Instead, it is marked, particularly in the very early stages, by a sequence of changes of the neural circuitry, and by a strategic exploitation of the environment with a limited set of motor skills that are present at each developmental stage. Finally, biological systems calibrate themselves in the presence of ongoing environmental and internal changes.

1.3 The “baby” humanoid robot

The “baby humanoid robot” is neither particularly small nor it grows in the standard meaning of the term – the physical structure is invariant over time. Conversely, the control strategies, modules, and their subparts evolve in the sense discussed below: the idea is to have a system, which at “birth” uses only simple controllers, such as reflexes, and successively “grows” by employing more sophisticated modules – i.e. the so-called internal models. The overall system’s initial state is thus characterized by a very limited number of free parameters, which can be easily estimated online. Concurrent controllers learn on the basis of how the reflex-like subsystems behave. The general principle is that of mimicking a “developmental process” where control is initiated from the very beginning of the agent’s life and, although imprecise, it constitutes a sort of “bootstrap procedure”. Noise, simulating defective command generation (muscle control), drives the exploration of the state space. Exploration and exploitation are carried out in parallel; the robot performs system identification and control at the same time. When some degrees of freedom are under a consistent control, the robot can start moving more joints. This progression is not simply due to our particular design choices, but is necessary in order to acquire proper behavior, and it is indeed exploited also by biological agents. The rationale behind this approach is that it should be simpler to design the system at “time t_0 ” and its “developmental rules” rather than the complete system. However, it is not clear at the moment whether it would be simpler to predict the final outcome.

Is this approach feasible? In order to remove any doubt, in this chapter, we present some of the results obtained after the robot acquired

orienting and reaching behavior. The results mainly indicate that the approach is at least feasible; further experiments will eventually compare these results with those derived from different learning methodologies. In particular, the experimental setup consists of the twelve degrees of freedom (DOF) anthropomorphic robot shown in Box 1, which is composed by a five DOF robot head entirely designed and realized at LIRA-Lab, an off-the-shelf six DOF manipulator (Unimation Puma 260), and a one degree of freedom torso. Two color cameras are mounted on the robot head. The head can move them independently around a vertical axis, and together around a horizontal one. Two more degrees of freedom in the neck allow the whole head to pan and tilt. The Puma 260 manipulator has six degrees of freedom. Both the arm and the head are mounted on a rotating base, which realizes a simple motion of the torso.

In order to understand the real complexity of controlling such a system, just consider how many state variables, input and output need to be examined, analyzed, and coordinated. As mentioned above, the sensory system consists of two color cameras, an inertial sensor (angular gyroscopic accelerometer) simulating the vestibular apparatus, and optical encoders, which provide proprioception (position and speed of each joint). From the purely system theory point of view, the physics of the robot can be described by 24 state variables – position and speed of each joint. However, from the learning perspective, the “state” space can be much larger. The robot must find out both the right time-sequence of motor commands from any possible starting configuration, and how to convert (or map) different signals back and forth from sensory to motor, from motor to sensory, and eventually from motor commands to predicted sensorial effects. These aspects of the learning problem will be analyzed in some of the following chapters.

1.4 Complexity and the “telephone switchboard model”

We introduce here the complete block control diagram of the robot (see Figure 1). It is worth noting that, though complicated, it could have been designed by hand – i.e. by tuning all controllers’ gain and mappings – although it would have been hard work. Consider that, Figure 1 sketches what is required in order to perform a relatively simple task: gazing and reaching. We did not address problems such as trajectory generation, obstacle avoidance, planning, and object persistency, memory, and so forth. Our robot behaves in a very primitive way: it is substantially stimulus bound – a stimulus appearance causes the robot’s reaction. This is to say that complexity and adaptability cannot be simply designed using the

same methodology and tools we might use to design, let's say a car. Perhaps, we need something else if we want to design a real “adaptive² agent”, admitting for the moment it can be designed. The point we would like to make is that at a first glance, it is not possible to infer behavior from the wiring itself. This adds something to the intrinsic difficulty: i.e. can we predict what the behavior of the whole system will be³? Can we design everything by hand – can we design adaptation?

Figure 2 shows something similar to Figure 1; it is a well-known block diagram representing the visual processing areas of the macaque brain. A question arises: can we understand perception and behavior just by observing the wiring of brain areas? Is further knowledge required? Consider also that the more we get into the details of the wiring, the more intricate the connections become. This is to say, even if we had a perfect understanding of the whole schematics, probably we would need some general purpose unifying principles. Van Essen (Van Essen, Anderson, & Olshausen, 1992), (Van Essen & Deyoe, 1995) tried to address the problem by extracting commonalities about the pathways and connections in terms of forward and feedback signals, and by characterizing accurately the different neural responses. Therefore, we consider that developmental studies might provide a different and complementary perspective.

Generally speaking, developmental studies allow starting with a simpler system. Considering motor control, for example the initial controller configuration is certainly simpler than the adult counterpart. Firstly, cortical influence is rather limited (and perhaps unreliable). Secondly, the behavior is very much stimulus bound: i.e. the appearance of a stimulus causes the infant to react – although it is not often easy to have the baby collaborate in the experiments. This is a much simpler situation compared to adults, where many control pathways are in place and contribute to the final outcome (behavior). Moreover, the study of development allows observing when and how the different subsystems come into play – for instance, dramatic changes in the control structure

² We are not claiming that adaptation has to be designed, what we should be able to design is the adaptive agent. Adaptive behavior, on the other hand, is perhaps acquired through the interaction with the environment.

³ The main goal of a design process is the comprehension and the capacity of making predictions about the final behavior of the designed product. For instance, in software engineering there is the need to have complete control over CPU resources, a prediction of response delays, memory usage, etc. Hence, real-time operative systems were designed aiming at the satisfaction of such performance requirements. The question can be restated as: to what extent can we make these predictions?

have been observed at particular stages (see section 1.5.2). In addition, the timing of development becomes an important factor, for the same reasons outlined above.

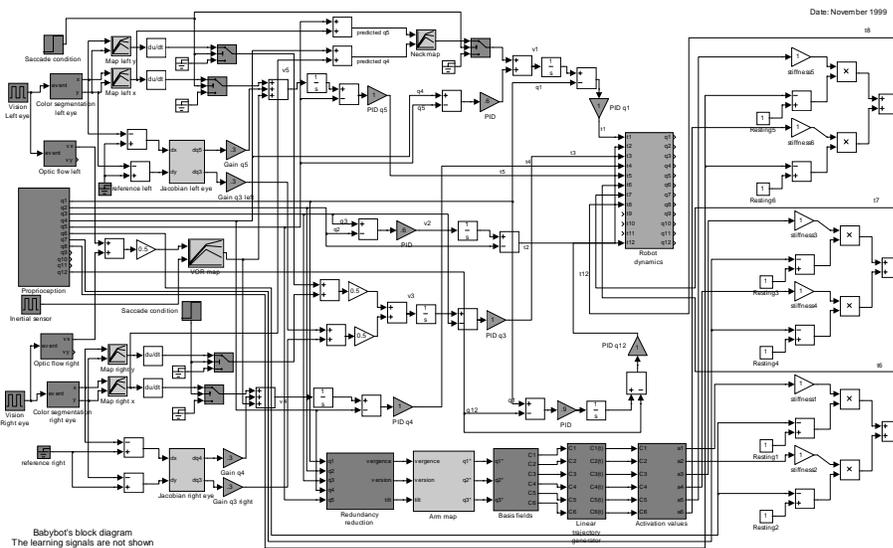


Figure 1 The pattern of interconnections in the Babybot. This schematic is the block diagram of Babybot's controller. The light gray blocks are those where learning and adaptation take place. Not all areas are active from the beginning. The medium gray central block is the robot mechanical plant – i.e. the system dynamics. Dark blocks are processing related areas either visual or motor. Triangles are gain of PD controllers. Many of them are tuned beforehand.

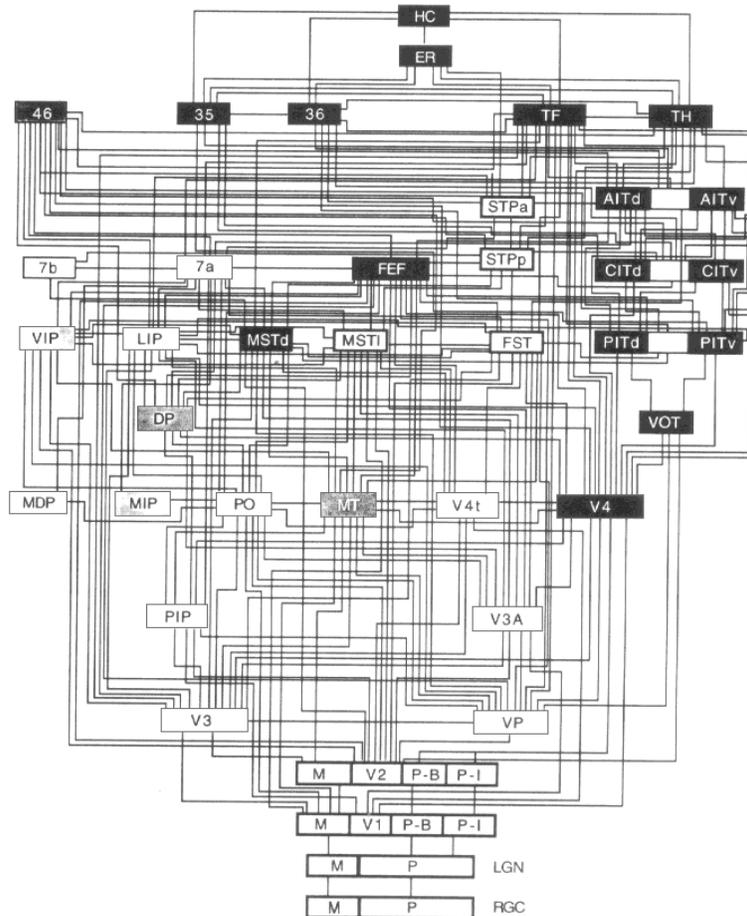


Figure 2 The pattern of interconnections in the macaque (adapted from (Van Essen & Deyoe, 1995)). Van Essen and coworkers pointed out that visual processing in primates involves dozens of different areas, and both forward and backward connections, with the former, perhaps, carrying out the processing per-se, and the latter type mostly implementing a sort of “flow control” structure – this view is oversimplified anyway. Connections are both hierarchical and concurrent, so processing is both serial and parallel at the same time.

As a metaphor, let us imagine we want to know how a car works: there are several reasonable approaches such as driving it, taking it apart, reading the manual, and going to the factory and seeing how it is put together. The “driving” approach takes a behaviorist point of view, though putting together neural responses and behavior is not an easy task – this approach substantially neglects the “inside”, the brain becomes a “black box”. The “taking apart” might indeed work: it bears resemblance with the connection and wiring analysis – it would include all the current techniques such as electro-physiology, PET scanning, fMRI, localized lesions, etc. “Reading the manual”... actually we do not have it, though one might think that DNA provides the ultimate manual. It is again hard work, even harder, to make behavioral inferences starting from the molecular level (reductionism). Further, consider that behavior is the result of complex interactions of the environment and endogenous processes – that is, it is partly phylogenic and partly ontogenic. Consequently, these complex interactions should be taken into account for a real comprehension of the whole “brain” issue. The last, approach is the “going to the factory”: in this case, we go into the details of how the system is put together. Again, there might be different feasible levels of analysis: from molecular – how genetic factors influence the final product – to neuron and population level. The latter level of analysis is suitable for approximate simulation and/or implementation on a physical robot, and it is the main topic of this book.

1.5 Development of gaze control and reaching

This section outlines some relevant developmental issues related to motor control, in particular, we focus on the acquisition of gazing and reaching (i.e. eye-head-arm coordination). Though this is not meant to be an exhaustive survey of the biological literature, it deals with the main aspects of development. We make the point that development is not the same as learning. In this respect, we see development as the process, which allows learning to take place. Roughly speaking, from now on, learning concerns the acquisition of a particular skill, either a map or a control parameter, while development defines the sequence of acquisition of such basic skills. In other words, development dictates which, among the infinite sequences of possible learning events, is realized. Bearing this in mind, we would like to mention a series of biological related findings, which further support the hypothesis that this is indeed the case.

1.5.1 Development of eye-head coordination

In the context of eye-head coordination, many authors have suggested the hypothesis that newborn's motor acts are controlled only by sub-cortical structures, with cortical control taking over at about two to three months (see (Atkinson, 1998)). Other models consider different visual functions (e.g. "what and where" streams (Goodale, 1989), (Milner & Goodale, 1995)) as developing with slightly different timing. Overall, the issue of differential timing in development can be seen in the context of learning as reducing the exploration⁴ space by constraining it. In fact, sub-cortical structures, active since birth, could actually guide the learning of cortical ones. Another way of looking at these facts is to consider the newborn at birth. Sub-cortical sensori-motor pathways are active and allow the baby to start his/her interaction with the external world. Meanwhile, development of cortical areas proceeds, but clearly the training data which can be collected from the environment are constrained by how the sub-cortical system is behaving (and by what the environment is generating – i.e. a deprived agent cannot learn).

The sub-cortical arrangement found in newborns can be seen as the initial "bias". Considering a broader perspective, the differential activation of brain areas is likely to provide each of them with a sort of bias: areas that develop first bias areas that develop later. Carpenter (Carpenter, 1988), (Carpenter, 1999) and Atkinson (Atkinson, 1998) described a few examples of the mechanism in the context of saccade generation and smooth pursuit control. This subdivision is not entirely clear, because it has been demonstrated that sometimes the cortex influences the sub-cortical pathways from the very beginning (although unreliably). There are instances where even parts of the same stream, or between different sub-cortical streams, fail to integrate.

There are some more experiments worth mentioning, which support the idea of the two systems (cortical and sub-cortical) developing with different time spans. It appears that 1-month-old newborns are often unable to disengage from an already fixated target. This is called "sticky fixation" and it is probably evidence that the sub-cortical system has the control at that age. Later on (at about 3 to 4 months of age), cortical modules (some authors suggested those located in the posterior parietal cortex, the frontal eye fields and the prefrontal cortex) develop and allow babies to disengage rapidly (Gilmore & Johnson, 1998).

⁴ This concept will be defined and detailed in chapter 2.

A complete picture should also take into account development of the visual processing system, which through time improves discrimination capabilities as well as perceived spatial resolution. In fact, newborns show poor visual acuity and proprioceptive feedback (see discussion on next section). This should be considered an advantage rather than a drawback. Poor resolution means that the number of free parameters that need to be controlled (or tuned) could be small. This is particularly true in the view of recent proposals on constructive learning (see (Quartz & Sejnowski, 1997), (Schaal & Atkeson, 1998)). Some kind of neural networks (Barron, 1993), (Fritzke, 1995) could actually take advantage of the initial low resolution to improve learning performance. Intuitively, it is easier to tune few parameters rather than many, even if in the long term the presence of a high number of tunable variables allows better learning.

There is further evidence that newborns plan saccades in a retino-centric coordinate system. This has been taken sometimes as the hallmark of the Superior Colliculus activity. Gilmore and Johnson have tested the hypothesis (Gilmore & Johnson, 1998). They found that 4-month-old infants fail in a “double saccade paradigm test”, while 6-month-old do not. The hypothesis here is that areas, which process spatial information in head- or body-centered coordinates, do not develop until 6-8 months of age. Even in this case, simplicity seems to dominate at the beginning, while more complex information processing needs more time to become functional.

Also the smooth pursuit system is at birth driven only by a sub-cortical circuitry. In fact, some asymmetries of the OKN in infants have been correlated to the functioning of the NOT (Nucleus of the Optic Tract). Later, improvement of motion detection takes place, supersedes, and complements functioning of the NOTs. This is supposed to involve development of the projections from the cortex to the NOTs, as well as the development of the magno cellular stream (sensitive to velocity stimuli).

Von Hofsten showed also that evolution of gaze control shifts, from a stage where head motion is used a little, to successive stages where neck and eye movements are combined effectively (Von Hofsten & Rosander, 1997). The reason for this pattern of development might be twofold: first, neck and eyes form a kinematically redundant system so that, coupling (i.e. by limiting the number of controlled degrees of freedom) the controllers could be an effective learning strategy. Second, as discussed above, coordinated motion of both head and eyes requires proprioceptive (or efferent copied) information, which is thought to be unreliable at birth.

Consider also that one of the major goals of the mobility of the eyes is that of positioning the high-resolution foveae at perceived interesting regions in the external world. Nevertheless, in order to obtain reliable visual information, it is likely that the more stable the images the better such information. The stabilization process is accomplished by various subsystems where both visual and vestibular information is used and combined (Sauvan, 1998). For instance, vision deals with low frequency perturbations, while the vestibular system works effectively at high frequencies (Miles, 1997), (Miles, Kawano, & Optican, 1986). The Vestibulo-Ocular Reflex (VOR) is thought to be present at birth, however, its proper use probably requires some gain adjustments (Panerai & Sandini, 1998), (Medendorp, Bakker, Van Gisbergen, & Gielen, 1999). This means that, even in this case, the development of the cortex shapes and improves the sub-cortical system functionality (i.e. by tuning the VOR gain as a function of the target distance and eccentricity).

In spite of these mechanisms, it would be difficult for a learner to gather enough information without exploring its state/control space. One natural mechanism, which might serve this function, is “noise”. It is also thought that the nervous system is deliberately generating randomness to increase the amount of variability (Carpenter, 1999). Instances of noise in the nervous system can be found in the immature command generation, where the pattern of connection between motor neurons and muscle fibers is far from its adult counterpart. In the former, each fiber might be innervated by more than one motor neuron. This is likely to cause problems in the control of movements. Another possible source of noise is incomplete myelination; the electrical properties of neurons (axons) are greatly dependent on the degree of insulation, which is provided by myelin. This noise source is likely to disappear as soon as myelination completes.

Other related work has been done in the field of “computational neuroscience” (Bizzi, 1974), (Goossens & Van Opstal, 1997), (Van Hopstal & Kappen, 1993) though it was often more concerned with the modeling of the “adult” oculo-motor system, than its development. These results could provide an additional constraint on how the “final product” might appear. A quite influential line of research gave rise to the so-called “Bizzi’s model” (Bizzi, 1974). Although this model is now questioned (McCrea, Gdowsky, Boyle, & Belton, 1999), it provides the first clear example of what the role of the vestibular system might be inside an eye-head coordination schema (at least in principle). In the model saccades were generated in a purely retino-topic coordinate frame, thus Bizzi’s model could not take into account movements outside the oculo-motor range (for instance,

remembered or acoustic targets). More sophisticated models are presently considered, which are often referred as “the common gaze feedback models”. They can, in theory, deal with targets outside the oculo-motor range and they also modulate the VOR gain appropriately (in order not to disturb the programmed eye motion, when the target is not yet acquired). Moreover, neural correlates are also available, which further validate these results (Stein & Meredith, 1993), (Wurtz & Munoz, 1996).

1.5.2 Development of reaching

At birth, a human infant can neither reach nor grasp. From a control point of view, the completion of two processes is required to perform successful reaching. First, any neural controller must be able to interact with its “plant” (i.e. the arm in this case), in such a way that “centrally planned”, complex actions can be executed. Second, visually specified goals must be linked to appropriate motor actions. These motor actions, in turn, must be suitable for moving the arm to the desired goal. There are a number of reasons that appear to explain newborn infants’ inability to solve these two tasks:

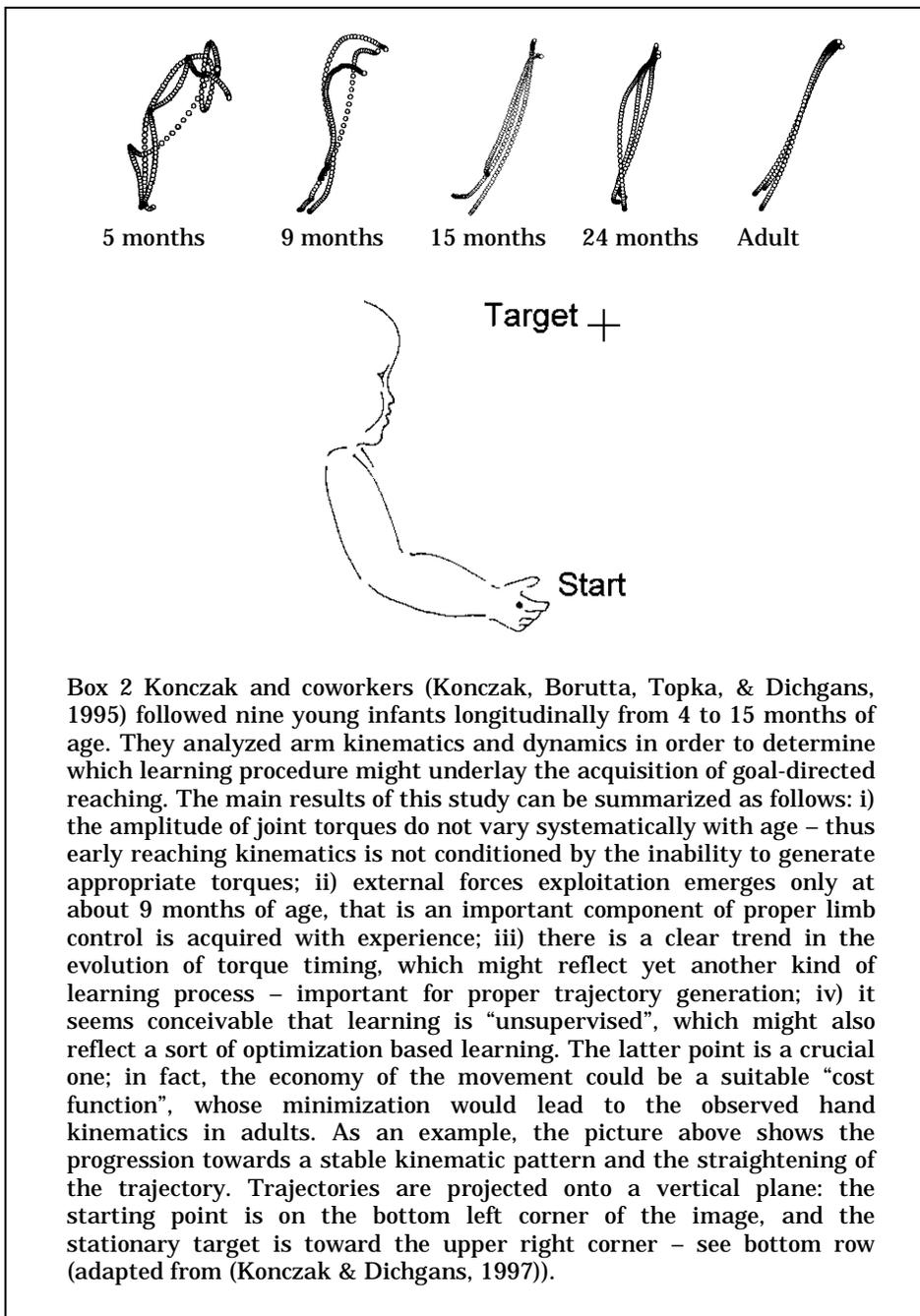
- They have limited postural control of the trunk, head and arms. Appropriate head and trunk righting reactions begin only to emerge 2-3 months after birth (Milani-Comparetti & Gidoni, 1967).
- They have limited knowledge about the physical makeup of their bodies (i.e. moments of inertia, viscosity, stiffness of their arm segments).
- They have only a limited movement repertoire consisting of an array of infant reflexes (i.e. grasping, sucking), and basal intra- and interlimb synergies (coupled flexor, extensor activity, coactivation) (Bekoff, Kauer, Fulstone, & Summers, 1989), (Hadders-Algra, Eykern, Nieuwendijk, & Prechtel, 1992).
- They have limited visual capabilities. During the 1st postnatal month, the visual system provides the infant with functionally useful, but unrefined vision at the level of approximately 5% of adult acuity level (20/200 on a Snellen scale). The infant can likely differentiate facial features from a distance of about 50cm. Objects beyond this distance are probably not seen clearly (Atkinson & Braddick, 1981).
- They have not established a finite neural control structure. Most cortico-spinal projections are not differentiated. There might be different processes, either growing- or pruning-based, which are

known to occur postnatally (Leary, 1992), (Quartz & Sejnowski, 1997).

Despite all these limitations, babies as early as one week of age will attempt small arm movements directed toward objects, and are capable of orienting towards and tracking a moving object by rotating both head and eyes, although their heads may wobble considerably (Trevvarthen, 1980). These early arm movements occur unpredictably, but they are not the result of random activity or pure reflex actions (i.e. they are goal directed).

While the arm movements of newborns are characterized by a rather fluid inter-joint pattern, reach and grasp motions of two- and three-month old infants reveal either short swiping motions or relatively long lasting jerky movements. These movements appear to be pre-programmed, "ballistic" motions, because trajectory correction is absent (Bower, Broughton, & Moore, 1970). That is, in early ontogenesis the role of visual information seems to be restricted to triggering the movement, rather than to visually guiding the hand toward the target by shifting the gaze back and forth between hand and object. Piaget originally proposed such process of "mutual assimilation" (Piaget, 1952). His claim was supported by work of Held and Hein (Held & Hein, 1963) studying the role of forelimb vision during the locomotion in kittens. Their experiments revealed that the view of the forelimbs, during the early stages of development, is essential for fine placements of the front paws. Additional experiments on baby monkeys demonstrated that a complete suppression of the vision of limbs and body, during the first postnatal month, strongly affects hand-to-eye coordination (Held & Bauer, 1967). When the sight is restored, monkeys tended to concentrate much of their attention on their hands rather than on the object to be grasped.

Given the experimental evidence, in recent years, the concept of mutual assimilation and the associated gaze-shifting hypothesis has gone under intense scrutiny. Alt and Trevvarthen (Trevvarthen, 1984) tested infants 16 to 20 weeks of age, that is, just at an age when infants begin to show their first goal-directed reaches. A lightweight screen attached with a headband to the baby occluded the view of arm and hand in either central or peripheral vision. None of the babies attempted to see their arm or hand before initiating a reach or while executing the movement. Along the same line is a result by Clifton and coworkers, who found that babies between 6 and 25 weeks of age did not rely on vision of the hand when attempting to reach for an object (Clifton, Muir, Ashmead, & Clarkson, 1993).



Box 2 Konczak and coworkers (Konczak, Borutta, Topka, & Dichgans, 1995) followed nine young infants longitudinally from 4 to 15 months of age. They analyzed arm kinematics and dynamics in order to determine which learning procedure might underlay the acquisition of goal-directed reaching. The main results of this study can be summarized as follows: i) the amplitude of joint torques do not vary systematically with age – thus early reaching kinematics is not conditioned by the inability to generate appropriate torques; ii) external forces exploitation emerges only at about 9 months of age, that is an important component of proper limb control is acquired with experience; iii) there is a clear trend in the evolution of torque timing, which might reflect yet another kind of learning process – important for proper trajectory generation; iv) it seems conceivable that learning is “unsupervised”, which might also reflect a sort of optimization based learning. The latter point is a crucial one; in fact, the economy of the movement could be a suitable “cost function”, whose minimization would lead to the observed hand kinematics in adults. As an example, the picture above shows the progression towards a stable kinematic pattern and the straightening of the trajectory. Trajectories are projected onto a vertical plane: the starting point is on the bottom left corner of the image, and the stationary target is toward the upper right corner – see bottom row (adapted from (Konczak & Dichgans, 1997)).

Babies contacted glowing objects in darkness, when vision of the hand was restricted, at the same rate as during normal daylight conditions, when they were able to see their hands. That is, the gaze-shifting hypothesis could not be confirmed for the development of early reaching in these studies. The empirical findings indicate that visual “guidance” of the hand is not necessary to establish object contact around the onset of reaching, although it might be necessary for learning to take place.

The first successful goal-directed reaches of human infants appear around the age of 4 to 5 months (Von Hofsten, 1991), (Konczak et al., 1995), (Thelen et al., 1993). The emergence of such behavior around that time is not coincidental:

- By that time, infants had enough time to calibrate their sensory as well as their motor subsystems. Visual acuity has improved considerably, and it is now in the range of 20/800. Around two-thirds of the infants at that age have obtained stereoscopic vision – an important cue of depth perception.
- Higher supraspinal motor centers are operational and reflex behavior can effectively be inhibited to enable the system to acquire more flexible, task-oriented motor behaviors (i.e. suppression of the grasp reflex or the asymmetric tonic neck reflex).

When young infants attempt their first reaches, their movements are jerky and look ataxic. In contrast to the stereotypic kinematic patterns seen in adults, infant hand paths do not follow a straight line, nor do the corresponding velocity profiles reveal a bell-shaped form (Von Hofsten, 1979), (Konczak et al., 1995), (Mathew & Cook, 1986). Within the first 4 to 8 weeks after the onset of goal-directed reaching, kinematic improvements are dramatic (see Box 2). At the onset of reaching, newborn’s hand trajectories consist of about five segments. Two months later, the number of movement units of the hand is halved. By the age of 7 months, a typical reach consists of one large transport segment and one or two additional units in the approach phase. During the approach phase, the palm is usually kept open – a precision or pinch grip has not yet developed. In this first phase of gross-motor reaching, infant motor systems learn to time their neural impulses in such a way that the hand does not over- or undershoot the desired object. In order to achieve this goal, they have to embed basal muscular synergies that are present at birth (e.g. flexing the elbow), into functional, task-adequate multi-joint movements. That is, during early reaching, emphasis is put on refining the transport, not the approach phase, nor on skillful handling of the grasped object.

About 3 months after the onset of reaching, infants reach consistently for objects in their surround and rarely miss their target. By the same time, infants reveal improvements in their manipulative skills (i.e. precision grip). Next to these advancements in the approach phase of the reach, infant motor systems continue to refine the transport phase. Kinematically, their hand paths become straighter, but more important, they now show signs of external force exploitation. For example, they learn that gravity and motion-dependent forces alone can extend their forearms. Consequently, they do not have to initiate elbow extension through muscular activation, but let gravity do the work (Konczak, Borutta, & Dichgans, 1997). Because of this learning process, infant movements become more economical – muscles will be only activated when needed. However, an adult-like skill economy will not develop before 24-36 months of age (Konczak et al., 1997).

Within the first year of life, infants also develop the ability to detour around a barrier to retrieve objects. That is, not only the pure motor act is acquired, but also its adaptive use. Diamond (Diamond, 1981) studied how infants reached for a toy inside a box, using a small transparent box with one face open, (the toy was always visible but could only be reached through the open side of the box). 7-month-old infants reach for the object only through the same side of the box they see the toy. The reaching trajectory follows the line of sight. Successful reaching is achieved only if the object is seen directly (i.e. not behind one of the transparent sides). At about 8-9 months of age, a separation of the line of sight from the line of reach may be observed: infants can look through one side of the box, while reaching through another. However, at this age they still need to see the toy through the opening on each trial in order to succeed (the memory of having seen the object is not enough). By 11-12 months of age, infants become perfect on the object retrieval task, being able to reach the toy from any side of the box efficiently.

1.5.3 Relevant issues for artificial development

The first, and perhaps the major, observation relates to the fact that the newborn is, in a systemistic way, a “complete” system in the sense that all sensory and motor components are present and functional. The performance of each component is not developed, but it is remarkable how, in spite of their relative immaturity, the performance of all components seems to be well matched.

Sensori-guided coordination is absent at birth but other mechanisms, such as motor reflexes and sensori-triggered motion, are present exploiting

the still limited sensory and motor abilities and allowing the infant to start some form of interaction with the external environment and the acquisition of the first sensori-motor experiences.

Throughout the developmental stages described previously, the maturation of all “sub-systems” proceeds harmoniously and the motor abilities are matched to the sensory and cognitive ones. It is worth noting, however, that this process cannot be modeled entirely as a learning process because, during development, the system itself drastically changes its own motor strategies. For example, from a purely reflexive system to a system capable of voluntarily initiating “dominant motor sequences”, arriving finally to complete voluntary, sensori-guided control.

During these phases, some of the abilities are only temporarily present (for example, some of the early reflexes) and are strategically used to take full advantage of the very early sensori-motor experiences. Later, they disappear as soon as these skills are replaced by more developed ones. For example, the infant does not learn to control simultaneously all the degrees of freedom of his/her arms, but the first exhibition of reaching behavior is a ballistic-like, posture dependent swiping motion of the arm, with the ability neither to correct the trajectory nor to control pre-grasping postures of the hand. In some sense, the system seems to practice with just a few joints before attempting more complex motor acts. The underlying control structure takes care of maintaining archetypal postures, controlling the remaining degrees of freedom in a “reflex-like” way.

Reflexes, such as the grasping reflex or the tonic neck reflex, are present and facilitate the interaction with the outside world, even with such a limited control strategy, in order to provide a sufficiently high success rate. In this respect, one could argue that if all degrees of freedom were under voluntary control, it would be a lot more difficult to learn complex motor actions. The fact that the infant is not perceptually skilled becomes, in this view, a positive factor because it makes successes more probable and easier to repeat. This is true, of course, if the system is designed in such a way that the motor, perceptual and cognitive abilities proceed harmoniously.

1.6 Seeing, reaching and touching

In order to substantiate the theoretical claim we shall make in the following sections, we carried out several experimental investigations. Though the primary concern was to show that a particular controller configuration is important for the learning process itself, we measured the robot performance after the training phase. In terms of orienting behavior,

the goal for the head subsystem was to acquire a visual target, act to get it in the foveae, and eventually coordinate the redundant degrees of freedom. The head controller, both learnt how to follow smoothly the movement of the target (i.e. a sort of smooth pursuit behavior), and how to orient the gaze toward a new perceived location (i.e. saccade-like movements). Moreover, coordination of head and eye movements can efficiently exploit the information coming through proprioception (i.e. by using a sort of vestibulo-ocular reflex – VOR), which is provided, in this case, by the inertial sensor and the optical encoders. As we shall see, a successful coordination requires the mapping, or frame of reference conversion, from the sensory to motor data. In other words, the robot has to learn also how to tune these transformations. Once a simple orienting behavior is established, the robot can start the process of acquisition of reaching, although a simple reflex-like arm extension mechanism was handcrafted into the system from the beginning. As soon as gazing is precise enough, reaching also improves quickly, allowing the robot to start a new kind of interaction with the environment: i.e. touching.

1.6.1 Phases and components

In order to acquire the correct information for building the maps or transformations described above, the robot must follow a precise “developmental” course (sketched in Figure 3). In practice, the system is able “at birth” to move the eyes only. Control, at that stage, is a mixture of random and goal-directed movements. With respect to the head-arm coordination, at this stage the robot possesses only a reflexive behavior simulating basic muscular synergies and spinal reflexes.

The initial task of the control process is that of calibrating the closed loop gains – in many cases biological systems have also to tune delays in order to have the relevant information “in phase”. Afterwards, gains can be properly adjusted (Von Hofsten & Rosander, 1997), (Distler, Vital-Durand, Korte, Korbmacher, & Hoffmann, 1999), (Cioni, Favilla, Ghelarducci, & La Noce, 1984). In this robot implementation though, delays were manually adjusted beforehand.

It is worth stressing that even at the very beginning the system is already moving in a “goal-directed” manner, although noise dominates the actual movements. In successive phases, the robot starts learning saccades, but only the eyes are moving. Indeed, this is necessary because otherwise the neck motion would disturb the estimation of the required eye commands (i.e. part of the required eye movement would be indirectly performed by the head motion). In order to relax this constraint, the robot

actually moved the head but with a very low probability. The latter is the developmental parameter, which has been used to intermingle the two learning processes. Of course, the smaller the head motion the better the training data.

Once eyes are under “proper” control, the whole head starts moving, at this point, the saccade maps are well formed and can be used to help coordination of the redundant eye-head degrees of freedom.

Concurrently, reaching steadily improves by storing more information in the head-arm coordination map. As a result, the initial reflexes become part themselves of the coordinative action. On the other hand, because reaching depends on gazing, during the initial phases, reaching improves slowly. Later, as soon as gazing obtains a reasonable performance level, reaching also improves quickly. It is worth stressing that, from the robot’s point of view, motor control can be seen as “learning” to combine the initial “skills” – i.e. reflexes – in order to obtain voluntary goal-directed movements.

The VOR is always turned on, and any stimulation allows the robot to obtain more data, which are used to tune the vestibulo-ocular response. When the first multi-joints eye-head movements are practiced, the VOR is already effective in facilitating coordination. This is important as pointed out in the following chapters.

1.6.2 The big show after learning

Once head and arm controls are in place, the robot can orient appropriately toward moving stimuli, follow them while moving, and eventually, it can try to touch the tracked object. Roughly speaking, Babybot starts by looking at objects, which are identified because of their motion. It can correctly saccade, and it possesses a sort of smooth pursuit ability. It is worth mentioning that only the eyes are controlled directly by means of visual information. The redundant DOF are easily “centrally” coordinated. This ability to gaze is the first step toward yet another visually driven behavior: i.e. reaching. By mapping gaze direction into appropriate motor commands, the robot can effectively reach for objects in extrapersonal space. Moreover, thanks to a low stiffness controller, Babybot can safely interact with humans and the external environment. If we wish to build a truly autonomous system, this robot-environment-humans interaction is of paramount importance.

Figure 4 shows the robot’s trajectories of both the fixation point and the arm end-point. They were acquired during an unrestrained experiment: that is, an experimenter handled the target, in such a way to

cause the robot to react. The whole experiment endured for about half an hour during which joint positions were recorded at $25Hz$ rate. Two more plots, in Figure 5, show the same trajectories from different viewpoints.

1.7 Discussion

This chapter, as well as presenting the formulation of our aims and goals, showed informally that biological systems, which manifest a remarkable adaptive and skilled behavior, do not come with monolithic control structures. On the contrary, especially in the very first periods of their life, they go through a peculiar sequence of developmental events. We argued, that newborns are not a sort of “tabula rasa”: they possess a series of stereotyped behaviors implemented as reflexes. We also pointed out, that noise might play a role in allowing the controllers to explore the available “state space”, but more importantly, that the control structure changes during development – mostly because new modules come into play and some of the existing ones become embedded or mixed together.

Furthermore, this chapter presented some experimental results from “brain sciences”, which support this view. We discussed what might be relevant for “artificial development”, that is for mimicking a developmental process in an artificial system. We also suggested that, through testing models by using a robot, we might address specific biological related questions, or suggest novel testable hypotheses.

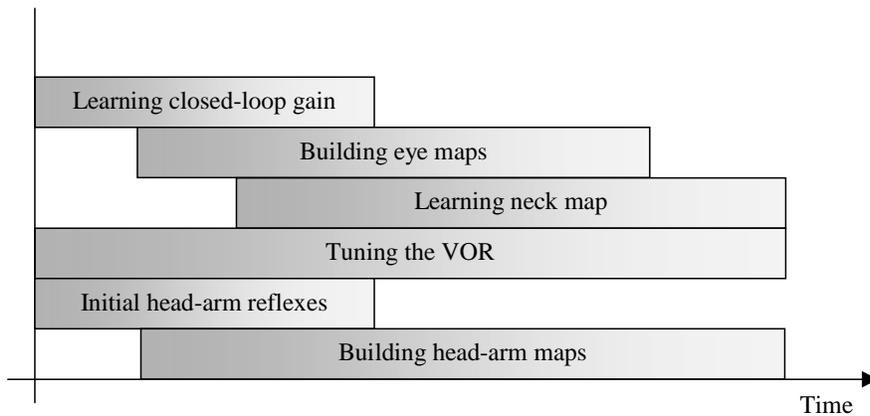


Figure 3 The developmental stages. The diagram above approximately shows the interleaving of the developmental stages; abscissa represents time. The first step is the acquisition of the closed loop gains; reflex-like modules control the arm subsystem. After a while, learning of the saccade control begins. Whatever movement of the robot also stimulates the inertial sensor: this information is used to tune the VOR. Eventually the eye-head coordination is acquired together with a more effective head-arm coordination map.

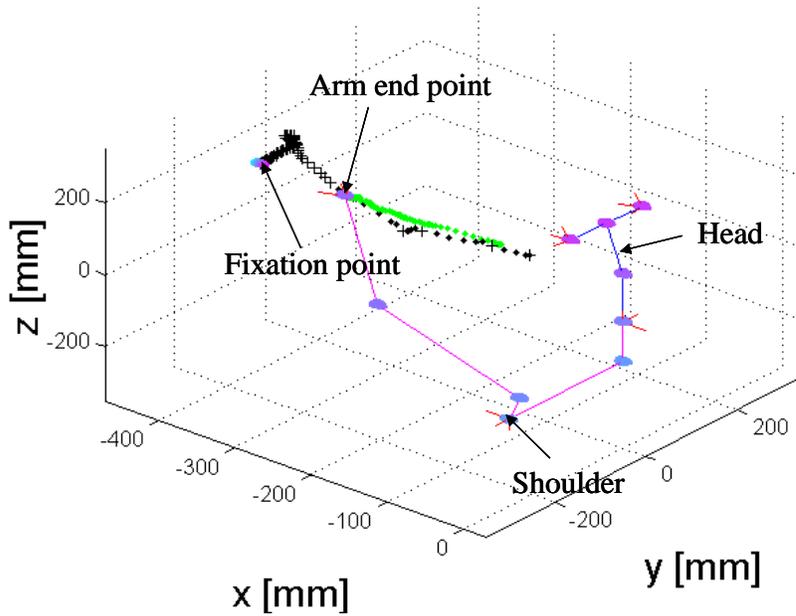


Figure 4 Gazing and reaching. Two trajectories are shown, the fixation point and the arm end-point respectively. The simple wire-frame model represents the robot. Small circles indicate joints; solid lines are the links. Concerning the fixation point, two different marks can be distinguished: the crosses represent the time instants when tracking was of smooth pursuit type, the small squares are related to saccadic control. Note that the arm end-point follows the motion of the fixation point up to the moment when the target is too far away to be reached.

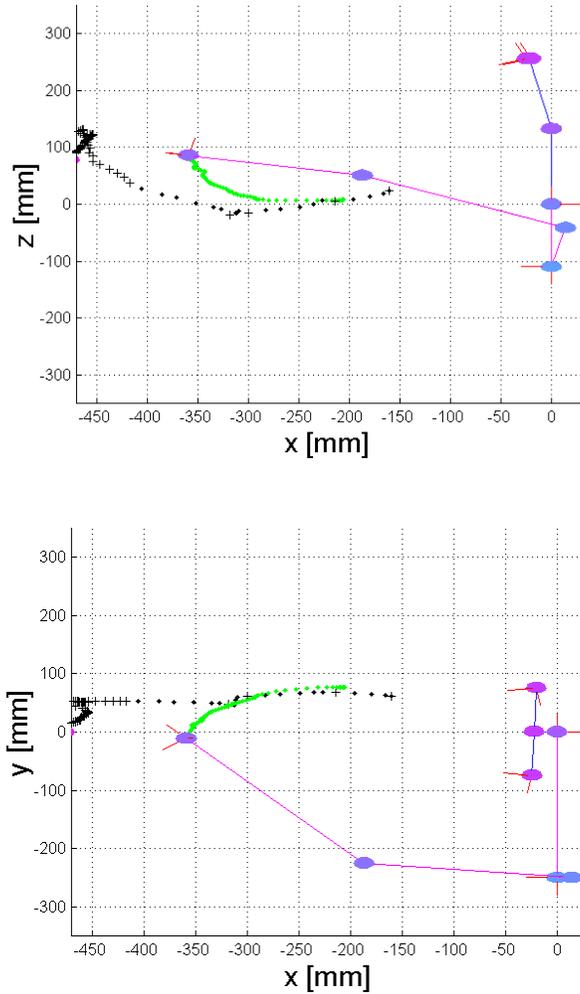


Figure 5 Two views of the same trajectory plot shown in Figure 4. The upper panel is the lateral view; the lower plot represents the top view. As before, fixation point and arm trajectories are shown. The fixation point motion is described by either cross marks (when smooth pursuit is active) or square marks (saccadic control).

2 The learning problem

It has been recognized that learning from examples is an ill-posed problem (Vapnik, 1998), (Poggio, Torre, & Koch, 1985), (Geman, Bienenstock, & Doursat, 1992). Every learner faces the so-called “theoretical pressures”, which require balancing competing needs in order for learning to be feasible. Recently, a number of theories on learning formalized these problems (Vapnik, 1998), (Carpenter & Grossberg, 1986), (Sutton & Barto, 1998). Generally speaking, a learner should be able to learn from incomplete information, using a limited number of samples, and quickly enough to cope with changes in the environment, as well as of its internal physical parameters (e.g. growth, malfunctions, etc). The first step for any learning agent is that of acquiring information through the interaction with the environment. However, without any *a priori* information, it is hard to tell which part of the “state⁵” space is worth exploring in order to solve a particular task. As a matter of fact, the size of the state space might consist of hundreds of dimensions, which precludes any type of enumerative search for a solution. However, it is not always true that the solution belongs to the whole state space; on the contrary, in many cases the actual problem rests on a lower dimensionality manifold (Schaal & Atkeson, 1998). This suggests that, if the learning process is carried out together with the identification of the relevant sub-manifold, a complete exhaustive search can be avoided.

It turns out that learners have two competing requirements in terms of exploring the control/state space, and in responding as much as possible appropriately to stimuli (i.e. exploit their knowledge). Recent research on human development suggests that such exploration component might be provided naturally by noise. In fact, newborns show several noise sources: due to incomplete structures (non-myelinated neurons are an example); due to unnecessary neural branching (such as in the neuro-muscular junction); and, by using random behavior actively (latencies on saccade generation). This role of noise during learning resembles the usage, in system theory, of broadband (e.g. white noise) input signal for system

⁵ The state space can be a proper state space, the parameters manifold, or a combination of the two depending on the kind of learning algorithm considered. The discussion presented here applies to all of them.

identification purposes. In other words, the system has to be “excited enough” in order for identification to be feasible. Of course, if the control were noisy, motor tasks would not be accurately fulfilled; on the other hand, if the learner were too “static”, it might never learn the correct solution to the task.

Further, other researchers provided evidence for the existence of a strong “goal-directed” behavioral component, even in newborns (Streri, 1993). There might be a twofold reason for that: firstly, even if high exploration rates might be useful, a complete random behavior could be remarkably inefficient⁶. Instead, a mixture of “goal-directed” command generation and noisy control could balance the two requirements. Secondly, the fact that the behavior is goal-directed can speed up the acquisition of the appropriate controller, e.g. imagine a task with a single target state, in this case a goal-directed agent might solve the problem for only a relatively small neighborhood of the target. On the contrary, a random explorative search has to test all possible states, unless some *a priori* knowledge is inserted into the system. Lacking of any constraint, the random “explorer” needs to visit all possible states prior to any actual control; otherwise, a possibly useful part of the state space might remain unexplored.

Furthermore, the cooperation of many control loops developing with different time spans can help in reducing the already mentioned exploration space. Roughly speaking, each control loop generates a bias for subsystems that develop later. In the context of “computational motor control”, one notable example of such a schema is the feedback-error learning model (Kawato, Furukawa, & Suzuki, 1987). In this case, an inverse modeling is carried out through the interaction of a learner with a much simpler feedback loop. Similar multi-loop structures can also be observed in the brain. An example of this process is the so-called cortical take-over, where cortical areas develop on top of sub-cortical structures. At birth, the sub-cortical loops control behavior, although imprecisely. As development progresses, the cortex takes over the control process and eventually establishes a functional projection toward the sub-cortical circuitry (Stein & Meredith, 1993). It is worth stressing that this process is not definitely on/off; in fact, the motor output could be a combination of multiple streams’ output as well. Furthermore, the delay and bandwidth involved in the various structures can be different thus providing the basis

⁶ If the state space is n -dimensional the search and selection of a proper behavior (in the worst case) can cost up to $O(n!)$.

for faster reactions (reflex-like) and accurate control at the same time (consider, for instance, the visuo-vestibular integration).

Implicit in the preceding discussion is the assumption that the learning agent is functional from the beginning, which means that the training data must be collected on-line. This is a major constraint for biological as well as artificial systems. Concerning biological systems, it is clear that they could not be some sort of “blank slate” at birth; they rather need to have some useful bootstrap functionality. These initial behaviors are usually reflex-like and stimulus bound in nature (Bekoff et al., 1989), (Hadders-Algra et al., 1992). They can be thought as the initial bias⁷, and perhaps their role is indeed that of guiding the system through feasible regions of the state space.

It is worth stressing that the exploration-exploitation tradeoff is closely related to the well-known engineering problem called “the curse of the dimensionality” (Bellman, 1956). In fact, the need for representational resources grows exponentially with respect to a linear growth of the number of dimensions. For an on-line learner, the time to explore the state space would suffer of this remarkable growth. Moreover, the bigger the space the sparser the data. Indeed, it has been shown that there is a limit on the mathematical consistency of topographic mapping based on neighborhood relationships (roughly 20 dimensions) (Scott, 1992). The latter also has been shown to be an overall organizing principle in the brain (Hubel & Wiesel, 1977), (Hubel & Wiesel, 1974). Hence it might be important to limit the size and dimensionality of the state space, which would allow topographic mapping to be carried on reliably.

To recap, we can ask the following question: what are the practical consequences, in terms of learning and particularly on the learner’s functional organization of the “theoretical pressures”? We should note that: firstly, if a system is highly biased, it is of course easy to train, although it might suffer of poor learning (approximation) performances. On the other hand, if it is too general it might be very hard to gather the required training set. Even in this case, learning would be neither optimum nor particularly close to it. Secondly, whenever the system is general enough, the state space tends to be extremely big. At the same time the learner needs to explore such a state space, though it needs also to behave properly (exploitation), which seems to preclude a pure random (enumerative) exploration. Biological systems overcome the impasse by

⁷ Proper bias selection leads to another impasse usually called the bias-variance dilemma.

adopting a series of different mechanisms. The following sections describe in details some of these aspects.

2.1 Bias and variance

In the context of function approximation (including some neural network models), the bias-variance dilemma (Geman et al., 1992) can be illustrated qualitatively by observing that, in the case of a norm-2 based error e , it can be decomposed in two terms (Ljung, 1987):

$$e = E \left\{ \left\| f(x) - \hat{f}(x, w^o) \right\|^2 \right\} + E \left\{ \left\| \hat{f}(x, w^o) - \hat{f}(x, w_T) \right\|^2 \right\} \quad (1)$$

The first addendum is called “bias”. It measures the distance of the optimally tuned approximator $\hat{f}(x, w^o)$ from the unknown target function $f(x)$. The second term (“variance”), on the other hand, represents the distance of the current parameter approximation $\hat{f}(x, w_T)$ from the optimal one. E represents the statistical expectation operator. The dilemma appears when we try to jointly minimize both components. In fact, if the model is highly biased, it might not grasp the true function embedded in the training set, although the variance might be low because of the reduced number of parameters. On the contrary, if we increase the model complexity, the bias is reduced but the variance is increased unless we also increase the training set size. In other words, simple models are easy to train and should be preferred unless we realize that they are too limited. In the latter case we can resort to some improved model by increasing the number of free parameters (i.e. the model complexity).

The effect of employing a high bias is that of over-smoothing: that is, the approximation does not faithfully follow the training data, because the model is inadequate to approximate it appropriately. On the contrary, if the model is too complex, the opposite happens: the approximation will follow every single variation in the data set, maybe also those which require a very high slope. In this case, generalization is lost, unless a huge training set is employed – infinite in the limit. The majority of the learning theory obtained results concerning the behavior of the approximations in the limit; of course, this is not realistic. Modern theory of learning, started considering also those cases where the training set is limited (Vapnik, 1998).

A sensible strategy to balance bias and variance could be that of controlling the model complexity so to avoid either over-fitting (see an

example in Figure 6) or over-smoothing – a very elegant solution, of mainly theoretical interest, is presented in (Vapnik, 1998).

Without entering into the details too much, it has been conjectured that growing networks are able to cope with the situation. Furthermore, some researchers reported that the expression power of such models, employing complexity control, might be superior in terms of learning performance (Quartz & Sejnowski, 1997).

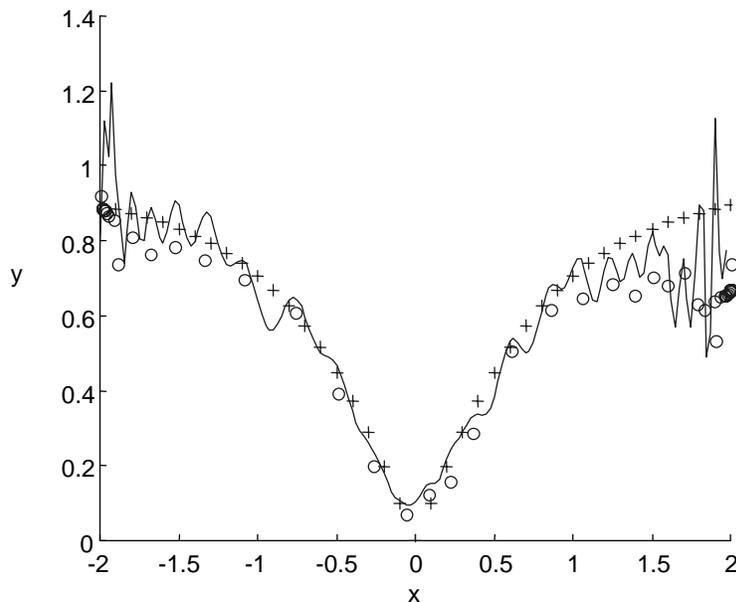


Figure 6 A neural network (RBF) over-fits the data. In this case, a very limited training set was used. The network has more units than training points and consequently cannot properly approximate the data.

Proper bias selection leads to the question of what the appropriate starting point is. As described in section 1.5, biological systems possess a repertoire of movements and reflexes, which are exploited as a sort of “bootstrap” functionality. In broad terms, they can be seen as a sort of bias – or basic behaviors. On the other hand, because the control system is “growing”, it is also true that the basic behaviors could be either extended

or removed if no longer necessary – i.e. the system is not limited to those behaviors or their combination.

2.2 Exploration versus exploitation

Though the bias and variance dilemma concerns general approximators, there is another major constraint for “on-line” learners. Actually, getting training data on-line poses yet another requirement: i.e. to explore the learning space. The “learning space”, depending on the algorithm considered, might consist, for example, of the Cartesian product of the input and the output space or the space of “tunable” parameters. Beside the size of such a space, which might consist of dozens of dimensions, the learner cannot only devote all resources (time) to testing different input-output combinations. It is clear, in particular for biological systems, that learning must take place in a reasonable amount of time. The learner could not just test all possible combinations (randomly) before taking a suitable action. It has also to employ its skills to try to solve the task itself (though it could not be the optimal strategy). The words of Sutton and Barto (Sutton & Barto, 1998) nicely resume this point: “...*in interactive problems it is often impractical to obtain examples of desired behavior that are both correct and representative of all the situations in which the agent has to act. In uncharted territory – where one would expect learning to be most beneficial – an agent must be able to learn from its own experience*” p.4.

In other words, it turns out that the learner has to balance these two requirements: exploration of the state/learning space and exploitation of the current abilities at the given phase. This is a particular feature of on-line learners and it is never encountered if we deal, for example, with traditional supervised learning. If we provide the system with a training set, of course the problem is simplified.

Moreover, we suggest that the organization of the learning modules might be designed in order at least to reduce the problem. This is because if many sub-systems have control of the same physical structure, it might happen that not all the state/learning space has to be explored. One module, for instance, can provide the training signal for another, thus reducing the actual search. This concept might be very well exploited by biological systems, in fact, as already pointed out, they do not come as a monolithic structure, rather they show a modular one where different parts develop one on top of the other.

To illustrate this last point, we carried out a simple simulated learning experiment. The idea here is first to test two components of a

double loop schema in isolation (namely an inverse model acquisition and a closed-loop controller), and finally to show how they perform together. The goal is that of controlling a discrete linear system in order to reach a stationary target. The learner does not have direct access to the state though, it can be measured through a non-linear channel as shown in equation (2):

$$\begin{cases} \mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{u}_t \Delta T \\ \mathbf{y}_t = k \arctan(\mathbf{x}_t) \end{cases} \quad (2)$$

where \mathbf{x}_t is the state vector at time instant t , \mathbf{u}_t the control variable and ΔT the time interval of the simulation. The measure of the state \mathbf{y} is non-linear, and k is a normalizing factor. The state space is bi-dimensional and the system is forced to lie within the region $\mathbf{x}_t \in [-5,5] \times [-5,5]$; the quantization step is equal to one (overall the space consists of 121 cells). The goal of the controller is to learn how to move the system to (0,0) in the smallest amount of time. We tested three different learning models, namely:

- The random explorer. This controller uses a map (look-up table) to associate previously visited states to commands. The idea is that of randomly interleaving “goal directed” commands and random motion. A new correspondence between visited states and motor commands is stored, if and only if, the new tested value can reduce the distance from the target more than the possibly previously stored one. Otherwise, the old value is kept in the map. The learner exploration space consists of 11^4 states.
- The closed-loop controller. This model learns the transformation between the error and the motor space, and uses it to drive a negative feedback loop controller. This is similar to what described later in section 3.1. The exploration space in this case is reduced to only 11^2 possible states.
- The double loop schema. It uses the inverse model (which is similar to the random explorer), though the closed loop controller now drives the exploration. The exploration space is reduced from 11^4 to 11^2 because the closed loop subsystem provides the correct training samples to the inverse model mapping.

The three systems were tested for 5000 control cycles. We measured the average reaching time from a random position within the state space to the target in (0,0). Further, we counted the number of successful trials

over the 5000 cycles. We kept the testing to the initial stages of learning in order to evaluate the learning performance themselves, rather than the steady state solution to the problem. The comparative results are shown in Figure 7. The plot represents the number of successful trials for the three different cases as a function of the exploration noise. Noise represents here the probability of taking a “greedy action”⁸ respect to a “random explorative” one.

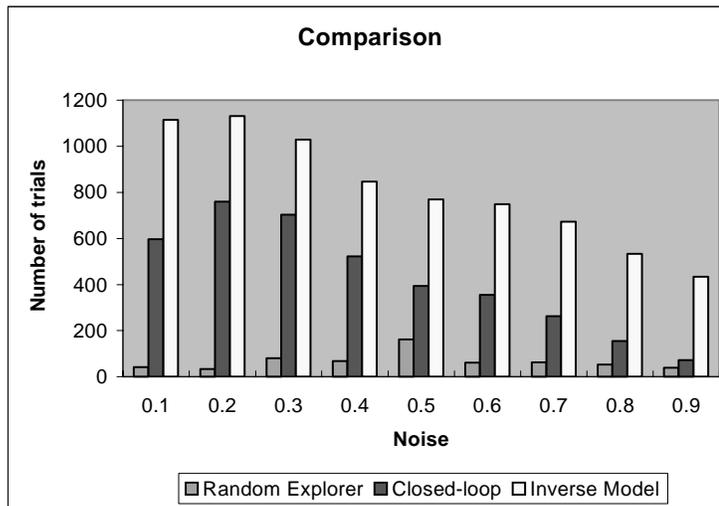


Figure 7 Results of the performance test of three learning algorithms. Abscissa represents noise (i.e. the probability of taking a “greedy” action versus an explorative one), ordinates the number of successful trials out of 5000 control steps. These results endorse the hypothesis that the solution to the control problem actually lies in a sub-region of the whole control/state space – there are in practice “wrong” regions of the state space, which do not need to be explored. In this case an algorithm such as the closed-loop or the inverse model based one performed comparatively better than a “random explorer”. In terms of performance the “inverse model” schema is running two orders of magnitude better than the “random explorer”.

⁸ Greedy means that the action is directed at exploiting as much as possible the current knowledge about the problem, without considering that in the long term, an exploratory one might lead to better results.

Lower values of noise mean that the system is behaving more greedily. As can be seen, there is an optimum of the noise level (the maximum in the plot), and this is attained at different noise levels by the various learning strategies. In fact, it is likely that the more goal-directed methods (i.e. “closed-loop” and “inverse model”) require a smaller amount of time in order to get useful information (i.e. a working controller), and consequently they require less noise. This is reflected also by the fact that they are performing comparatively better.

2.3 Stability and catastrophic forgetting

Another problem, frequently encountered in connectionist models, is the so-called catastrophic “interference” or “forgetting”. While doing sequential learning, that is presenting a sequence of patterns to be learned, connectionist models show an abrupt forgetting of previously learned patterns (French, 1999) – human learning, on the contrary, shows a gradual smooth forgetting. The problem is the downside of the stability-plasticity dilemma; in other words, how can a distributed network be plastic to new inputs and, at the same time, stable to previously learned associations?

The problem turned out to be tightly linked to the presence of a single shared set of weights. In practice, the fact that new inputs influence all weights in the network causes also interference with old learning. In order to alleviate this effect, networks with semi-distributed sets of weights have been proposed. The solution separates new learning from old learning. This has been shown to allow partially graceful degradation.

There is no clear theoretical study on this problem. An outstanding model, which has been suggested to have links to how the brain eliminates the problem, is the dual architecture. In practice, there are two modules, one is a standard connectionist network (called processing network), and the second is a control network, which has the task of modulating the weights of the first (Rueckl, 1993). There are clear hints that the solution lies halfway in distributing the representation of the weights and in separating processing from control, in order to discriminate newer inputs from older ones. Some researchers suggested that part of the brain organization might indeed be employing this mechanism, with the hippocampus being a fast learning module and the neocortex a gradual, slower, one.

In this sense, we argue that some of the key factors for the success of a learning architecture might be the particular design choices, in terms of distribution and representation of the input patterns – see localized

receptive fields – and the interaction of modules with different structural architectures. As shown earlier in section 2.2, a single structure would fail in the task, while many interacting modules might succeed.

2.4 Dimensionality: the big curse

Yet another major problem faces learners: the curse of the dimensionality. The problem was first noticed by Bellman in the context of dynamic programming (Bellman, 1956), where he noted that the need for memory resources grows exponentially for a corresponding growth of the number of dimensions of the state space.

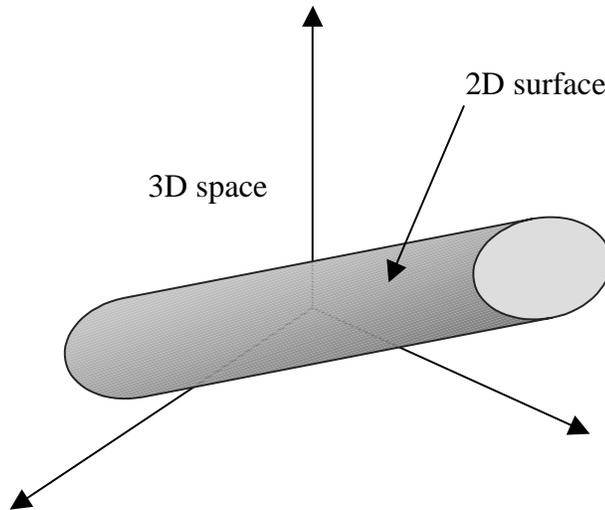


Figure 8 A cartoon drawing illustrating the effect of constraining the state space. In this case, dimensionality is reduced because the problem itself is inherently two-dimensional. Furthermore, another principle is shown: though the initial formulation is made on \mathcal{R}^3 , the actual problem has its own precise limits – i.e. it is defined on a limited set of points. On top of this, the learner can apply a variable resolution-coding schema, thus maximally exploiting a limited amount of resources.

As an example, imagine that we have to allocate a memory location for each element of an N -dimensional space. Imagine also, you decide to sample uniformly the state space; each component of the N -dimensional

state vector would be quantized using Q different levels. The total required amount of memory, in this situation, would be Q^N – i.e. exponential in the number of dimensions. Clearly, as N grows we would have problems obtaining and handling such a large amount of memory.

Is there any solution? Any chance to cope with this problem? Actually, the answer is “maybe”. There might be methods to alleviate the problem. One can devise a schema with variable resolution, where the sampling is changed dynamically on the basis of some “usefulness” criterion. This is also observed in the brain, where most of the mappings belong to this category – just think about the sensory maps (Hubel & Wiesel, 1974).

The other strong hypothesis we argue about is that the actual problem does not necessarily belong to the whole N -dimensional space. Some authors (Schaal & Atkeson, 1998) noticed that in many cases the actual solution lies in a lower dimensionality sub-space. Our conjecture is that, furthermore, each single module may need to know only about a limited part of the state space, which may be determined by how the module itself is connected (inserted) in the whole schema. The picture below shows this principle graphically. On the other hand, it is hard at the moment, to go farther than this, because other questions arise: how can we design learning modules in order to exploit this beneficial interaction? What is the role of the environment in shaping such interactions? Unfortunately, we do not have answers yet.

2.5 Is the environment interaction necessary?⁹

We address here the necessity of “embodiment” in the light of the bias-variance dilemma discussed above. As pointed out by Vapnik (Vapnik, 1998), the problem of learning from a small training set can be addressed by means of the Structural Risk Minimization (SRM) technique. Roughly speaking, SRM consists in choosing “hypothesis spaces” with increasing VC capacity, and finding the optimal one in “some sense”. The theory has been extended also for cases where the VC dimension is not directly utilizable (i.e. it is infinite) (Evgeniou, Pontil, & Poggio, 1999). Although these results are mainly of theoretical concern, they suggest what one can expect to be learned by a statistical learner from a limited amount of information.

In practice, at least for the standard SRM technique, it is possible to determine bounds on the consistency of the solution. It turns out that the bounds depend both on the “empirical error” and on the “VC dimension”

⁹ Refer to (Vapnik, 1998) for the definition of VC dimension.

itself. In this sense, a high VC dimension gives a low “empirical error”, though the quality of generalization decreases consequently. Note also that the true “unknown” function may lie outside the hypothesis space, so that the minimum might not be necessarily small – this is called “approximation error”, and can be evaluated by means of techniques borrowed from standard approximation theory (Lorentz, 1986).

Furthermore, Evgeniou et al. (Evgeniou et al., 1999) have shown that SRM can be related to regularization theory. In this sense, there is a formal justification of many of the artificial learning techniques, such as Radial Basis Functions networks. It is fair to say that, from the practical point of view, “cross-validation” is the most effective technique because SRM (involving the VC dimension) is computationally intractable. In other words, both SRM and regularization methods propose, in order to cope with the ill-posed nature of the problem, to optimize yet another parameter: the capacity of the approximator for SRM, or the smoothness coefficient, in the case of regularization. Both strategies are equivalent under some precise conditions.

What is the link to biological learners? What’s about “artificial learners”? Again from the theoretical results, we notice that a consistent solution needs to be found by regularization, or by exploring the sequence of hypothesis spaces. That is, the regularization parameter has to be optimized as well – read cross validated. This has a profound impact on the learner, because without the interaction with the environment there is no chance to perform a proper cross-validation. On the other hand, for an embodied learner, there might be the chance to evaluate the network “model” by measuring some “performance parameter” on a global level. This sort of signal behaves as a reinforcer by telling the learner whether the model is “good” or “bad” for the task.

A biologically plausible mechanism, which is supposed to provide reinforcers, is the emotional system. Speaking from a purely biological point of view, emotions can be seen as “states produced by reinforcing stimuli”. Some neural structures like the amygdala, the orbital cortex, and the cingulate cortex, as well as other sub-cortical areas are involved in the processing of such signals (Adolphs, Tranel, & Damasio, 1998), (Ledoux, 1996). Among them, the amygdala seems to play a crucial role, being implicated in the learning of associations between stimuli and reinforcers. Without entering into details, the emotional system areas collect a huge amount of different sensory afferences, and combine them in order to establish “some general properties” of a given situation (like the goodness of the situation). Besides allowing fast reactions, for instance, in dangerous

situations, those areas influence the sensory processing, and the learning itself. In this light, it is clear that the agent-environment interaction becomes a key factor for a successful learning process.

2.6 Do we need to “perceive” in order to control our movements?

Many recent results favor the fact that the brain possesses specialized areas dealing with different kind of information for different purposes. An example, and perhaps the most influential, is the Milner and Goodale (Milner & Goodale, 1995) proposal of the two streams (i.e. “what” and “where”). These vision-related processing pathways have been correlated either to the response properties of neurons (magno vs. parvo) or, more recently, to different frame of coordinates for the representation of the visual space (allocentric vs. egocentric).

This is yet another example of the compartmentalization of the brain, where different modules, though working in parallel and together, carry out different computational processes.

The relevance to these results for our developmental model comes mainly from further experimental evidence (Atkinson, 1998), showing a different developmental time course of the different pathways. As already mentioned, in light of what discussed in section 2.4, the “state space” of the system, might be constrained (for the learning purpose) because eventually all these pathways converge and control the same “plant”. In other words, not all the variables (connections) are free to change, and not all the space needs to be explored.

There is evidence also for different performance levels within the attentional system, whether or not an “involuntary” attentional mechanism is employed in a given task (Posner, 1980). More recent results showed that there is a dissociation between the proximal and far spaces, which is related to the underlying coordinate systems – egocentric for proximal/manipulation space, allocentric for distal/perceptual space (Maringelli & McCarthy, 1999).

Though not explicitly investigated in this book, the same principle might be useful for designing a complex artificial system. Consider, for instance, that one of the most influential “copied” models was the Marr’s computational model of the visual system (Marr, 1982). In brief, by processing images AI researchers tried to extract precise physically meaningful quantities, such as 2D/3D representations of the visual scene – probably they were somehow misguided by Marr’s intuitions. Is it really what we need for controlling our movements? Do we really need to “perceive” in order to, let’s say, grasp a visually identified object?

Our personal feeling is that the answer is no. That is, motion control can be performed by a simple processing, taking care to extract only the relevant information. This is also economical, in terms of the required computational power: resources are allocated where needed. Another question is whether the robot can identify what is relevant for a task just by using a learning procedure.

As a matter of fact, it could be better to have many simple modules working together, rather than a huge single control structure (aka old-AI-style). Nonetheless, the former approach has been pursued only recently (Brooks, 1986).

Our experiments on the robot show, in some sense, that this is the case. Resolution is not high, vision is crude, the controllers simple, but nevertheless the system can gaze appropriately and reach for objects in its extrapersonal space. That is, nine degrees of freedom are under adaptive control just by employing about 4000 visual receptors (overall), and by using very standard PCs!

2.7 Noise is everywhere

As we suggested at the beginning of this chapter, noise might play an important role in driving the exploration of the state space, which is necessary for learning to be carried out. This is true for whatever identification schema employed. Generally speaking, identification needs information about the unknown plant. If we restrict our discussion to linear dynamical systems, for instance, the identification procedure would consist of providing a suitable input to the plant, and observing its output. The best input is a unitary pulse, its output the pulse response of the system that, in the case of linear system, would uniquely characterize its behavior for all possible inputs. In the frequency domain, the pulse is characterized by a flat unitary response over the whole spectrum. Thus, the pulse stimulus provides the maximum amount of information for the system identification purpose.

In everyday life, plant dynamics may be strongly non-linear thus requiring many different inputs in order to characterize completely the system's response. Nonetheless, the necessity to use broadband input signals (stimuli) remains.

Noisy inputs have the same spectral characteristics; for instance, white noise has a flat response over the whole spectrum. Again, in real life problems, even a band limited (but large enough) signal can offer significant advantages over a narrowband signal. This situation has, in reality, profound links to the exploration-exploitation dilemma discussed

above. In fact, though exploration is facilitated by noise, which allows easy system identification (read learning), it compromises the task execution.

We conjecture that some natural “noise sources” in the brain could actually facilitate learning by keeping the “plant” stimulated enough. They would play the role of the aforementioned broadband signals. The amazing part of the story is that these sources are likely to disappear during development; that is, once the plant is mostly identified, they do not need to be operative anymore. These can be seen as an endogenous mechanism allowing system identification; it is clear that another major role in learning is played by the environment influences, we should not forget that by “depriving” a learner, learning itself is compromised.

One possible source of noise might be due to the immature pattern of neural innervations (Purves & Lichtman, 1980). Concerning, for example, motor control, at birth each muscle fiber may be innervated by more than one motor neuron. On the contrary, the adult counterpart shows that each muscle fiber is contacted by one, and only one, motor neuron. This pruning phenomenon is probably determined by trophic factors.

Another similar situation can be observed in the Purkinje neurons in the cerebellum. In the adult, each Purkinje cell is innervated by only one climbing fiber, while during development inputs come from many fibers.

These early configurations might cause problems, and consequently noise, in the control of movement thus serving as an endogenous noise generator.

Another source of noise, during development, is caused by the lack of myelination. Neurons, which are known to be myelinated in the adult, complete the process only postnatally (Kandel, Schwartz, & Jessel, 1991). Myelin provides an insulation sheath to axons, and consequently reduces transmission times for action potentials. On the contrary, when myelination is lacking, such as in some pathologies (multiple sclerosis), conduction delays increase or sometimes transmission is blocked. In this cases, defective transmission leads to defective movement control.

These results suggest that beside the actual stimulation, coming through the sensory channels, some endogenous “devices” might be essentials for proper learning to take place.

2.8 Discussion

This chapter reviewed some of the characteristics of the learning problem in the light of the developmental approach we proposed in chapter 1. In particular, we showed that biological systems, of course, are equipped to deal with the troublesome situations outlined in the previous sections.

For instance, bias and variance might be balanced by growing networks, but also by employing a suitable initial state; it can be seen as a general problem concerning learning either at the network level or at the group of networks (areas). We pointed out that gathering the training set is not an easy task because, in spite of the huge amount of information coming through the sensory channels during the interaction with the environment, the learner has to balance the resources devoted to exploration and those dedicated to exploitation. Noise, in various forms, improves this exploratory tendency by keeping the “plant” excited enough, so that identification can be efficiently carried out.

In many cases, it would be also hard to explore the whole state space, because it might require exponential time with respect to the number of dimension – a biological system, in an ecological context, would be dead ages before it could complete the exploration.

One point, which emerged during this discussion, is that the internal organization of the learner must be flexible. We argued that this “high level organization” should be explicitly designed together with the artificial learner, because eventually it might be a key factor for its successful behavior (survival).

3 A developing robot

This chapter outlines Babybot's adaptive control structure, which has been modeled based on the developmental framework introduced in previous chapters. As mentioned before, the idea is to have a system, which at "birth" uses only simple controllers – the initial configuration – and successively "grows" by employing more sophisticated modules (e.g. an inverse model, maps, etc). This sequence of successive developmental events was described in section 1.6.1; for clarity, we repeat it here (see Figure 9 below). The overall system initial state is thus characterized by a small number of free parameters, which can be easily estimated on-line. The concurrent controllers then learn on the basis of how the simpler loops are behaving. State space exploration is driven by additive noise, which simulates defective command generation (muscle control). Exploration and exploitation processes are carried out in parallel; in practice, the robot performs system identification and control at the same time. After eye movement control reaches a reasonable level of performance, the robot starts moving more degrees of freedom (i.e. the neck). Even at this level we stressed the biological parallelism by adopting a schema, which closely resembles the solution found in many species, including humans. We equipped the robot head with an inertial sensor, simulating the vestibular apparatus, which can sense the rate of rotation of the head respect to a vertical axis. Beside the extension of the working bandwidth as shown in (Panerai et al., 2000), the use of the inertial information also simplifies command generation. As we shall see, the system can work under the hypothesis that eyes and head controllers are decoupled, the vestibulo ocular reflex (VOR) carries out the necessary compensation. Concerning vision, the system employs space-variant images (Sandini & Tagliasco, 1980), (Sandini et al., 1981), which resemble the distribution of the photoreceptors in the human retinas (a brief description of the space variant geometry is contained in section 5.1). Image resolution is kept at minimum (images are 64×32 pixels), with the general idea of starting the robot with limited sensory capabilities. Without entering into the details of the actual visual processing (described in the appendix), we can say that the robot is able to compute the presence of a target and extract its position and velocity relative to the cameras (in retinal coordinates). The sensory abilities are complemented by

proprioception, which is provided by optical encoders (one for each mechanical joint). The control variables of the robot are the joint velocities. An appropriate low-level closed loop controller (usually a PID controller for each joint) generates the proper motor driving torques. An exception is the arm control schema, which is based on the equilibrium point hypothesis (EP). As in biological systems, the position and the impedance characteristics of the robot arm are the result of the interaction of the stiffness control of spring-like simulated actuators (muscles).

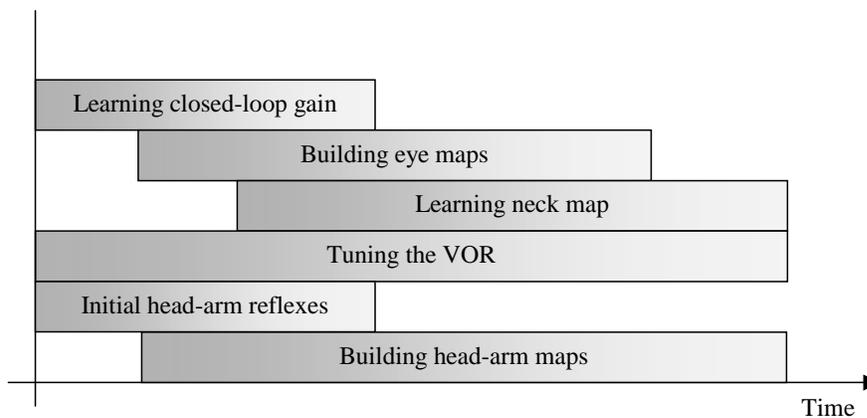


Figure 9 The developmental stages. The diagram above roughly shows the interleaving of the developmental stages; abscissa represents time. The first step is the acquisition of the closed loop gains; reflex-like modules control the arm sub-system. After a while, learning of the saccade control begins. Whatever movement of the robot also stimulates the inertial sensor: this information is used to tune the VOR. Eventually the eye-head coordination is acquired together with a more effective head-arm coordination map.

3.1 The noisy initial configuration

As starting point, consider the problem of moving the eyes toward the target. The simplest solution might use positional information to drive a negative feedback loop. The fundamental problem in such a strategy is that of converting the target position, which is expressed in retino-centric coordinates into motor commands. The latter are expressed with respect to

a motor (or joint) coordinates system. If this is the case the error is described by¹⁰:

$$\mathbf{e} = \mathbf{C} \cdot \mathbf{s}(t) \quad (3)$$

where \mathbf{e} is the position error expressed in motor coordinates, $\mathbf{s}(t)$ the retinal error and \mathbf{C} a coordinate conversion matrix. The matrix \mathbf{C} must be designed in order to stabilize the closed-loop system. In this case the generated motor command is:

$$\dot{\mathbf{q}} = -\lambda \cdot \mathbf{e} \quad \lambda > 0 \quad (4)$$

with $\dot{\mathbf{q}}$ the control variable and λ a positive constant gain. \mathbf{C} can be determined imposing an exponential decay rule of the error, which is guaranteed if (and only if) $\mathbf{e}\dot{\mathbf{e}} < 0$. Taking the derivative of \mathbf{e} and substituting yields:

$$\mathbf{e}\dot{\mathbf{e}} = \mathbf{e} \frac{\partial \mathbf{e}}{\partial \mathbf{s}} \frac{\partial \mathbf{s}}{\partial \mathbf{q}} \dot{\mathbf{q}} = -\lambda \mathbf{C} \frac{\partial \mathbf{s}}{\partial \mathbf{q}} \|\mathbf{e}\|^2 < 0 \quad (5)$$

which is verified if $\mathbf{C} \frac{\partial \mathbf{s}}{\partial \mathbf{q}} > 0$. A possible choice of \mathbf{C} is thus:

$$\mathbf{C} = \left(\frac{\partial \mathbf{s}}{\partial \mathbf{q}} \right)^{-1} \quad (6)$$

Learning of the matrix \mathbf{C} is carried out by acquiring discrete samples of the variation of the retinal error $\Delta \mathbf{s}$ due to a variation of the joint variable $\Delta \mathbf{q}$. A least-square approach is used to compute the components of \mathbf{C} . Further, it is important to define how to obtain the samples. Performing random movements according to the following strategy can easily generate them:

$$\dot{\mathbf{q}} = -\lambda \mathbf{C} \mathbf{s}(t) + \boldsymbol{\eta}(\mathbf{0}, \sigma) \quad (7)$$

¹⁰ The following math is valid for either scalar or vector.

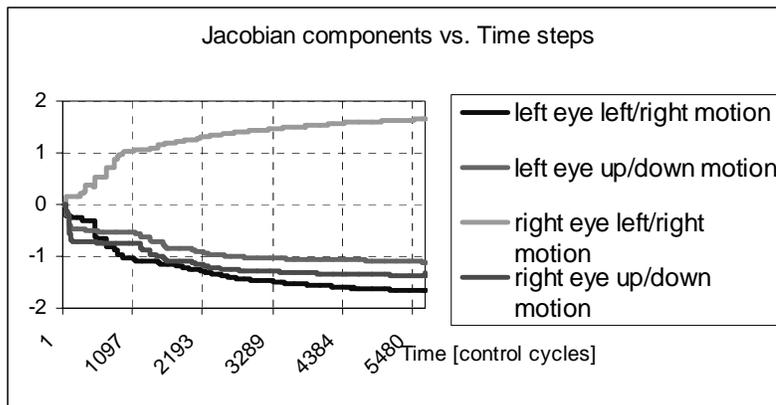
Box 3 Learning the closed loop Jacobian matrix as described in section 3.1. The relevant components of the two matrices controlling the eyes are plotted with respect to time. The point here is that the learning process is convergent. In general, considering a discrete-time case:

$$\Delta \mathbf{q} = \begin{bmatrix} j_{11} & j_{12} \\ j_{21} & j_{22} \end{bmatrix} \cdot \Delta \mathbf{s} \quad (8)$$

where j_{xy} are the unknowns, $\Delta \mathbf{s}$ the variation of image plane position (either the target or the background), and $\Delta \mathbf{q}$ the change of joint angle which caused the detected motion $\Delta \mathbf{s}$. Note also that $\Delta \mathbf{s}$ can be assimilated to the optical flow: $\Delta \mathbf{s}=(u_0, v_0)$. Equation (8) can be rewritten as:

$$\Delta \mathbf{q} = \begin{bmatrix} \Delta \mathbf{s}^T & 0 & 0 \\ 0 & 0 & \Delta \mathbf{s}^T \end{bmatrix} \cdot \begin{bmatrix} j_{11} \\ j_{12} \\ j_{21} \\ j_{22} \end{bmatrix} \quad (9)$$

and, by collecting at least two points – i.e. $(\Delta \mathbf{s}, \Delta \mathbf{q})$ – equation (9) can be solved for j_{11}, j_{12}, j_{21} , and j_{22} . The same procedure is applied for both eyes. The four traces presented are the only non-zero components; in fact, for our configuration the two Jacobian matrices are diagonal. These four quantities control the closed loop behavior of the three eye-related joints (q_3, q_4, q_5). Abscissa is time expressed in control steps (40ms each); ordinates are represented as *rad/pixel*.

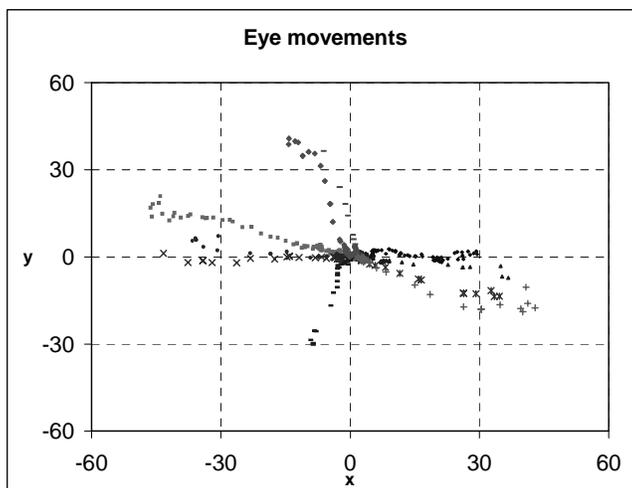


The first term is the closed loop formula described above; the second term η represents a zero mean uniform noise with standard deviation σ . It is worth noting that, at the beginning noise dominates ($C=0$, $\sigma \neq 0$), while as learning proceeds the closed loop term takes over the control of movements. Moreover, in order to guarantee convergence the standard deviation σ has to be reduced to zero. We applied the following criterion:

$$\sigma = k \frac{1}{t} \quad (10)$$

where t is the time. The value of k was determined experimentally in order to balance exploration duration and noise amplitude. Experimental results are presented in Box 3 and Box 4.

Box 4 Emergence of goal directed eye movements. We recorded several eye movement trials, with the robot working unrestrained – i.e. the task was simply to foveate some visually identified target – targets appeared randomly within the robot's field of view. The results from the first stage are plotted below, where abscissa and ordinates represent the image plane, and different graphical signs mark trajectories (the target position at each control step – 40ms period). As expected all the trajectories are converging to the fovea. This plot was obtained after the first stage of the development process as described in section 3.1. It is worth noting that in this case the movements are still quite slow, and the number of “points per trajectory” is high.



3.2 The cortical take-over process

Although the closed loop approach described above is effective, it does not mean it is also efficient. In fact, a closed loop approach would always lag behind a moving target. Moreover, if the perceived target lies in the periphery of the visual field, the robot would take several control steps to move the cameras toward it. Probably, for the same reasons, saccade movements evolved in those animals capable of moving their eyes. In our artificial system, the requirement for generating fast movements, emulating saccades, is to know exactly (as precisely as possible) the transformation between retinal error \mathbf{s} and the corresponding motor command $\Delta\mathbf{q}$, that is:

$$\Delta\mathbf{q} = \hat{\mathbf{f}}(\mathbf{s}) \tag{11}$$

Under the hypothesis of a stationary target and a closed loop control in place as described above, the gathering of training pairs (each of them has the form $(\Delta\mathbf{q}, \mathbf{s})$) is much simplified. The retinal error \mathbf{s} is acquired at the beginning of the motion, while the required motor command can be measured when the retinal error is zeroed. An explicit exploration is not actually needed because the closed loop system is already generating proper commands (directed at reducing the retinal error – and eventually zeroing it). In order to relax the stationary target hypothesis, it is possible to acquire a new training example as soon as some control cycles have been performed. In this case motion of the target would influence the measure of the motor command only a little. Furthermore, if we assume that targets generally move with equal probability in each possible direction, the mean of the measure error would be zero. The output of the proposed mapping is then used to generate saccades. A proper velocity command is generated by converting (deriving) its output $\Delta\mathbf{s}$ (i.e. $\dot{\mathbf{q}} \equiv \Delta\mathbf{q} / \Delta T$). Saccade initiation is controlled by another module, which issues a saccade command each time either the retinal error is greater than a fixed threshold (catch-up saccade) or a moving target is detected but a target is not currently being tracked.

The overall loop controlling eye motion is shown in Figure 10; note also that the same kind of control loop has been used for the eyes and the common tilt axis. Before proceeding further in the design of the controller, a few considerations concerning performance issues are worth noting. As comparison term, we can consider the direct inverse modeling approach,

which (in theory) might be able to solve the same problem. Although it could in fact learn the model used to generate saccades, the inverse modeling scheme had to be trained off-line. Furthermore, the training set should be generated prior to any actual control. Even intermingling control and training cycles would not remove the switching process. In our model, as well as in others (Kawato's feedback error learning – our model is similar but it uses position instead of velocity in the inverse model, and the error is not directly used to train the model), control and training are two parallel processes. This clearly fits much better in a developmental approach; besides, the most important property of our schema is that the process is intrinsically goal directed.

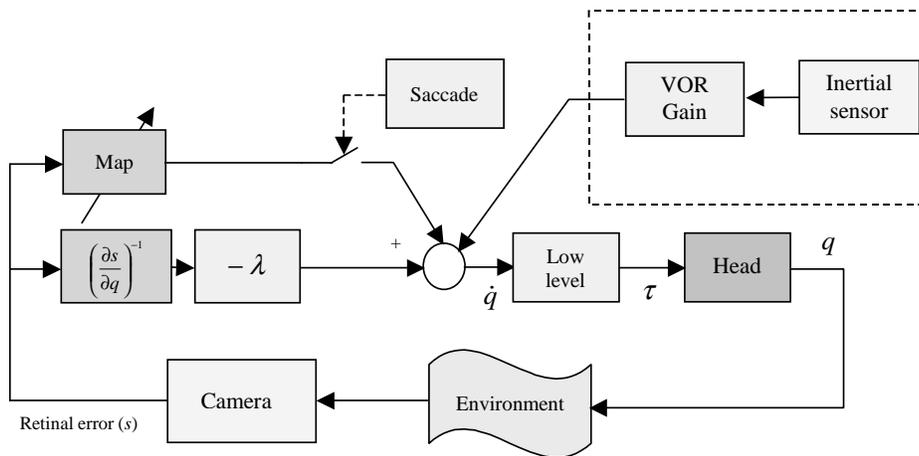
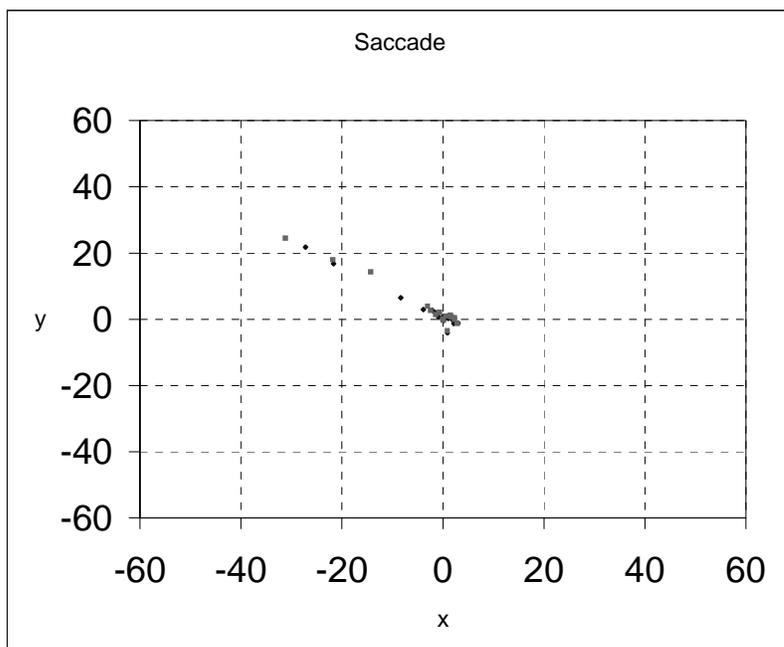


Figure 10 The eye control schema – simplified. It consists of a closed loop and a feed-forward secondary loop. The loop using the inverse Jacobian is derived from a classical visual servoing approach. The secondary loop consists of an inverse model (indicated by “Map”). It is activated whenever necessary – retinal error greater than a threshold – and generates a fast motion of the eyes in order to foveate the target. The goal of the network is to learn the inverse model. λ is a positive constant gain. It is tuned in order to obtain stability of the closed loop system. The input to the robot controller is a velocity command. A low level controller (PID) generates the motors’ driving torques τ . The block identified by “Saccade” is the governing logic (i.e. the threshold mechanism issuing the “start” signal for the fast motion).

Box 5 An exemplar trajectory after learning. Note as the first three steps are enough to reduce the retinal error to less than five pixels, afterwards the target remains in the fovea. This plot has been obtained after learning of the closed loop controller was completed and the saccade maps almost converged to a stable configuration. The two sets of points are relative to the left and right eye. In this case, it is clear that, the target appeared on the left side of the robot.



For a particular target (goal) or subset of targets, the system learns only the relevant subset of the control space. Probably this is not an issue for the simple mapping we considered here, but it might be an advantage if the number of dimensions is increased. The direct approach in this case, even for the single goal case, would explore the whole control space. The fact that the whole state space has to be (randomly) explored limits the speed of adaptation in cases where the system parameters are changed (consider for example the change of physical properties of the plant due to growth – link lengths, inertia, sensor resolutions, etc. – for artificial

systems change in mass distribution, unmodeled parameters, partial failures).

3.3 Controlling more degrees of freedom, eye-head coordination

Following this hypothetical line of developmental events, there is a stage when the neck comes into play in the orienting behavior. It is at the same moment when the proprioception becomes reliable and consistent. Gilmore et al. (Gilmore & Johnson, 1998) suggested that the shift might be also observed in the coordinates system governing these movements (from retino-centric to head-centric).

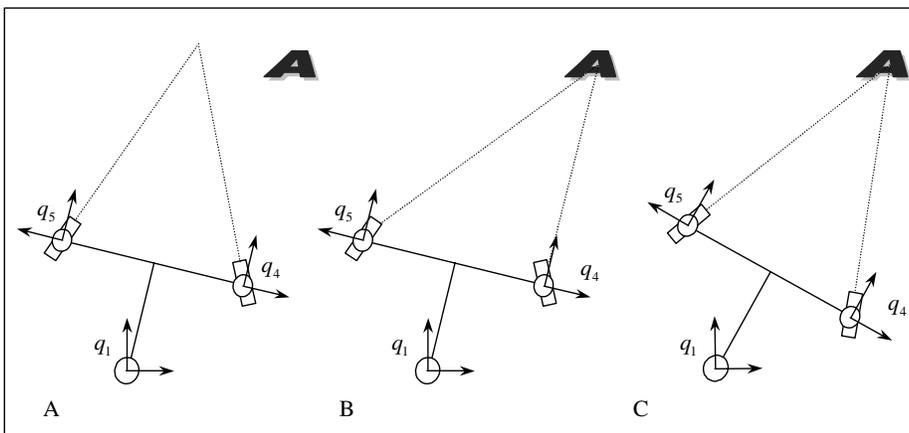


Figure 11 A hypothetical eye-head coordinated movement. The left panel (A) represents the initial situation preceding a saccade: a target indicated by the big “A” character appears within the robot’s field of view. The middle sketch indicates that even before any actual motion is started the robot computes the final eye positions; this efferent signal is the used to determine the required neck rotation. Once the appropriate commands are computed they are fed into the head low-level controllers, though because of the different inertias and programmed accelerations the eyes get to the target before the head motion is completed. On the other hand, because of the inertial sensor and the VOR the whole motion remains coordinated and eventually the head/eye system reaches a stable configuration as shown in panel (C).

Whatever motion strategy, it has to deal with the “degrees of freedom” problem. In fact, the same fixation point in the 3D world can be obtained using different configurations of the joint angles. The head-eye system is kinematically redundant; consequently, from the kinematic point of view, a further constraint has to be employed. We required the head system to achieve a symmetrical vergence configuration: that is, the neck should move in order to be roughly heading toward the target. A hypothetical situation is sketched in Figure 11.

Each degree of freedom was allowed to respect its physical constraints: roughly speaking, the eyes move faster than the neck because of the different inertias. Limiting the accelerations appropriately also enforced this behavior. Concerning neck motion control, the proposed two-loop system can deal with the situation. A PD controller governs the closed loop module as before. Its goal is that of zeroing the difference between the two eye angles:

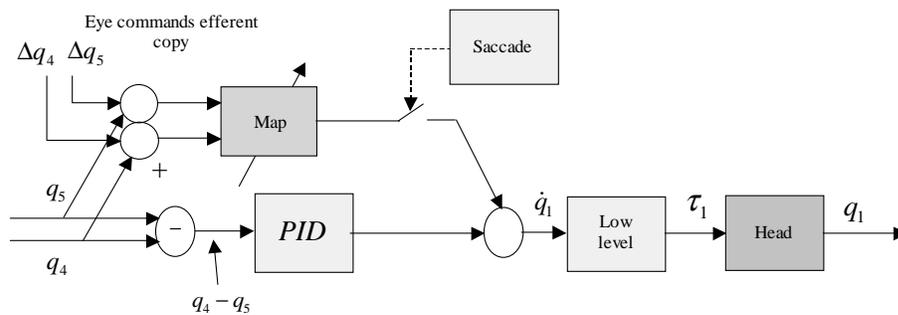
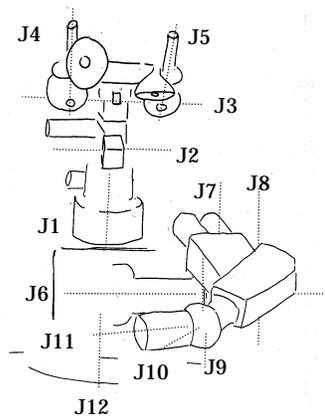
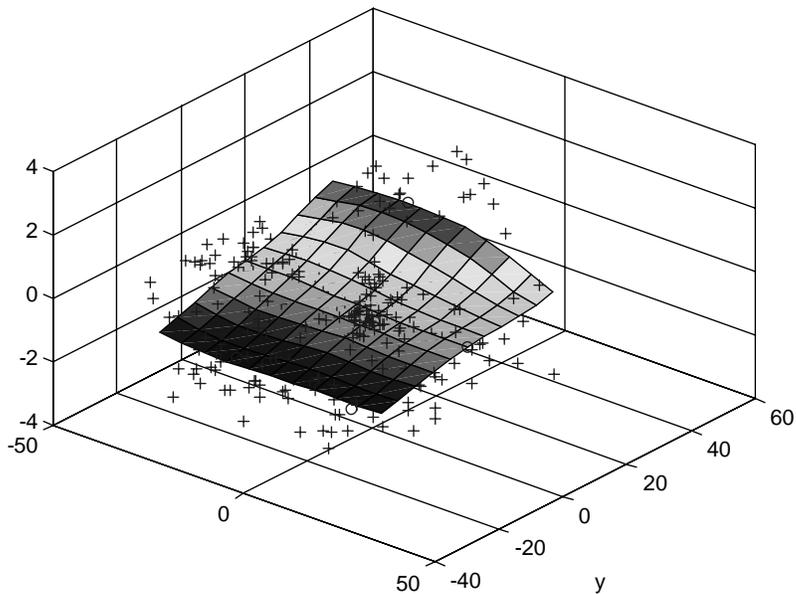


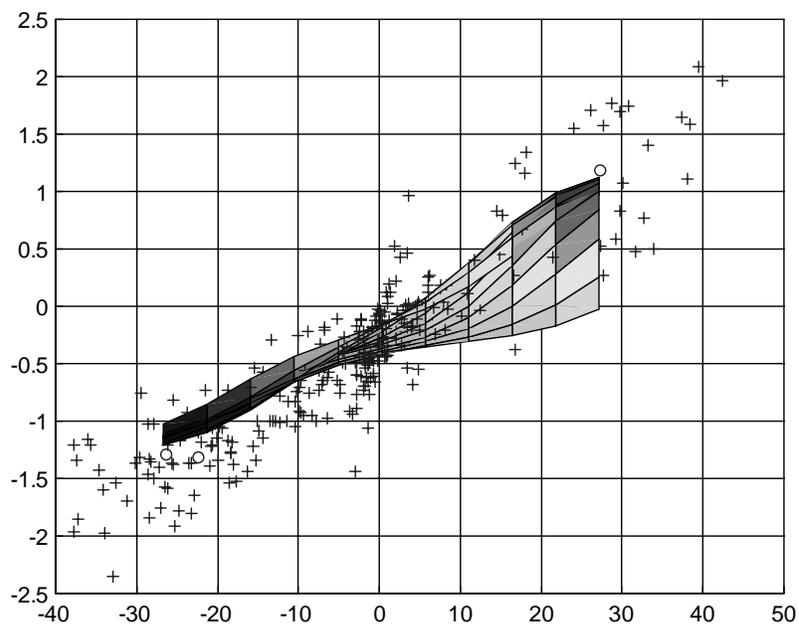
Figure 12 The neck control schema. It employs the same working principle of the eye controller. However there are a few important differences. First, there is not direct visual feedback, on the contrary, eye positions drive the movement of the head – the PID controller has to move the head in order to maintain a symmetric vergence configuration as much as possible. Second, the saccade-like movement is based on the prediction of the eye positions at the end of the saccade – i.e. efferent copy. Δq_5 and Δq_4 are the output of the eye maps; they are combined with the actual eye positions to get a prediction of the eyes’ orientation, this eventually allows estimating the required head rotation.



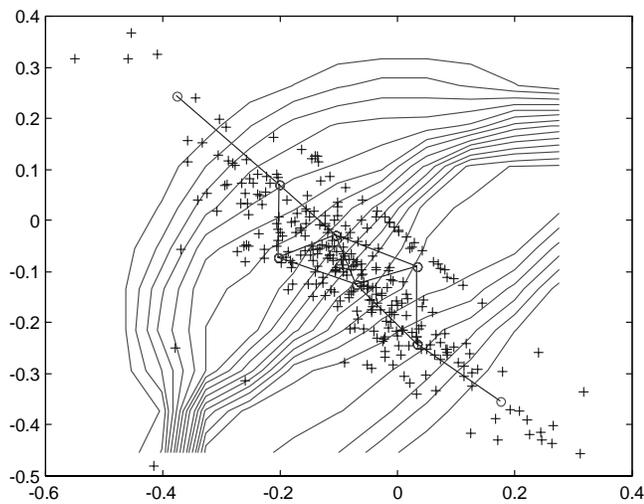
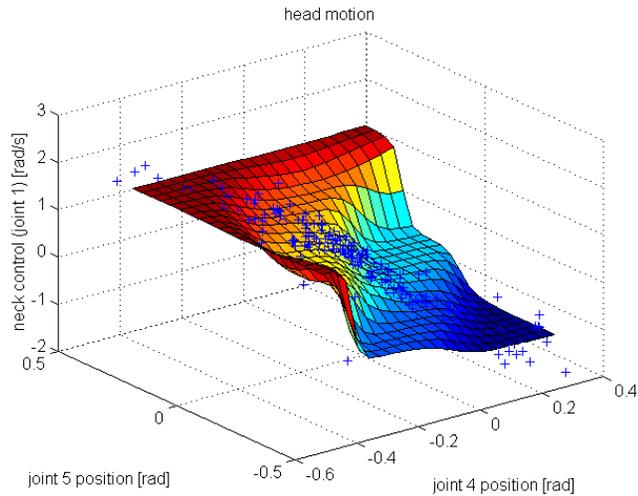
Color plate 1: the experimental setup consists of a five degrees of freedom robot head, and an off-the-shelf six degrees of freedom robot manipulator, both mounted on a rotating base: i.e. the torso. The kinematics resembles that of the upper part of the human body although with less degrees of freedom. From the sensory point of view, the Babybot is equipped with two space-variant cameras microphones for acoustic localization, an inertial sensor simulating the vestibular system, and proprioceptive information through motor encoders.



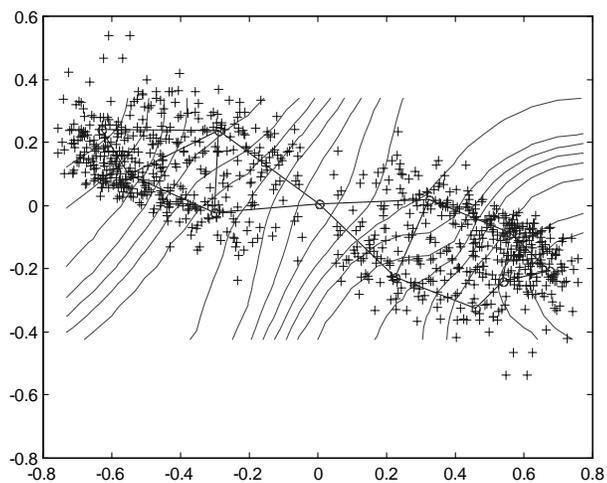
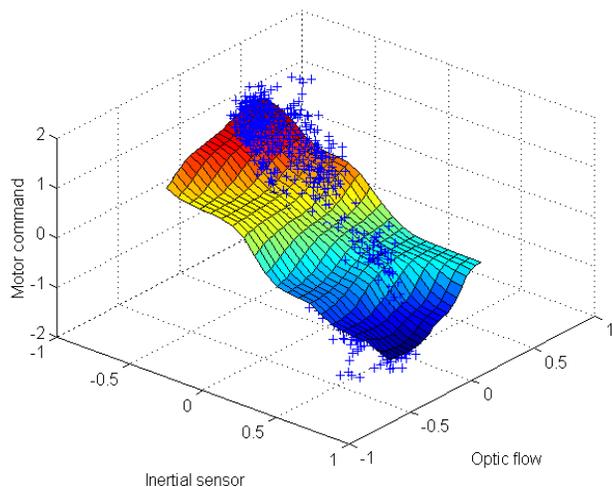
Color plate 2: a view of the left eye map (see also Color plate 3). The “+” sign represents the most recent 300 samples of the training set, and the circles the position of the unit’s centers. The plot has been obtained after about 90000 steps performed using the most recent 300 samples from the training set. The input space (x,y) is the image plane in Cartesian coordinates (bear in mind that the actual data are acquired in the space variant log-polar plane described in section 5.1), the output (the height of the surface plot) is the angle required to foveate a target appearing in the corresponding (x,y) image position.



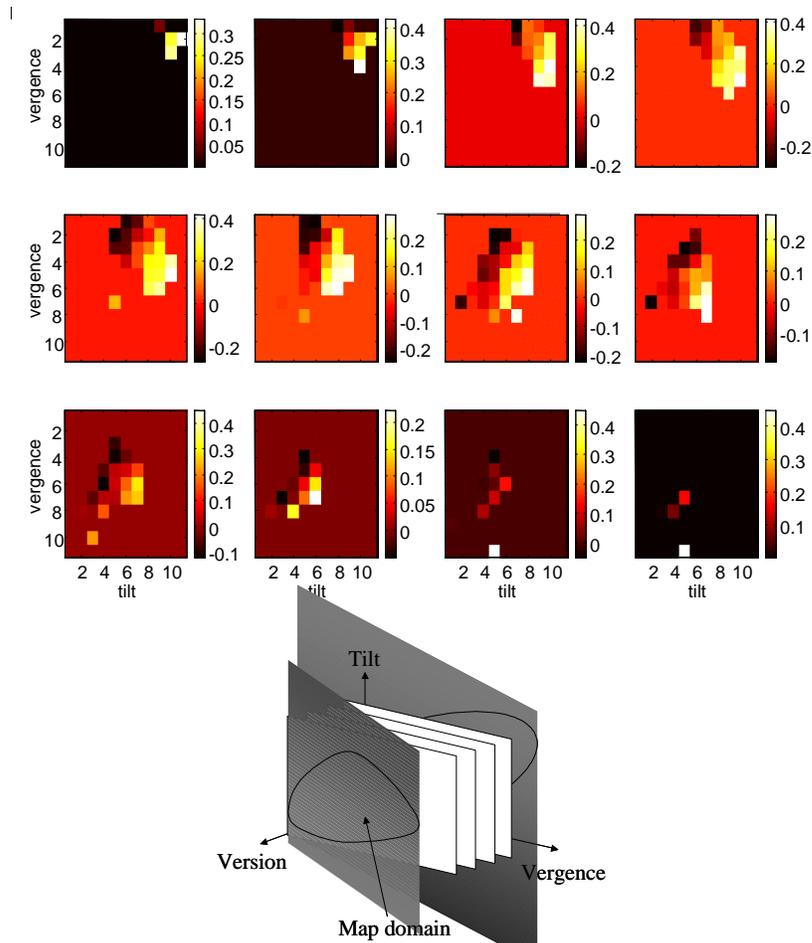
Color plate 3: a different view of the left eye map (see also Color plate 2). The “+” sign represents the most recent 300 samples of the training set, and the circles the position of the unit’s centers. The plot has been obtained after about 90000 steps performed using the most recent 300 samples from the training set. The input space (x,y) is the image plane in Cartesian coordinates.



Color plate 4: the head control map. The output is the required velocity command (approximately the angle multiplied by the control rate); the input is the predicted position of the two eyes as described in section 3.3. The bottom panel shows a 2D plot of the neural network. The “+” signs are the training samples, and circles stands for the positions of the units; the solid lines represent the contour lines.



Color plate 5: learning the vestibulo-ocular compensation: the VOR map. The graph above shows the VOR network output after about 10000 learning steps. Note that because of the dependence on the optic flow, the map allows taking into account the visual input appropriately. There is also a dependence on the inertial signal as expected.



Color plate 6: reaching lookup table. The figure above shows one component of the reaching lookup table after more than 1000 reaching trials. In order to display it (see sketch at the bottom), the following procedure has been applied: i) the input domain has been divided into a regular grid (for a total of $11 \times 11 \times 12$ cubes); ii) 12 slices of varying “version angle” are plotted as 11×11 2D maps; iii) the color intensity represents the output of the map controlling joint 6 (shoulder); iv) the outputs of all units falling into the same cube have been averaged. The output is the position of the EP in joint space, which is expressed in radians.

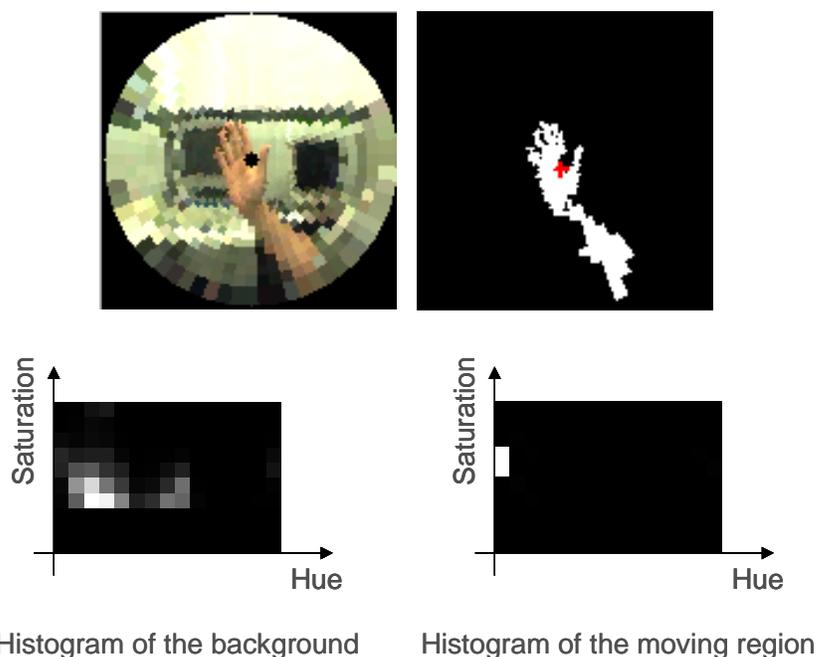


Original image



Log-polar image

Color plate 7: an example of log-polar mapping, note as radial structures in the flower (petals) map to horizontal structures in the log-polar image. Circles, on the other hand, map to vertical patterns. Furthermore, note as the central part of the flower occupies about half of the corresponding log-polar image.



Color plate 8: color processing. The upper row shows a typical image from the robot's point of view: original image (left) and color segmented image (right). All the processing is carried out in the log-polar domain; images are mapped back to the Cartesian space for visualization purposes. The lower row contains the HS histograms: background (left) and object (right). Note that, as the histograms do not overlap, segmentation can be performed reliably. Hue values range from 0 (red) to 360. Saturation ranges from 0 (gray level) to 1 (full color).

$$\dot{q}_1 = PD(q_4 - q_5) \quad (12)$$

Also in this case an “inverse model” map can improve performances. It maps the predicted eye positions to the proper neck motion. The control diagram is sketched in Figure 12. Formally:

$$\Delta q_1 = \hat{f}({}^{pred}q_4, {}^{pred}q_5) \quad (13)$$

where Δq_1 is the neck motion command, ${}^{pred}q_{(4,5)}$ the predicted eye positions. What does “predicted eye positions” mean? They are the current eye positions updated by the saccadic eye motion. The eye movement can be recovered using the eye maps, even before any actual motion has been started. In equation form:

$${}^{pred}q_{(4,5)} = q_{(4,5)} + {}^{saccade}\Delta q_{(4,5)} \quad (14)$$

This sort of signals ${}^{saccade}q_{(4,5)}$ is known in biological literature as an “efferent copy”. Despite the simple case where the two addenda of equation (14) are in the same reference frame, the use of efferent copy signals raises a series of other concerns, which are outside the scope of this book – such as multisensory integration (Morasso & Sanguineti, 1997). Eventually, both eye and neck commands are generated and fed into the controller at the same time.

It is worth noting that, although the proposed neck motion strategy can work, there are still some performance related issues to consider; in fact, the motion of the head is likely to disturb the eye movement process (either saccade or tracking). This is especially true if the neck is performing relatively fast movements. In that case, by applying the described control strategy, the robot would likely overshoot the target. The overshoot is eventually compensated by the visual feedback. However, vision is slow compared to the kind of motion we are dealing with (i.e. a saccade might last from 40 to 100ms). Thus the resulting motion, though convergent (i.e. stable), would have poor performance (oscillations).

We may observe that many species developed dedicated sensory systems devoted to measuring the motion of the head/body in space. We equipped our system with a similar device: an inertial sensor (a solid state gyroscope), as described in (Panerai & Sandini, 1998). In order to keep

things simple, only inertial information was used (this is a simplification because the brain actually integrates visual, proprioceptive, and inertial information (Gdowski & McCrea, 1999), (Crowell, Banks, Shenoy, & Andersen, 1998)). Our artificial vestibular sensor can measure neck angular velocity and, in the context of head-eye coordination, it comes into play by counter-rotating the eyes whenever the head moves (VOR). The VOR loop is sketched in Figure 10 together with the saccade control schema. For the scope of this section, in order to analyze accurately the saccade behavior, only the simple constant-gain VOR case is considered. In practice, even this simple schema improves the robot performance considerably. More sophisticated strategies are considered in details later (Panerai et al., 2000).

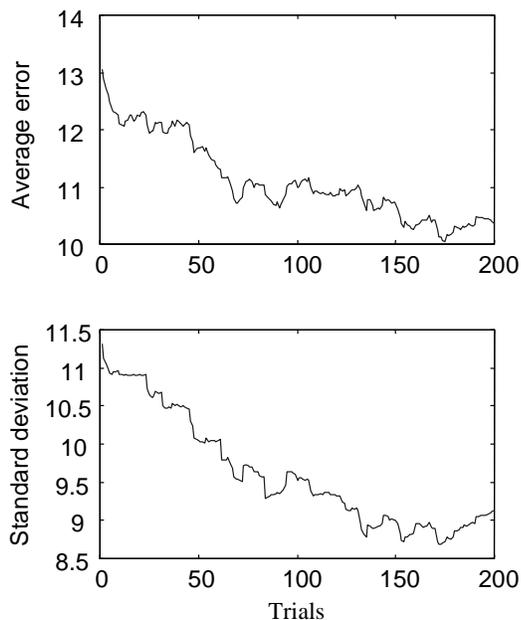


Figure 13 Robot motor performances. The upper plot shows the moving window average of the residual retinal error (i.e. at the end of a saccade). The lower plot is the standard deviation of the same 300 samples. Abscissas represent the number of trials. Note also that the error is computed over the space-variant geometry of the retinal layout; consequently they should have been plotted on an exponential scale rather than the linear one to take into account the compression due to the logarithmic sampling.

Neck tilt control is accomplished simply by using a PD controller as in the case of joint 1, that is:

$$\dot{q}_2 = PD(q_2 - q_3) \quad (15)$$

The behavior is roughly the same as joint 1; in fact, the PD controller links the two redundant degrees of freedom (eye tilt and neck tilt) as before. Although a vestibular compensation could be, in principle, applied also for up/down movements, at the moment our inertial sensor can only measure rotation around one axis (i.e. the vertical), consequently there is no such compensation for the tilt motion.

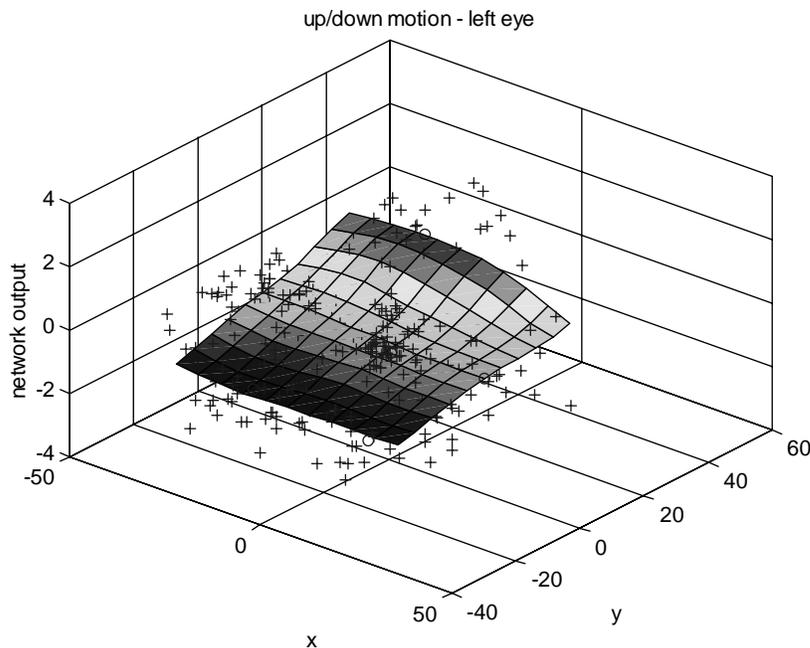
3.3.1 What the maps look like

As far as the gazing behavior is concerned, what we shall expect from the robot is that the average number of steps required to foveate the target should decrease over time – i.e. as the learning progresses saccades become more precise, and consequently the number of closed-loop steps after a saccade decreases. Indirectly, this is also shown in Figure 13, where the residual error after saccades is plotted versus time steps. The rationale is that: if the robot improves its performance, the retinal error, on average, decreases over time. We recorded from the robot for 300 trials in noisy conditions; that is, cluttered background, moving targets, etc. Of course the error on the single trial can be influenced by those unpredictable perturbations. It might depend also on the quality of the approximation on that particular point of the map. We thus applied a 100-sample wide moving window to the raw data, and estimated the mean and standard deviation inside the window. A clear trend toward the reduction of the average retinal error, as well as its standard deviation can be observed. More importantly, the time to acquire the target (not shown) is reduced (the better the saccade, the fewer control cycles are required to foveate a spotted location).

Another way of looking at these facts is by examining the trajectories of the target in the image plane. We again recorded several eye/head movement trials at different learning stages; the robot was working unrestrained – i.e. the task was simply to foveate some visually identified target – targets appeared randomly within the robot's field of view.

The results of the first stage are presented in Box 4, where abscissa and ordinates represent the image plane, and different graphical signs mark trajectories (the target position at each control step – 40ms period). As expected, all the trajectories are converging to the fovea. This plot was

obtained during the first stage of the development process, as described in section 3.1. It is worth noting that in this case the movements are still quite slow, and the number of “points per trajectory” is high. An exemplar trajectory after some learning is shown in Box 5. Note as the first three steps are enough to reduce the retinal error to less than five pixels, afterwards the target remains in the fovea. This plot has been obtained after the learning of the closed loop controller was completed, and the saccade maps almost converged to the configuration shown in Figure 14 and Figure 15. The maps are implemented by using the Growing Neural Gas algorithm described in section 5.5.



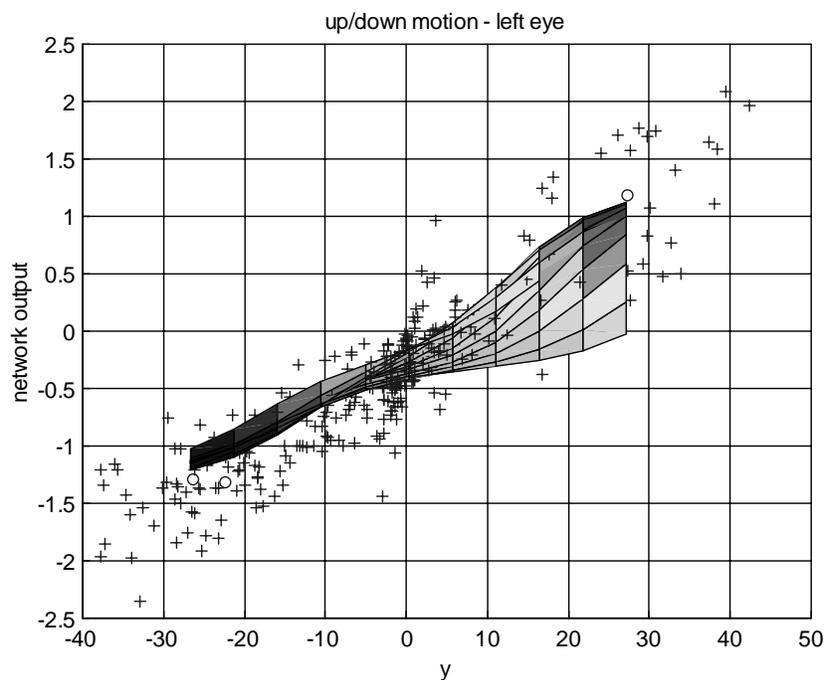


Figure 14 Two views of the left eye map. The “+” sign represents the most recent 300 samples of the training set, and the circles the position of the unit’s centers. The plot has been obtained after about 90000 steps performed using the most recent 300 samples from the training set. The input space (x,y) is the image plane in Cartesian coordinates (bear in mind that the actual data are acquired in the space variant log-polar plane described in section 5.1), the output (the height of the surface plot) is the angle required to foveate a target appearing in the corresponding (x,y) image position.

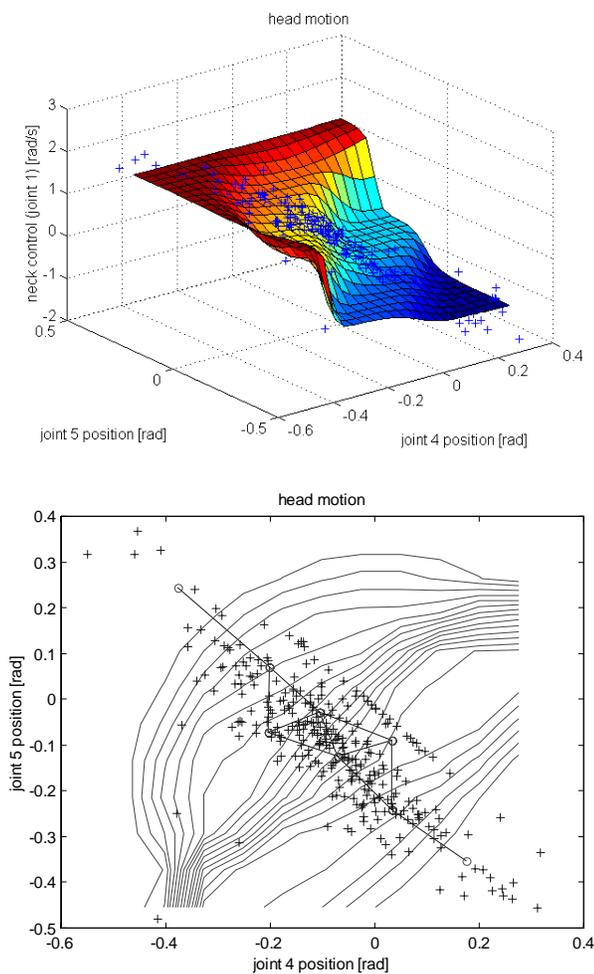
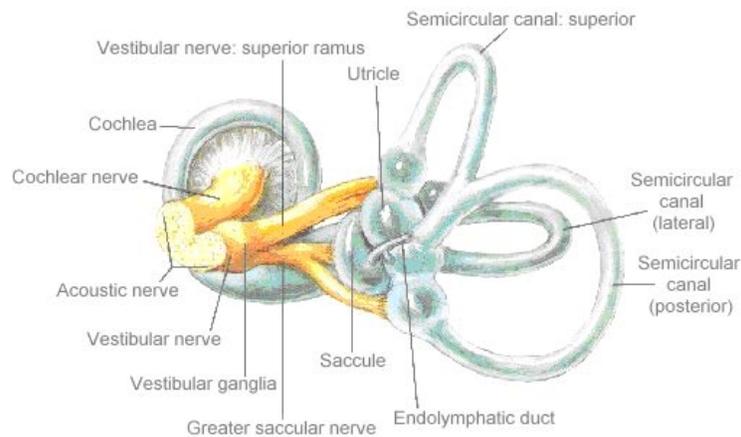


Figure 15 The head control map. It is obtained after about 30000 control cycles. In this case the output is already the required velocity command (approximately the angle multiplied by the control rate), the input is the predicted position of the two eyes as described in section 3.3. The bottom panel shows a 2D plot of the neural network. The “+” signs are the training samples, and circles stands for the positions of the units; the solid lines represent the contour lines. Note also that the upper-right quadrant is empty, because it corresponds to divergent-eye configurations.

Box 6 Animal vestibulo-ocular reflex. In animals with fixed eyes, like many insects and some birds, compensatory head or body movements produce retinal image stabilization. Primates and many vertebrates with an efficient oculo-motor apparatus, rely mostly on compensatory eye-movements. As a matter of fact, the “hardware” triggering compensatory motor responses is common to many biological species. A wide range of mechano-neural transducers, functionally equivalent to rotation- and translation-sensitive mechanisms, are found in many species (Wilson & Jones, 1979). One can speculate about the advantages of these particular motion sensing “transducers”, but nevertheless it remains that such a particular design solution has been naturally selected to deal with the image stabilization problem. In primates, the mechanism controlling the direction of gaze on the basis of inertial information is called Vestibulo-Ocular Reflex (VOR). It is subdivided into angular VOR (AVOR) – generating oculo-motor responses to angular head motion – and translational VOR (TVOR) – generating responses to linear head motion (Paige, 1991), (Schwarz, Busetini, & Miles, 1989). In the case of the AVOR, three ring-shaped sensors (called semi-circular canals) sense angular velocities along three perpendicular directions. In the case of the LVOR, the sensing is performed by the otoliths organs, which sense linear movements in horizontal and vertical directions, and orientation of the head with respect to gravity (Kandel et al., 1991). The vestibular reflexes are known to operate in open-loop, are very rapid and work best for high frequency movements of the head (Keller, 1978), (Benson, Guedry, & Melvill Jones, 1970), (Wilson & Jones, 1979). On the other hand, the visual reflexes, like the Opto-Kinetic Reflex (OKR), operate in closed loop, they are slower and respond better for lower frequencies of head movements (Baarsma & Collewyn, 1974), (Micheal & Jones, 1966). The human sensory apparatus is shown below.



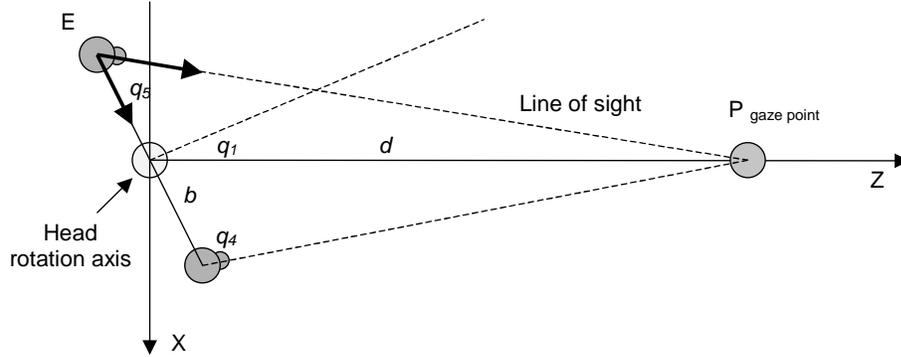


Figure 16 Geometry of the head-eye system showing the parameters relevant for the inertial and visual measures. This is the reference model for the kinematic analysis of the stabilization as described in text. P is the gaze point at distance d from the head rotational axis (i.e. the neck). b is the interocular distance or baseline.

3.4 Tuning the Vestibulo-Ocular Reflex

We consider here the compensatory eye movements required to maintain stable fixation of a target at distance d , when the head rotates around a vertical, off-centered axis. Figure 16 shows the geometry of our binocular system for this case, and indicates the most relevant geometrical parameters: the inter-ocular distance (or baseline) b , and the viewing distance d , measured from the head rotational axis to the gaze point. The analytical relation among these parameters can be derived by considering the kinematics of this model, and by imposing the constraint that the eye E maintains gaze at point P when the head rotates (see (Panerai et al., 2000) for details). Simple vector rules and differentiation with respect to time leads to the following expression of angular velocity ω_5 :

$$\omega_5 = \left[1 + \frac{dZ_1 - \frac{b^2}{4}}{d^2 - 2dZ_1 + \frac{b^2}{4}} \right] \omega_1 \quad (16)$$

where $Z_l = \frac{1}{2}b\sin(q_l)$ represents the Z-coordinate of the left eye. Equation (16) determines the relationship between eye velocity ω_5 , and i) the geometrical parameters of the eye-head system (i.e. b) and ii) the distance d of the fixation point P , for any given velocity ω_l . Equation (16) also makes explicit the inverse dependence upon distance. Interestingly enough, dependence upon distance is clearly evident in primate's RVOR responses (Biguer & Prablanc, 1981), (Snyder & King, 1992), (Viirre, Tweed, Milner, & Vilis, 1986), (Hine & Thorn, 1987), (Crane, Viirre, & Demer, 1997), (Telford, Seidman, & Paige, 1998). These findings suggest that fixation distance might play an important role in simplifying/synthesizing efficient oculo-motor responses to rotational movements, especially in the close range domain. In this regard, note that the eye velocity, ω_5 , required to maintain fixation on near objects can be as much as twice the value of ω_l , for fixation distances in the range 25-200cm; in other words, the optimal amount of ocular compensation needed to achieve a stable gaze can change rapidly with distance of fixation.

One point worth stressing here is the fact that, the range over which fixation distance influences the optimal control of the eye compensatory gain (at least from the kinematics point of view), may not be very relevant for locomotion (e.g. a robot walking/navigating and fixating at a long range). On the other hand, this range overlaps entirely with manipulation workspace and, in this respect, might justify appropriate control circuits (e.g. a robot manipulating objects).

When looking at an eccentric target in near space, the compensatory eye movements required to maintain binocular alignment during head rotations are different for the two eyes (Hine & Thorn, 1987). From the kinematic point of view, the origin of this asymmetry is clear if one compares the analytical expressions of the two angular velocities (i.e. left vs. right eye). The optimal compensatory velocities are given by equation (16) for the left eye and by the following for the right eye:

$$\omega_4 = \left[1 + \frac{dZ_r - \frac{b^2}{4}}{d^2 - 2dZ_r + \frac{b^2}{4}} \right] \omega_1 \quad (17)$$

where $Z_r = -\frac{1}{2}b\sin(q_r)$ represents the Z-coordinate of it. The expressions are almost identical except for the sign of a few terms encoding the opposite position of the eyes. Equations (16) and (17) reflect the asymmetric

requirement of the optimal response. For example, in the case of an object at 30cm distance, an angular deviation from the frontal direction of 30deg introduces a relative gain difference between the two eyes of about 0.2; with a head velocity of 200deg/s , deviating 30deg from the frontal direction gives a relative angular differential velocity of 40deg/s . Thus, the angular velocities of the two eyes can be rather different in the near space. Although in humans there is clear evidence that, during compensatory eye movements, binocular alignment is not strictly maintained (Collewijn, Erkelens, & Steinman, 1995), this constraint might be more important for a robot vision system, especially if the system uses binocularly derived cues to control camera movements.

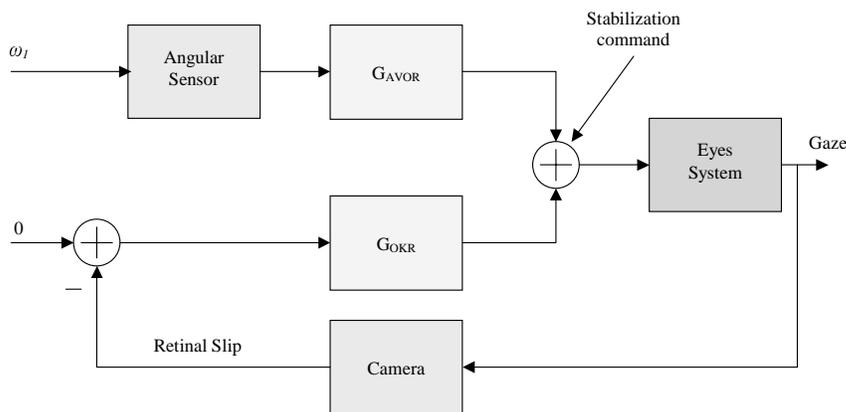


Figure 17 The simple method for image stabilization. The angular sensor measures the rotation of the head. Vision, in parallel, senses the retinal slip. The two sources of information are linearly combined (using gain G_{AVOR} and G_{OKR}) and fed to the head control system. The inertial information is processed open loop.

3.4.1 Visuo-inertial integration: the simple way

The issue of synthesizing simple and effective control strategies that integrate visual and inertial information appears challenging. In an anthropomorphic binocular system, the rotational axes of the eye and of the head do not coincide. Therefore, a rotational movement of the head causes both a rotation and a translation of the eyes. If the fixation point is at infinity, the inertial information alone will be, in principle, sufficient to stabilize gaze perfectly. However, this situation might represent a

minority of cases. In general, as we have outlined above, to generate an adequate compensatory response, the information about viewing distance is required. We suggest a control strategy exploiting range information to tune the stabilization performance. The structure of the basic visuo-inertial mechanism implemented is sketched in Figure 17. In the block diagram, the visual and inertial information are simply added together (in the biological literature this early modeling was named the “linear summation hypothesis”). The compensatory movement, generated by the inertial information, limits the amplitude of image motion to a range of values measurable by visual algorithms. Optic flow is therefore used to measure the residual error (called residual optic flow – ROF) present in each image after inertial compensation.

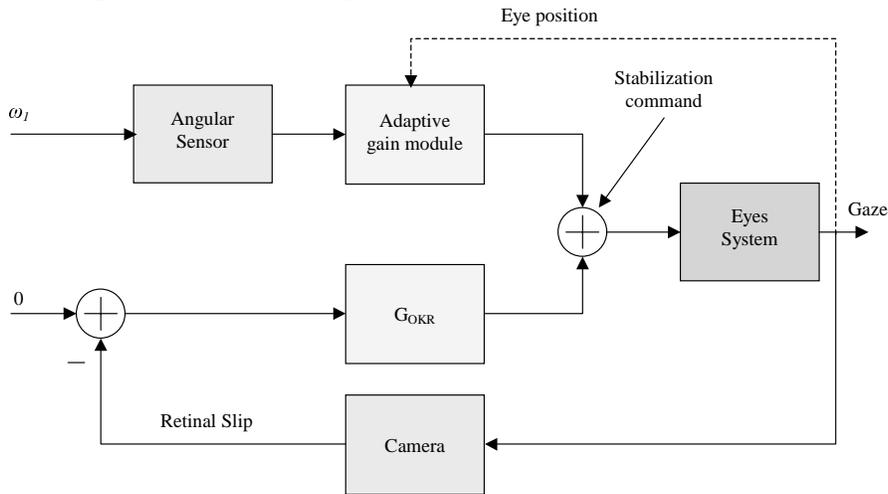


Figure 18 Adaptive tuning of compensatory gains. The compensatory gain of the eyes is tuned according to distance of fixation. The adaptive gain module simply replaces the constant one of the previous scheme (Figure 17).

In order to work optimally also in a context where the fixation point is changed in depth, a tuning scheme which adapts the compensatory gains to changes in distance is proposed. The scheme exploits range information, which is derived on the basis of proprioceptive information. The distance d of the object being fixated can be derived from the vergence and version angles, using the following equation:

$$d = b \frac{\cos q_4 \cos q_5}{\sin \theta_v \cos \theta_p} \quad (18)$$

where $\theta_p = \frac{1}{2}(q_4 + q_5)$ and $\theta_v = (q_4 - q_5)$. Distance information is in turn used to adaptively change the compensatory gains of both eyes. The diagram in Figure 18 describes the information flow, which implements this scheme. The tuning functions we propose to use at this stage are those obtained from the kinematic analysis. A different approach is described in the following section.

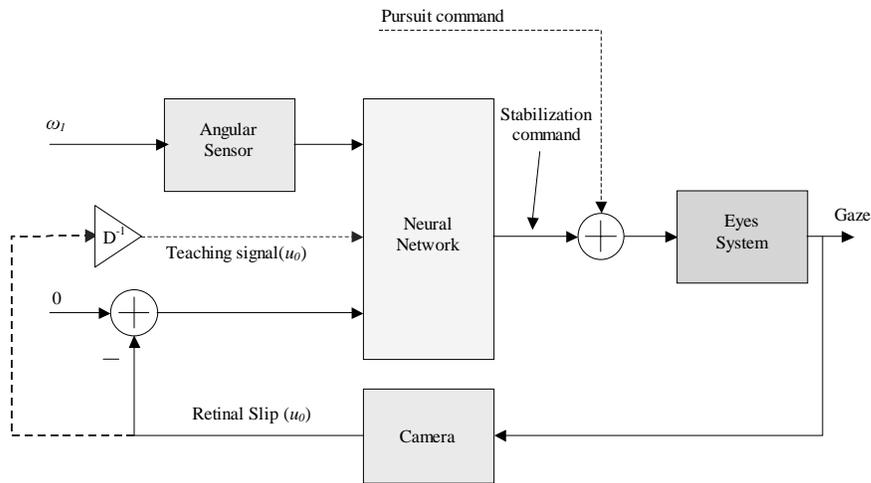


Figure 19 Learning compensation commands: VOR gain tuning. In order to generate an appropriate stabilization command, inertial information (from the sensor) and retinal slip (from optic flow) are combined. The neural network is responsible for building such association. The teaching signal is the optic flow itself, which has to be minimized for stabilization to be effective. It is worth stressing that for the schema to work, the robot must be interacting with a real environment; that is the system evaluates on-line performance (the residual optic flow) to update the network's parameters.

3.4.2 Tuning VOR: the hard way

The persistent association of vestibular signals related to head turns, and visual signals related to image motion, guides learning in biological systems. However, the two source of information are weighted differently, as the dynamics and latencies of such signals are different. Concerning the development of the VOR in humans, the reflex itself is known to be present at birth, although probably proper tuning (i.e. learning of the appropriate compensation command out of the vestibular and visual information) requires some gain adjustments and control (Finocchio, Preston, & Fuchs, 1991). Concerning the Babybot, as before, we envisaged two signals, which are relevant for image stabilization: the inertial signal and the retinal slip. The latter is measured through optic flow (OF) processing as described in section 5.2. The two sources of information are combined as shown in Figure 19. As a matter of fact, the network first collects the inertial signal and the OF then produces a suitable compensation command – that is eye velocities. Eventually, these compensation components are combined with either saccade or tracking commands. The learning process consists of incrementally adjusting the network parameters to improve/reduce a predefined “performance index” over time. To the extent of our application, this index is a first order approximation of the residual optic flow (ROF). In formulae, the network has to minimize (see also section 5.5 for a detailed description of the network architecture):

$$\min_{v_i} \left\| f_x \left(-\frac{T_x}{Z(0,0)} - \omega_1 \right) + \sum_i v_i U_i(\xi, c_i) \right\| \quad (19)$$

where f_x is the focal length of the camera, $Z(0,0)$ is the distance to the fixation point, ω_1 the head angular velocity, the sum over i represents the actual network output. ξ in this case is the vector (ω_1, u_0) . The first addendum of equation (19) is indeed the analytical expression of the optic flow obtained from the general flow equation for a moving observer, whose motion is constrained to rotations around a vertical axis and to translation along a horizontal, fronto-parallel axis – i.e.:

$$u_0 = f_x \left(-\frac{T_x}{Z(0,0)} - \omega_1 \right) \quad (20)$$

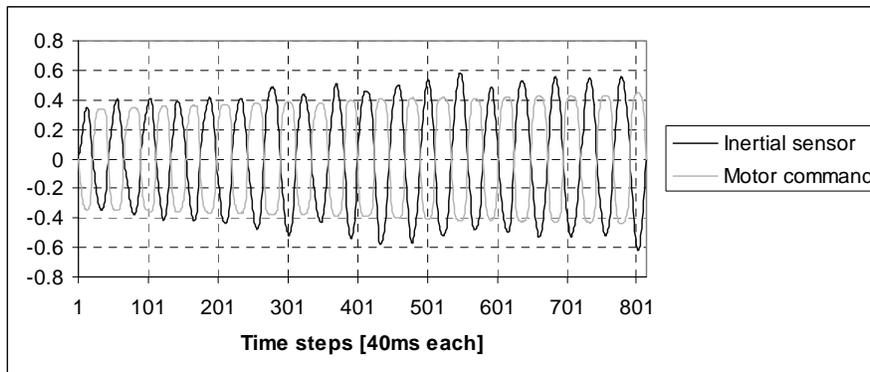
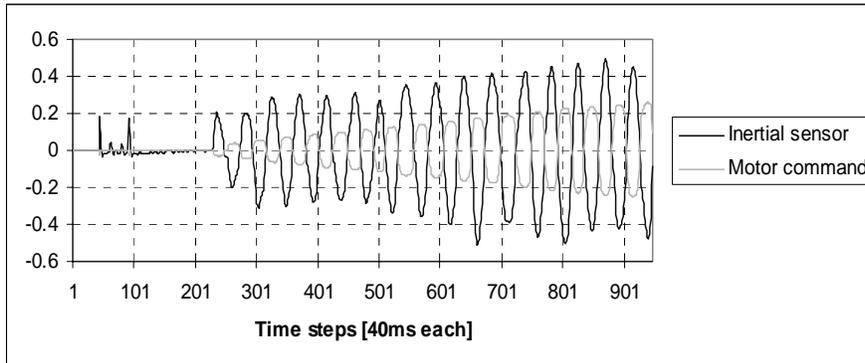
Note also the explicit dependence of optic flow on the fixation point distance $Z(0,0)$, and that the translational motion is due to the off-axis rotation of the eyes as a consequence of the head motion. In order to carry out the minimization, the network learning rule is modified as follows:

$$\Delta v_i = \eta_2 \left(\sum_j v_j U_j(\xi, c_j) + u_0 - v_i \right) \cdot U_i(\xi, c_i) \quad (21)$$

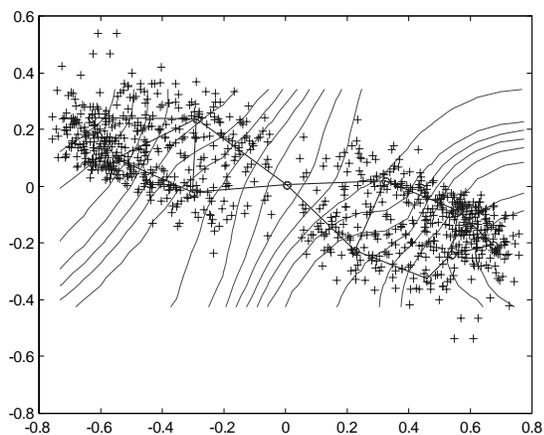
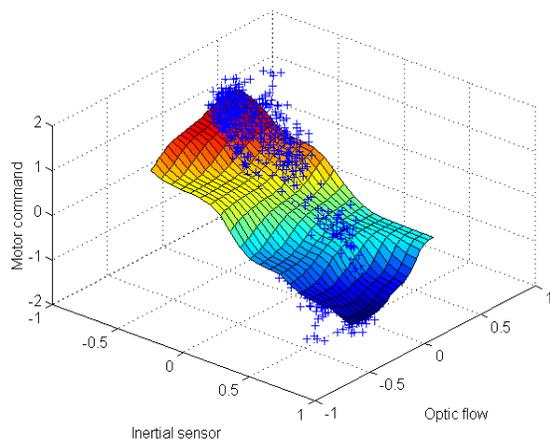
that is, the target output is shifted by the quantity u_0 from the current network output. Whenever, stabilization is perfect, $u_0=0$, no adjustment is necessary and in fact, $\Delta v_i=0$. It is worth stressing that time, which is not explicitly indicated in equation (21), plays a fundamental role in this schema. In fact, the optic flow used as input to the network is actually one time step before of that used as stabilization measure. That is, the measure of the network performance can be obtained only one step after the network has been used to generate a motion command. A delay line in Figure 19 indicates this last point.

Box 7 shows some experimental results obtained out of the proposed approach. Two graphs present the acquisition of a compensatory command at two successive stages of the learning process. The upper plot shows the very first moments of the learning; the lower one a phase where the neural network almost stabilized its output – the generated compensatory response merely grows. Some more experimental results are presented in Box 8 and Box 9.

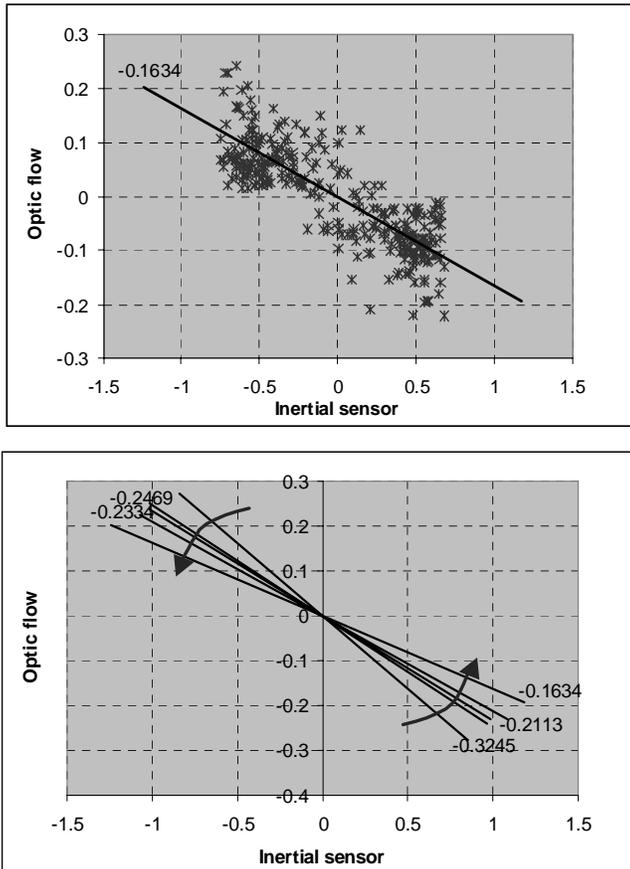
Box 7 Learning the vestibulo-ocular compensation. The plots below show how the compensatory eye commands are acquired. The robot is stimulated manually by random rotation of the torso. This generates both inertial and visual information. The neural network combines the two sources of information, and a compensatory command is generated in order to minimize the optic flow. The first plot shows the inertial sensor signal and the generated motor command (eye velocity) just at the beginning of training – both signals are normalized in the range ± 1 ; abscissa is time expressed as control cycles (40ms). The motor command in this first stage steadily increases because compensation is not yet effective. On a second plot, the same quantities are displayed after some training has taken place. In this case, compensatory motor response is still growing, but at a slower rate.



Box 8 Learning the vestibulo-ocular compensation: the VOR map. The graph below shows the VOR network output after about 10000 learning steps. Note that because of the dependence on the optic flow, the map allows taking into account the visual input appropriately. In practice, the system generates a higher (amplitude) motor command in situations where the optic flow is higher – i.e. when the retinal slip is high, it is reasonable to use a higher stabilization command. There is also a dependence on the inertial signal as expected.



Box 9 Learning the vestibulo-ocular compensation: minimizing the optic flow. In order to evaluate the learning process, we analyzed the ratio between the inertial information (i.e. the stimulus) and the optic flow (the stabilization performance). The rationale is that if the network is behaving correctly the optic flow should, on average, decrease as learning progresses. The analysis consisted of plotting the optic flow versus the inertial signal. This operation was repeated for 300 sample-long portions of data, extracted at different consecutive instants of time over the training period. The slope of the linear fitting of those data gives an indication of the ratio between stimuli and stabilization. The two pictures below show this analysis graphically for an exemplar set of 300 points (top), and for different portions of data (each of them 300 points long). As learning progresses the slope decreases showing that the optic flow is effectively minimized.



3.5 Saccade and VOR interaction

This section describes the role of inertial information during gaze redirection: that is, saccades. In particular, the results show that when the inertial information is used the robot behavior is characterized by a faster fixation and a smaller overshoot of the target.

In a series of repeated trials, a target is presented to the robot in fronto-parallel plane at a distance of 100cm. Three eccentric positions are chosen, respectively 5, 10, and 15 degrees eccentricity. These eccentricity values represent a reasonable sampling of the robot visual field given the focal length of the cameras (7.5mm in this experiment). The movement of the eyes is programmed as described in section 3.3. The saccade movements lasted about 4 control cycles ($30 \times 4 = 120ms$), and in this experiment only color information was used to extract target position and program the saccades. The stabilization performance is measured quantitatively in two cases: i) the robot generates compensatory eye movements by using the inertial information; ii) the inertial information is not used. The performance measurement is obtained in terms of: i) the target overshoot, that is, the transient target position error after the saccadic part of the eye control is completed; ii) the time interval required for a stabilization index (ISI¹¹) to fall below a given threshold – see section 5.3 in the appendix for the description of the stabilization index. In particular, overshoot is computed as the difference between the minimal retinal error and the maximal retinal error after the saccadic movement. The threshold for the ISI index was empirically determined to be 0.3. In fact, when ISI falls below this value, processing of image features and extraction of dynamic parameters lead to accurate and robust measurement. Figure 20 shows the dynamic trajectories of eye-head coordinated movement for a saccade performed at 15deg angular eccentricity. The measurements are obtained when inertial information is used. From top to bottom the trajectories represent: the left eye movement, the head movement, the angular velocity measured by the inertial system, and the corresponding gaze angle.

¹¹ Stabilization performances can be evaluated either by estimating image motion through optic flow processing or by using a sort of cross-correlation over time. The correlation measures the degree of similarity between two subsequent log-polar images. We employed a normalized correlation (ISI) as defined in appendix (section 5.3), so that good stabilization maps to ISI close to zero, and vice-versa poor stabilization is represented by ISI values close to one.

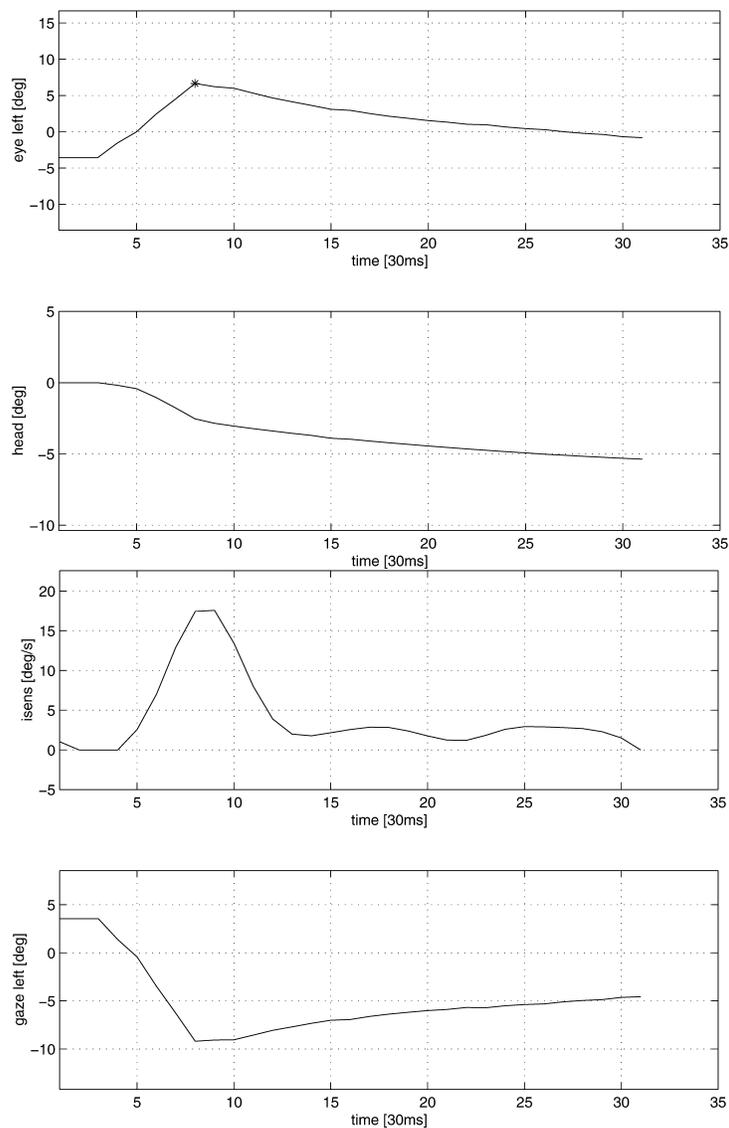


Figure 20 Position and velocity information during a gaze redirection experiment. From top to bottom: the left eye position, the head position, the inertial sensor output, and the gaze position are shown. The head velocity sensed by the inertial sensor is used to generate compensatory eye movements. The end of the saccadic part of the eye movement is marked with a '*' symbol.

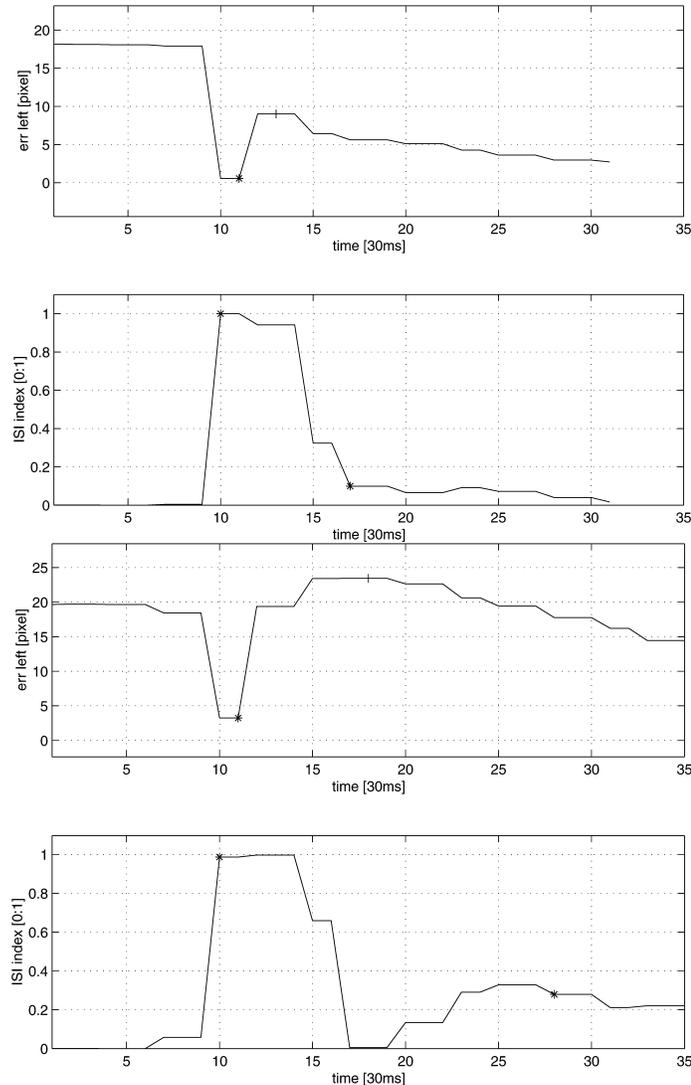


Figure 21 Visual parameters computed during gaze redirection. From top to bottom: target retinal error (left eye), image stabilization index (ISI) for the left eye, target retinal error (right eye), and ISI (right eye). The overshoot is measured as the difference between the minimum (*' symbol) and the maximum retinal error ('+' symbol) after the saccade. The time interval required for the ISI to fall below a stable threshold of 0.3 is delimited by two '*' symbols.

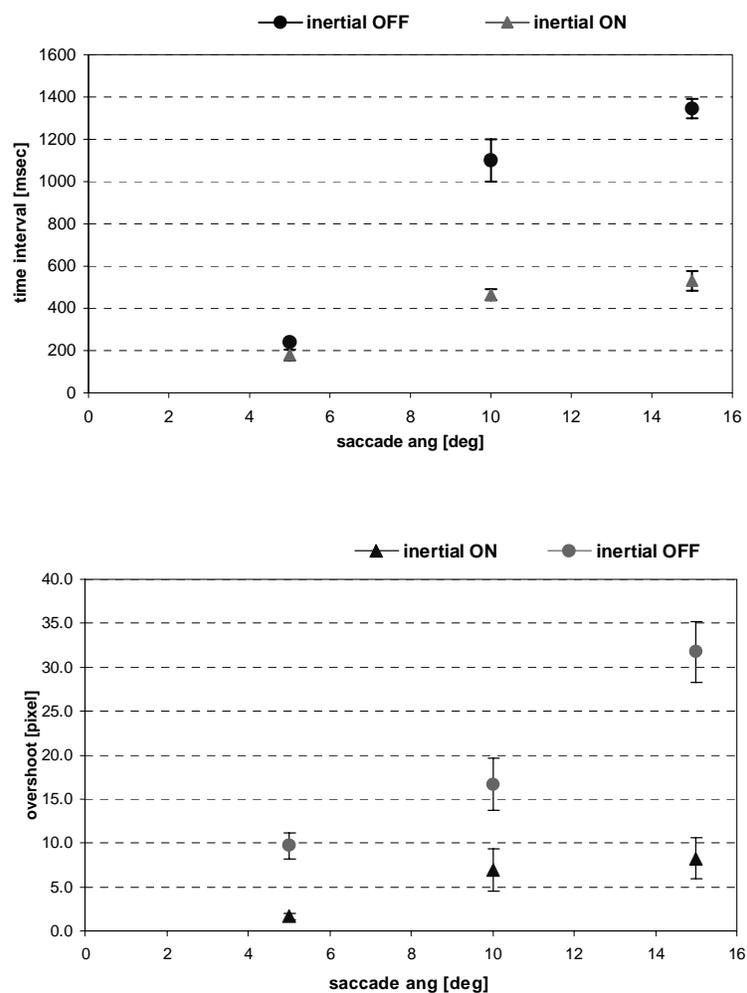


Figure 22 Parameters measuring stabilization performance during coordinate eye-head movements. Top: retinal error overshoot: the inertial and non-inertial case. Bottom: time interval required for the ISI to fall below the 0.3 threshold.

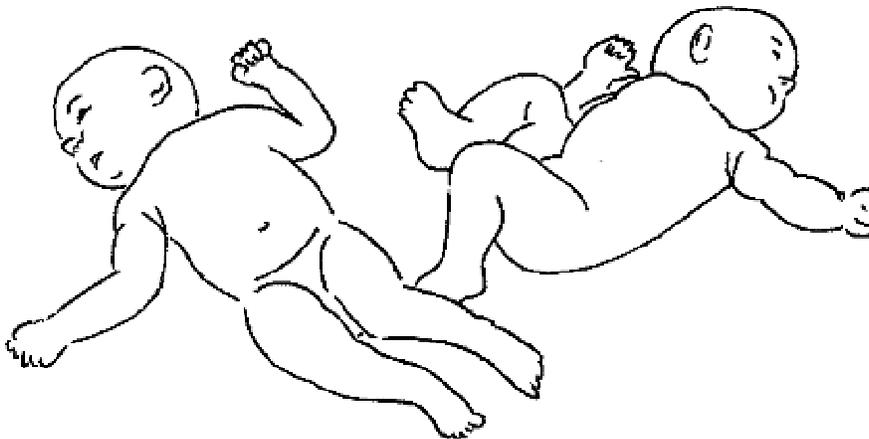
Figure 21 shows the target position and the ISI index measured during the same experiment. The comparison of the data in Figure 21 shows that: i) in the compensated case (left), the overshoot of the target is smaller than 12 pixels (see the difference between the marks ‘*’ and ‘+’ around the time unit 15); in the non-compensated case the overshoot becomes as large as 35 pixels; ii) the ISI index becomes smaller than the required threshold earlier in time (at about time unit 25 versus 55 in the non-compensated case). These measurements have been performed several times (N=6) and for different saccade amplitudes, namely 5, 10, 15 degrees eccentricity. Data have been averaged and standard deviation computed. Figure 22 summarizes the results. The overshoot for the compensated case is always smaller than the corresponding non-compensated case. At the same time, the stabilization interval required for the ISI to fall below the robust threshold in the compensated case is considerably smaller. In general, the data show that the use of inertial information in gaze redirection strategies leads to two important advantages: i) an earlier stable image of the new spotted location; ii) simpler motor control strategies for gaze-line redirection involving coordinated eye-head movements.

3.6 Reaching

We now have a reasonable understanding of the developmental progression in infant reaching – as outlined in section 1.5.2. Yet a mere description of a process of biological development does not shed any light to the issue of how physiological mechanisms of development are helpful for building complex artificial systems. A second step is necessary to bridge this gap. This step is to outline the control problems that have to be solved by human infants when trying to reach for objects in their immediate workspace. The first question to ask in the context of motor control is: what physiological or movement parameters does the system actually have to control to achieve its goal of reaching for a target in extra-personal space? To answer this question, consider that each limb segment of the human arm is moved by a set of actuators with spring-like properties. In the resulting mass-spring model, inertia, viscosity, gravity, etc have to be estimated in order to move the plant appropriately. Alternatively, the controller needs at least, a reasonable approximation of their values. That is, a first step could be the identification of the plant parameters. A second step, before goal-directed reaches are possible, is the mapping of sensory maps onto available motor maps. The system must be able to localize objects in extra-personal space, and should have knowledge of where its

limbs are positioned relative to the object – though this might not be always necessary. In a traditional learning paradigm, these two processes of calibration (i.e. plant identification and coordinates space transform) have to be completed before the system can begin to work on control. From an engineering perspective, this implies that calibration and control are separated. Consequently, many neural network models of arm control follow a learning paradigm, where the first step is the calibration of the system. In theory, some external teacher provides such plant knowledge, or the system calibrates itself by performing certain training movements. Subsequently, it learns to “control” the arm (Kalveram, 1991), (Kuperstein, 1988).

Box 10 The Asymmetric Tonic Neck Reflex. The stimulus for the ATNR is the turning motion of the head. This head turn triggers a complex bilateral synergy. The infant’s arm is extended to the side the infant is looking, effectively bringing the hand into the field of view. The contralateral arm is flexed as part of crossed extensor reflex spanning both homologous limbs. Thus, this multi-muscle synergy, coupling arm and head movements, provides an effective mean of linking visual and proprioceptive maps. The picture below shows the ATNR in two 3-month-old twins. Note the typical “fencer” position with one arm flexed and the other arm extended. The ATNR can be elicited up to the 4th postnatal month.



Such a separation of calibration and control is not observed in the development of biological systems. Here calibration and control are not two distinct and sequential phases of development, but are intertwined, proceeding in parallel, and build upon each other.

Today this view of a parallel development of calibration and control processes seems widely accepted by researchers working on neural modeling of adaptive eye-hand coordination. Yet, most researchers model this process as learning and not as a developmental operation (Jordan & Flash, 1994), (Kuperstein, 1988). Implicit to such an approach of artificial eye-hand coordination is the premise that all behaviors of the system have to be learned. However, this assumption is not necessarily true for biological systems. One major difference between a biological and traditional AI system is that a biological system does not come as a *tabula rasa*. In a wide variety of different species, one can observe stereotyped inborn movement sequences that are clearly unlearned. Ethologists have argued for a long time that many behaviors, especially those of lower animals, cannot be explained on the basis of sensori-motor learning alone (Eibl-Eibesfeld, 1970), (Gould, 1982). Newborn human infants already possess a repertoire of coordinated movements. For example, they can perform a series of complex multi-joint bilateral movements (e.g. kicking, grasping, etc) and have available a set of so-called primitive patterns that are triggered by a sensory stimulus. Yet these motor primitives may also serve a second function. They help to build up a relationship between vision and proprioception. For example, during pre-reaching the presence of the Asymmetric Tonic Neck Reflex (ATNR) plays a crucial role in allowing babies to see their hand and in increasing visual fixation of the hands (White, Castle, & Held, 1964), (Bushnell, 1981). A description of the ATNR is presented in Box 10.

However, not all patterns of early motor behavior are stimulus-bound. The orienting and pre-reaching behaviors of infants already contain a goal-directed component. The frequency and extent of these movements depends on the emotional and attentional state of the infant and not exclusively on the presence of an interesting visual object. For Trevarthen (Trevarthen, 1984) these early behaviors indicate that infants are born with a certain knowledge of or readiness for object exploration.

Given these premises we proceed discussing how they fit into the artificial development of the reaching behavior.

3.7 Reaching: the idea and beyond

The approach we shall follow here is directly based on motor primitives, representing multi-joint synergies (e.g. arm extension). In this case, a single command may produce complex multi-joint coordinated movements without the voluntary control of each individual degree of freedom (DOF). One example of such multi-joint synergy is the ATNR presented above. Another example is the grasping reflex, which activates a coordinated grasping movement of the hand when the palm touches an object. In order for this approach to be feasible and effective, the crucial points are how to represent the motor primitives, their developmental rules, and the mechanisms of sensori-motor mapping.

As far as the coding of motor primitives is concerned, one possible procedure is the so-called force fields approach originally proposed by Mussa-Ivaldi and Bizzi (Mussa-Ivaldi & Giszter, 1992), (Mussa-Ivaldi et al., 1993). It is not our intention to present the whole mathematical details here – the model is described extensively in section 5.6. According to the force field theory, the action of reaching a point in space can be described by a force vector field converging to an Equilibrium Point (EP). The EP can be thought of as the point toward which the end-point of the limb is moving at each instant of time, and a limb trajectory can be represented by a sequence of EPs. The trajectory in space of the EP does not correspond to the actual trajectory of the arm and is, therefore, called “virtual trajectory”. Of course, there is nothing magic about the origin of this “force field” formalism. In practice, each force field is simply the result of the action of muscles and the EP is simply the intersection of muscles’ torque-length characteristics (see Box 11).

The mechanism is well suited to implement the kind of motor reflexes present at birth. Each reflex can be represented by a force field; this is in turn obtained by activating a synergy of simulated muscles. Of course, the robot’s actuators are not muscles, but they allow being torque controlled – i.e. by programming the current that flows into the motors, it is possible to simulate whatever characteristic in software, although with some limitations. The latter is of fundamental importance; the fact that the actuator’s response can be controlled means that we can, at least in theory, choose any impedance characteristic of the arm end-point. This means also that we might decide to control the end-point stiffness, adapting it to the characteristic of the task. We did not investigate further this aspect of the “field-based” controller; we just programmed the robot to use a constant low stiffness value. As a consequence, the robot is partially compliant, i.e. it can interact safely with the environment (including humans).

The situation, however, becomes more complicated when goal directed movements – such as reaching a point in space – are considered. In this case, the trajectory has to be controlled or initiated on the basis of sensory information. If this information were provided as a 3D position of the target in space, the kinematics of the eye-head system as well as of the arm would have to be explicitly considered in order to select and combine the appropriate force fields.

The solution we propose here is based on the use of a direct mapping between the eye-head motor plant and the arm motor plant. One premise we make is that the position of the fixation point coincides – at least at some stage of the control process – with the object to be reached. In other words, reaching for an object starts by looking at it. Under this assumption, the fixation point can be seen as the “end-effector” of the eye-head system. The positions of the head with respect to the torso, and that of the eyes with respect to the head, uniquely determine its position in space relative to the shoulder. Consequently, at least in principle, the arm force fields can be obtained by a transformation of these plant variables. We will call this approach “motor-motor coordination”, because the coordinated action is obtained by mapping motor plant variables into motor plant variables.

Given these considerations, we start here a simple experiment, where the robot is constrained to move in a plane – i.e. only two joints are controlled. In this case the mapping can be expressed by:

$$\mathbf{C} = f(\mathbf{q}) \tag{22}$$

where f is the unknown true function which must be approximated by learning, \mathbf{q} is the head joint angle’s vector (limited to 2 joints in this case – i.e. neck panning and the common tilt) and \mathbf{C} is the arm activation vector, which has dimension equal to four – this is arbitrary, and it was predetermined by the experimenter. Two muscles in a push-pull configuration control each joint, and consequently four motor primitives allow all possible synergies of the (*2 muscles* × *2 joints*) four muscles. The controller, in this case, is shown in Figure 23. The (\mathbf{q} , \mathbf{C}) pairs required to estimate the function f are measured whenever the system is fixating its own hand (and not when the gaze is fixating the target). The values of the activation vector \mathbf{C} are stored in a growing look-up table (the motor-motor coordination map), whose input space \mathbf{q} is sampled with variable resolution up to a maximum predefined by the experimenter. Each unit

can store a corresponding output vector \mathbf{C} . The vector stored in the closest cell to the query point gives the output of the map.

If the table is queried with a point \mathbf{q} and no unit has been previously allocated in an appropriate neighborhood (which implicitly defines the maximum resolution), a new unit is instantiated at the position \mathbf{q} . Values lying inside the activation region (nearest neighbor) of each unit are inserted using an averaging procedure.

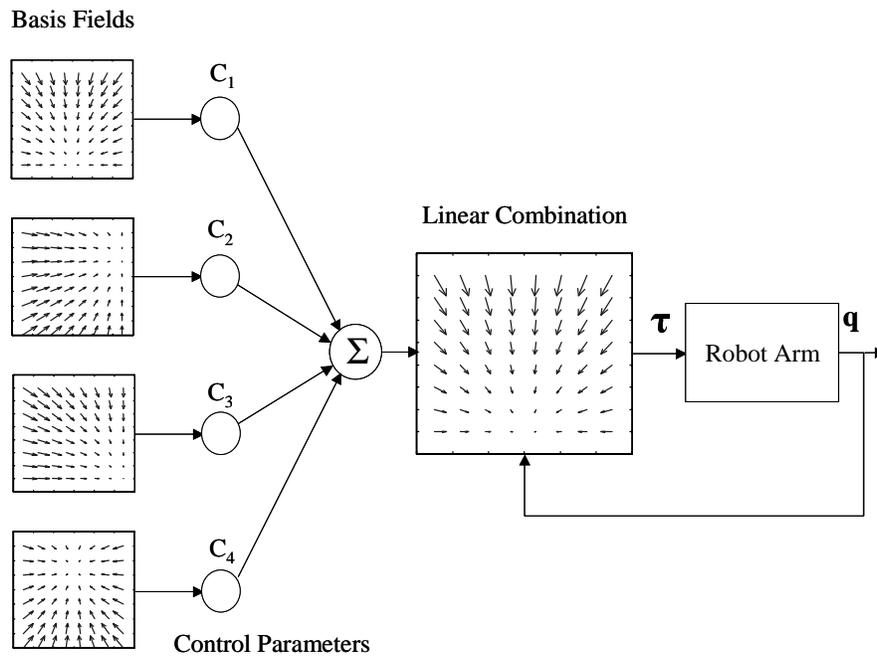
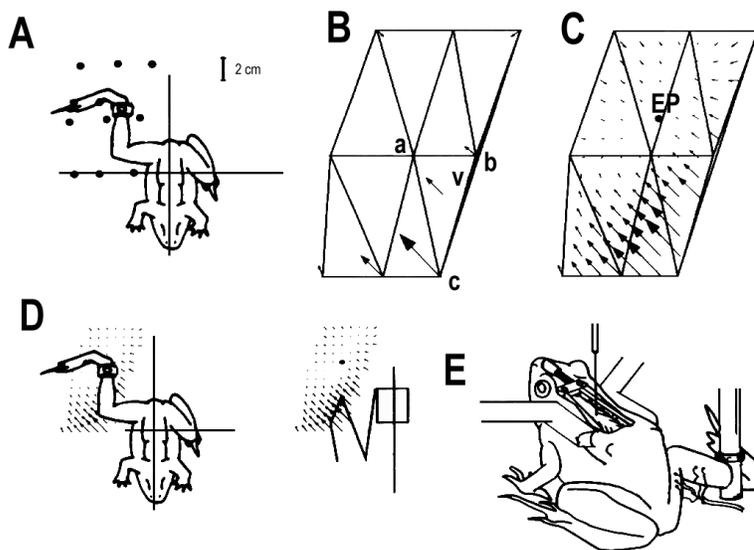


Figure 23 Controller structure: motor primitives, represented by torque fields are combined (weighted by C_1 , C_2 , C_3 , and C_4). The overall field “guides” the arm end-point toward the EP.

Box 11 Bizzi and colleagues' experiment. By electrically stimulating a few sites in the frog's spinal cord, Bizzi and coworkers showed that the limb movement could be coded in terms of "force fields" and an Equilibrium Point (EP). The experiment consisted in exposing the spinal cord (as shown in figure E below) in order to implant an electrode; the frog's limb was connected to a force sensor. The limb was moved passively to different positions within its workspace and the forces with and without stimulation were recorded (see panel A and B). The position of the electrode was kept constant for an entire set of measurement. Force samples were interpolated, as shown in panel C, in order to get a continuous approximation of the underlying force field (the EP was identified as the point where the field is equal to zero). It was surprising that among all possible fields, many of the measured ones were of convergent type with a single EP. By stimulating different sites, the experimenters identified about four different fields. It has been shown also that the vast majority of limb movements can be coded by just combining these fields linearly. This is even more surprising given the non-linearity of actuators and neural responses. This led to the hypothesis that limb placement can be obtained by moving the EP, this in turn might be obtained by a linear combination of the four identified fields (which are termed "basis fields"). Panels D and E show the position of a force field with respect to the limb workspace. Adapted from (Mussa-Ivaldi et al., 1993).



3.7.1 Initialization of the motor-motor map

The first problem to be solved is how to initialize the map in a meaningful way (or in other words, what type of motor primitives should be used as the basis of the learning procedure). In natural systems, this is obtained by reflexive mechanisms like the ATNR, which might have the role of maintaining the arm within the field of view. In our experiment, the robot utilizes a discrete approximation of the ATNR by initializing the head-arm map so that the arm is extended roughly in the direction that the head is turned. The map stores three initial values for each of the four elements of the activation vector \mathbf{C} corresponding to three head positions. Each component of the map is virtually empty, apart from the three “dots” representing the values \mathbf{C} corresponding to three head positions. The three activation vectors uniformly span the arm workspace and were computed, so that whenever they are used the arm end-point would move into the camera field of view. Consequently, even if the choice of just three positions is arbitrary, this initialization of the head-arm mapping is advantageous with respect to a random sampling of the workspace for two reasons. First, the system is put in the conditions to be able to learn from visually measured errors (the arm is kept in the field of view); second, the initial values implicitly limit the exploration space to accessible and safe regions of the workspace. It is worth noting that initially the head can explore its entire workspace while only three positions of the arm are possible. The goal of the learning procedure is to fill the empty space of the maps.

3.7.2 Trajectory generation¹²

The extracted activation vectors \mathbf{C} cannot be applied instantaneously because the arm has high inertia and the friction of the reduction gears. The application of a step command would bring the torque outside the operational range of the motors. To avoid this situation, a mechanism transforming the activation values obtained by the map into smooth sequences is required. Such gradual rise in force is also observed in biological motion (Kandel et al., 1991). A possible biological mechanism for incremental rise in force levels is motor unit size, with smaller units discharging first during the contraction (Hennemann & Mendell, 1981). To achieve a smooth rise in torque, we applied a linear interpolation for a fixed number of steps between the initial and the final activation values:

¹² Please note that a complete description of the arm control model is contained in section 5.6.

$$\mathbf{C}_{t+1} = \mathbf{C}_t + \frac{\mathbf{C}_{final} - \mathbf{C}_{initial}}{n_{steps}} \quad (23)$$

where \mathbf{C}_t is the activation vector at the t^{th} time step, \mathbf{C}_{final} the target activation vector, $\mathbf{C}_{initial}$ its value when the command was issued, and n_{steps} the number of steps. At each time instant t , it is possible to determine an EP, which is a function of \mathbf{C}_t by imposing:

$$\mathbf{T} = \sum_j \mathbf{T}_j = \sum_j C_j \sum_i \mathbf{I}_{ji} \boldsymbol{\tau}_i = \mathbf{0} \quad (24)$$

The sequence of EPs defines the arm's virtual trajectory. However, the sequence of \mathbf{C} through time also determines the shape of the trajectory. \mathbf{C}_t can be considered as a set of parameters, which are learnable in principle. In fact, they could be tuned in order to straighten the trajectories or to reduce overshoots. Consider the usual Lagrange equation for a planar manipulator:

$$\mathbf{T} = A(\mathbf{q})\ddot{\mathbf{q}} + B(\mathbf{q}, \dot{\mathbf{q}}) \quad (25)$$

where \mathbf{T} is the generalized torque applied to the arm. Substituting the expression for \mathbf{T} , generated by the set of elastic actuators and controllers, as previously defined, yields:

$$\sum_j C_j \sum_i \mathbf{I}_{ji} \boldsymbol{\tau}_i = A(\mathbf{q})\ddot{\mathbf{q}} + B(\mathbf{q}, \dot{\mathbf{q}}) \quad (26)$$

Two considerations stem from the previous equation: i) the real trajectory of the arm is determined by the shape and evolution in time of the torque field (left hand side of equation); ii) as already pointed out, the shape of the torque field is controlled by \mathbf{C}_t . If the system were able to tune \mathbf{C}_t , aside from the simple linear interpolation, it could also modulate the resulting arm trajectory precisely. Although this may be a sensible strategy (for example to learn how to get a straight trajectory instead of a curved one), it was not investigated further in this experiment.

The overall control scheme is shown in Figure 24. The first stage of the processing is implemented in the map containing the arm activation vector. These values are interpolated and the output from the trajectory

generator is sent to the actuators simulator (identified by the block “Field generation”), which eventually generates the torque commands.

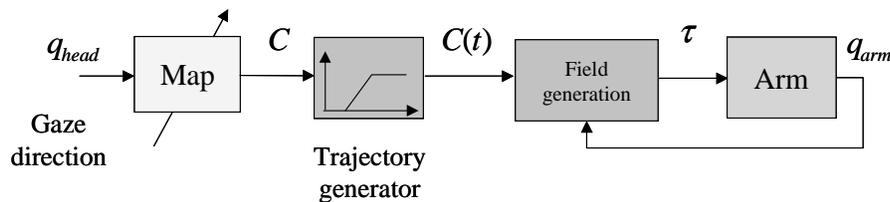


Figure 24 The overall arm’s control scheme. The position of the head (q_{head}) queries the map that “computes” the activation vector for the arm. This stage is followed by a trajectory generation that interpolates linearly between activation vectors. The resulting force field is then computed and used to generate the torques, which drive the arm motors.

3.7.3 The learning procedure

The learning algorithm can be formally described as follows:

Repeat forever.

1. A proper stimulus appears in the field of view.
2. The head moves in order to fixate the spotted stimulus.
3. By fixating the visual target the robot also initiates arm motion by computing the arm activation vector \mathbf{C} in the following way:

$$\hat{f}_i(\mathbf{q}) + \mathbf{n}$$

The term \mathbf{n} describes a zero-mean uniform noise component introduced to simulate errors in the arm control. \hat{f}_i is the estimate of f at the i^{th} iteration.

4. The arm controller uses the vector \mathbf{C} to compute the actual torques to drive the motors. Consequently the arm moves toward the new EP.
5. At this point the arm is as close as possible to the target (initially it is not very close but certainly it is in the field of view), so that the system can re-direct the gaze to its own hand.
6. As a result of the previous step, a new pair (\mathbf{q}, \mathbf{C}) is available which is used to update the map by computing the value $\hat{f}_{i+1}(\mathbf{q})$ in the following way:

$$\hat{f}_{i+1}(\mathbf{q}) = \hat{f}_i(\mathbf{q}) \frac{n_v - 1}{n_v} + \frac{\mathbf{C}}{n_v}$$

where n_v is the number of visits of the cell corresponding to \mathbf{q} . If a cell close enough to \mathbf{q} does not exist a new unit is added to the mapping.

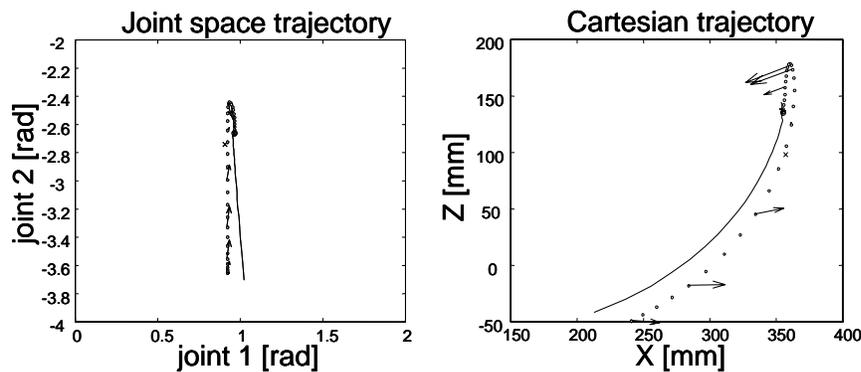
7. The arm then returns to a fixed resting position near the chest.

It is important to note that if the procedure were noise free, the motion of the arm toward the target (end of step 3) would always bring the end-effector in the same final position and the system would not be able to learn (in fact, it would always update the same cell of the map with the same vector \mathbf{C}). The motor-motor mapping, at least initially, does not necessarily bring the end-effector near to the fixation point (it will bring the arm as close as possible to the target on the basis of what has been learned so far). However, instead of correcting the error by moving the arm, the direction of gaze is redirected to the end-effector and the arm motor command previously issued is associated to the new eye position. In other words, the role of the visual target appearing in the environment has the only function of initiating the arm motion, while the learning process is based on the act of looking at the end-effector. As the learning process proceeds, the initial arm motion gets closer and closer to the visual target, and eventually, the corrective gaze shift will not be necessary unless kinematic changes occur.

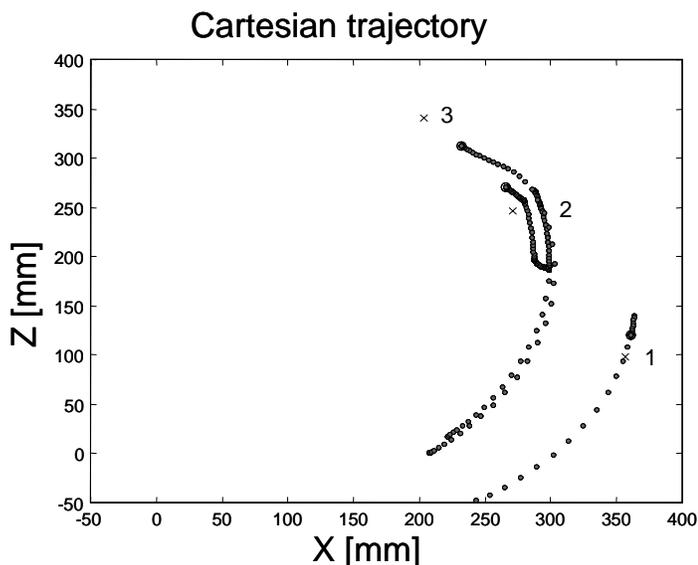
Box 12 Reaching trajectories. This experiment was performed to illustrate the performance of the proposed approach. It describes the learning of ballistic reaching movements toward static visual targets. In order to test the performance of the system at different learning stages, the position in the arm's workspace of three targets was calibrated beforehand by manually positioning the end-effector at target center and storing the corresponding joint angle values measured by the encoders. Each target consisted of a piece of cardboard about $5 \times 5 \text{ cm}$ in size. During the training the target of the reaching task was manually moved by the experimenter over the arm's workspace while the reaching behavior was continuously activated. From time to time training was suspended and performance evaluated. During the evaluation phase, the three targets in the calibrated positions were activated one at a time and the trajectory of the arm stored. The reaching error was measured by computing the Euclidean distance between the pre-calibrated target positions and the position of the end-effector at the end of the reaching movement. At least 30 trials (10 for each target) were executed and the average error and standard deviation were computed. During this evaluation phase the map update was stopped and the noise term removed. The reaching error before and after 51 trials, and after 134 trials are reported in the following table:

Number of trials	Before	51	134
Error (<i>mm</i>)	77.8 ± 15.0	39.5 ± 12.0	28.8 ± 8.9

It is important to note that trajectories are not learned by the system. They are just a consequence of the applied control strategy as described in section 3.7. A typical arm trajectory after training in joint and Cartesian coordinates is shown in the figure below.



In both graphs, the presence of overshooting of the real trajectories is observed. This is the effect of motion dependent forces, which are unknown to the controller. As a consequence the torque applied in the initial part of the movement brings the end-effector beyond the target. The “force field” approach, however, corrects this overshoot by applying a force in the opposite direction and partially compensates this lack of dynamic information. In our current schema there is no chance to “learn” how to avoid this overshoot because this would require tuning other parameters such as the stiffness, or the presence of compensating modules, which explicitly take dynamics into account (Ghez, Gordon, Ghilardi, & Sainburg, 1996). By observing the plots below similar considerations can be drawn. In this case reaching movements toward three different targets at the end of the training phase are shown. The trajectory toward target 1 shows the same overshoot described before. The opposite happens when the most distant target 3 is reached, where the end-effector undershoots the target. The remaining error can be attributed, in part, to intrinsic errors of the learning process, but also to the accumulation of errors deriving from friction. The trajectory toward target 2 shows a back-and-forth motion with the final position reached after a couple of adjustments. This behavior is caused by the fact that the system is continuously operating and, consequently, whenever the end-effector partially covers the target, the head shifts the fixation point over the center of gravity of the remaining visible part. This change of fixation generates a new “force field” and consequently, a new trajectory. Eventually, the visible part of the target does not change and the arm reaches its final position.



3.7.4 Biologically plausible trajectory profiles

One of the most striking characteristics of human arm trajectories is that they show a remarkably stereotyped profile, both in terms of position and velocity (Abend, Bizzi, & Morasso, 1982). It is still not clear what criterion is employed by the CNS to generate such repetitive patterns. Many authors, for example, proposed an optimization criterion based on the minimization of either jerk or torque, with different flavors in order to take into account timing and smoothness of the trajectories (Jordan, 1996). This is not to say that the CNS is explicitly minimizing a sort of cost function, but nonetheless the mathematical formulation can help understanding which principle the CNS may use. Although we did not address the problem of trajectory generation, apart from the simple interpolation schema proposed in section 3.7.2, trajectories were measured for analysis purposes.

In spite of the simple linear interpolation, we observed a consistent bell-shaped profile – an example is shown in Figure 25. On the other hand, the trajectory itself is not straight, and shows remarkable overshoots, perhaps because of the lack of dynamic compensation. This is to say, as pointed out by some authors (e.g. (Gomi & Kawato, 1997)), that the CNS might need to take into account dynamics when moving a limb, because under such low-stiffness control self-generated forces might contribute substantially to the total torque at joint level. In our model, dynamics is not explicitly considered, thus it is not surprising that trajectories are neither straight nor precise on the target. In any case, the “force field” approach allows considering even more basis fields, and we may imagine that some of them are built by a learning procedure. In other words, the robot can acquire more basis fields, which compensate for the self-generated forces, or other external disturbances – such as gravity.

Therefore, a further stage of development can explicitly use such modules and exploit external and motion dependent forces to draw trajectories in extrinsic space that are even more efficient.

3.8 Improving reaching by employing more DOF

Beside the simple 2D case presented above, the goal of the robot is actually to learn positioning in 3D. The 2D experiment was illustrative, and showed all the concepts employed for the full 3D experiment. On the other hand, some changes were necessary in order to speed up learning. The difference between the 2D and 3D experiment is described below. In any case, it does not change our concept about reaching: that is, a

representation of gaze in motor coordinates is mapped into a representation of the arm motor commands.

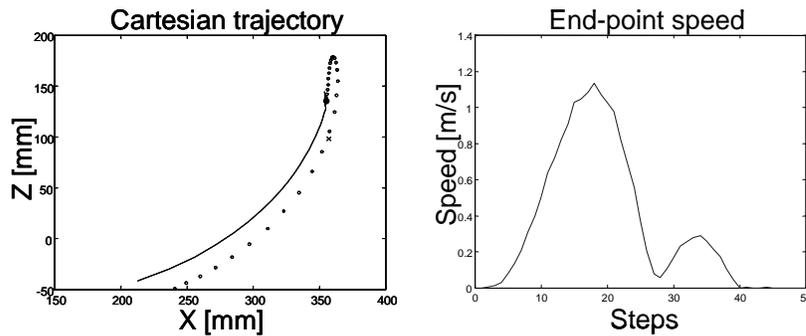


Figure 25 The trajectory showed a bell-shaped velocity profile. On the left, the trajectory of the arm end-point: abscissa and ordinate represent the plane where the arm motion was constrained – bear in mind it was a 2D experiment. The dashed line is the actual trajectory sampled at 40Hz, the solid line is the “virtual trajectory”. In spite of the virtual trajectory that moves directly to the target, the actual motion showed an overshoot. On the right, the hand speed has a bell-shaped profile. Note the two bumps corresponding to the first large “transport phase” and a second corrective movement. Time is expressed in control cycles (25ms), and speed in meters per second.

3.8.1 Reducing input size

If we were following the approach described above, we might proceed by using the vector of head joints as input to the learning module – the lookup table. The head joints vector implicitly represents the gaze direction. The position of the fixation point in space can be coded by only three variables, while the joint vector has five. Are all these inputs actually necessary? The answer is no. In fact, by analyzing the controller behavior, we may notice that the head has two control-imposed constraints. First, vergence has to be symmetric – this is not to say that the condition is always respected in the transient, though for the steady state this is exactly the observed situation. Second, the overall tilt controller links the two available “tilt DOFs” into one. As before, the configuration the head is aiming at, has the eyes frontal (in terms of tilt) to the target. By simply analyzing the configuration of the head while

fixating, we may discover that gaze can be represented by only three variables. We carried out a Principal Component Analysis on the final head configurations, and found that 93% of the final head positions are described by only three components.

For this reason, we do not need to code the precise position of the fixation point. That is, it is safe to assume gaze can be coded in terms of angles that specify its direction, which is what is important for the reaching task.

A suitable coding of the gaze direction is in terms of the vergence, version, and tilt angles. Without resorting to any kinematics, they can be expressed as:

$$\begin{aligned}\theta_{vergence} &\approx (q_4 - q_5) \\ \theta_{version} &\approx \frac{1}{2}(q_4 + q_5) + q_1 \\ \theta_{tilt} &\approx q_2 + q_3\end{aligned}\tag{27}$$

3.8.2 Reducing output size

Looking at the controller structure, we may notice that also the output – i.e. the network target function – can be simplified. In the 2D example, the network outputs were the activation values. As shown in section 5.6.1, given an EP, we can easily compute the activation vector. The idea here is to employ the position of the EP as network output, and in a later stage, convert the EP position into the activation values. The “exploration space” size is reduced and, at the same time, the network output size is limited to the number of controlled joints, which is independent of the actual number of basis fields (which are six in the 3D experiment)– there are three controlled joints in the 3D experiment. Another advantage is that position and stiffness can be controlled separately. Though we always used a constant stiffness matrix, it might be a sensible strategy to vary the end-point stiffness during motion. For instance, when the robot interacts with unknown objects, the stiffness might be reduced in order not to damage either the robot itself or the external object. Kawato and coworkers (Gomi & Kawato, 1997), measured human arm stiffness during multi-joint movements and discovered that postural stiffness is generally higher than movement related stiffness.

Consequently, the arm control schema for the 3D case has been modified as depicted in Figure 26. The learning sequence is not changed if

compared to the 2D experiment; in particular, the map is initialized as shown in section 3.7.1, and implemented as a growing lookup table.

In the 3D case, we recorded the trajectory of the arm end-point. In addition we recorded the gaze direction, from which it is possible to recover the position of the fixation point in space. The two quantities have been plotted during different reaching movements in order to illustrate quantitatively the behavior of the robot. Note that, the position of the fixation point provides an estimate of the target position; in fact, whenever the retinal error is below a small threshold, we can assume that the position of the fixation point is indeed the position of the target. Box 13 shows the reaching map plotted as a set of 2D maps – see caption. Box 14 and Box 16 show two 100 samples long reaching trials extracted from about half an hour of continuous operation of the robot.

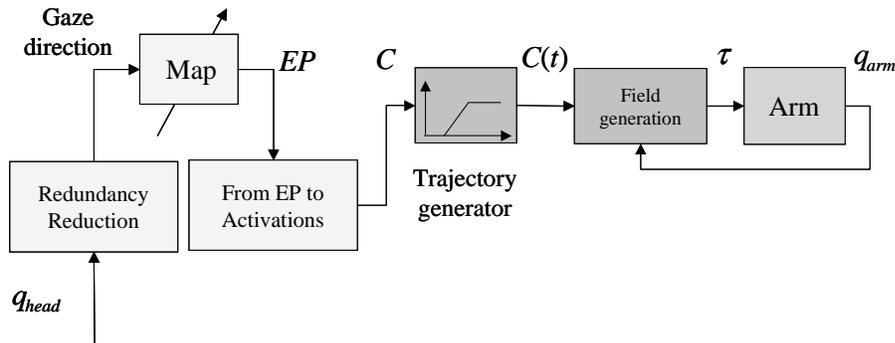
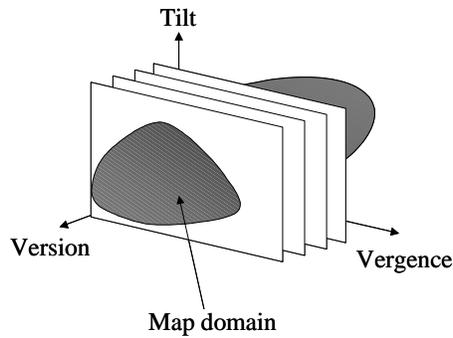
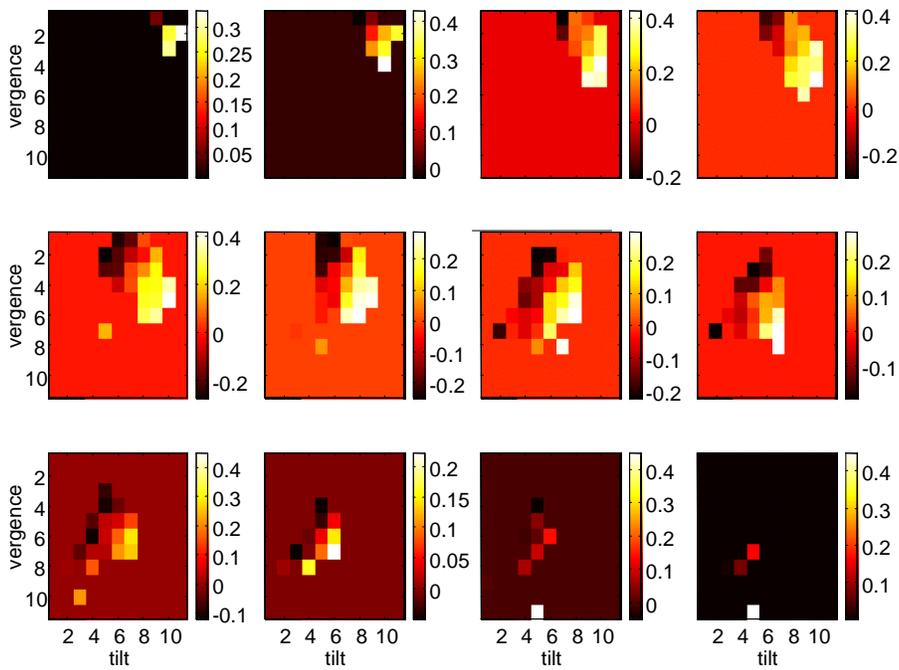
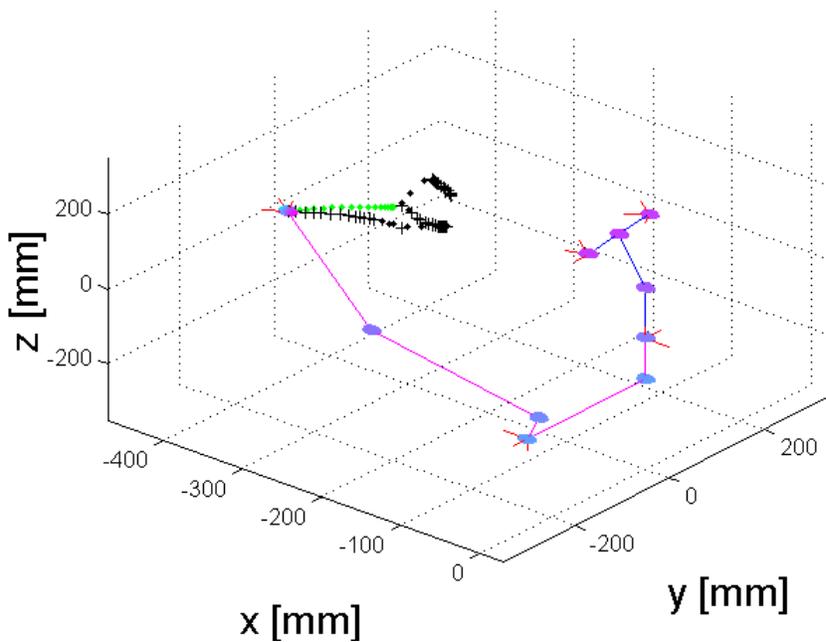


Figure 26 The arm control schema: 3D reaching experiment. This schema roughly resembles Figure 24 apart from the two “light gray” blocks which carry on the “redundancy reduction” and the computation of the arm activation values starting from the equilibrium point position.

Box 13 Reaching lookup table. The figure below shows one component of the reaching lookup table after more than 1000 reaching trials. In order to display it (see sketch at the bottom of this box), the following procedure has been applied: i) the input domain has been divided into a regular grid (for a total of $11 \times 11 \times 12$ cubes); ii) 12 slices of varying “version angle” are plotted as 11×11 2D maps; iii) the color intensity represents the output of the map controlling joint 6 (shoulder); iv) the outputs of all units falling into the same cube have been averaged. The output is the position of the EP in joint space, which is expressed in radians.



Box 14 Reaching trajectories. In the 3D case we recorded the trajectory of the arm end-point. In addition, we recorded the gaze direction, from which it is possible to recover the position of the fixation point in space. The two quantities have been plotted during a reaching movement in order to illustrate quantitatively the behavior of the robot. Moreover, the position of the fixation point provides also an estimate of the target position. In fact, whenever the retinal error is below a small threshold, we can assume that the position of the fixation point is indeed the position of the target. The simple wire-frame model represents the robot. Small circles indicate joints; solid lines are the links. Concerning the fixation point, two different marks can be distinguished: the cross marks represent the time instants when tracking was of smooth pursuit type; whereas, the small squares are instead related to saccadic control.



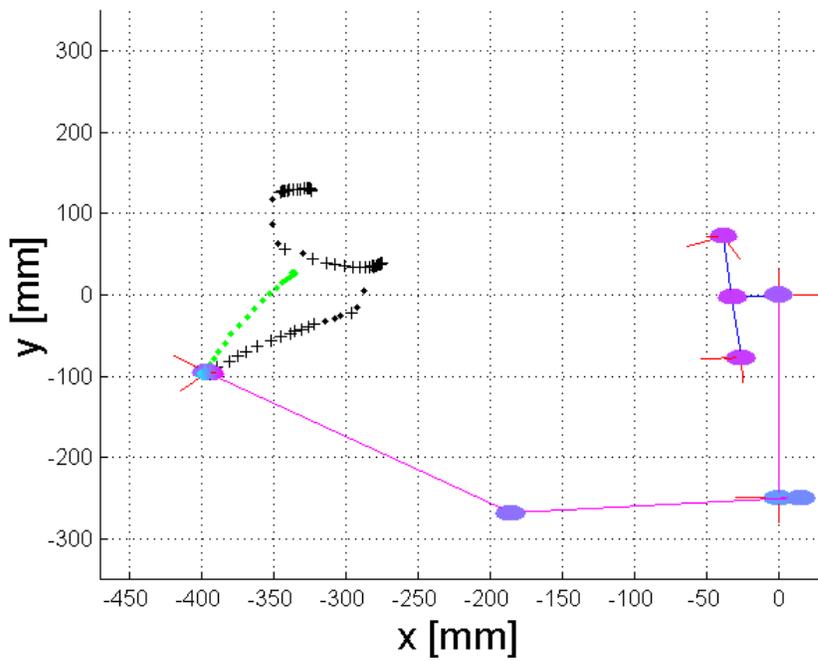
3.9 Discussion

This chapter presented in detail, the robot adaptive control structure and a series of experimental results in support of the theoretical claims we made earlier. In the first part, we showed how the robot could acquire orienting behaviors by carefully exploiting some of the “developmental principles” we outlined in the previous chapters. For example, the control structure “at birth” consists of only a simple closed loop controller, whose explorative behavior is driven by noise. A more sophisticated model based saccade generator develops on top of the former. Furthermore, initially, the robot only moves a few joints; proper coordination of the redundant degrees of freedom is built only when these initially working joints are under a sort of “voluntary” control.

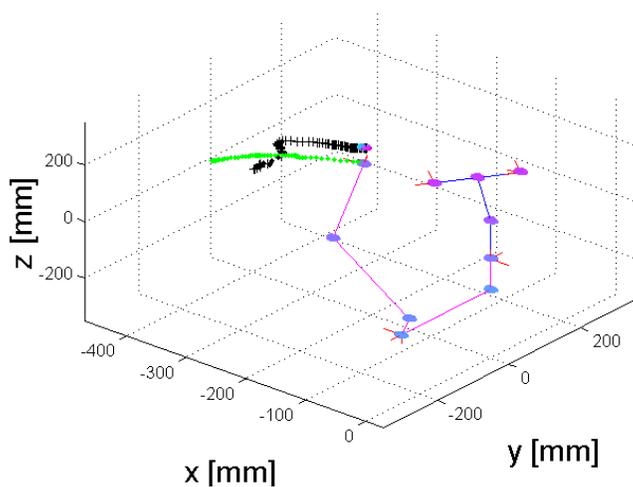
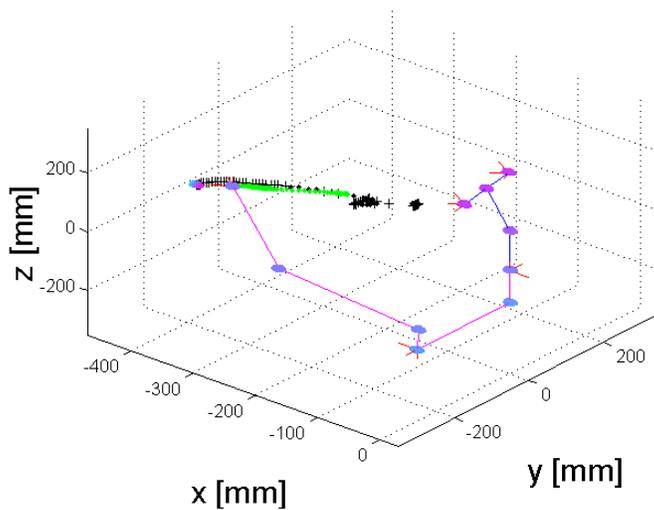
In parallel, we initialized the head-arm coordinative action by using only simple “handcrafted” spinal reflexes. These initial controllers become eventually part of the coordinative action: i.e. reaching.

The biologically inspired design was exploited at many levels, including (but not only) the low stiffness control of the arm motion, the use of the inertial information in the control loop (VOR), and the space variant resolution of the cameras.

Box 15 Reaching trajectories: top view. As in Box 14, two trajectories are shown. Two different marks can be distinguished: the cross marks represent the time instants when tracking was of smooth pursuit type; whereas, the small squares are related to saccadic control.



Box 16 Reaching trajectories: two representative trajectories of the robot behavior. As before, two different marks can be distinguished: the crosses mark smooth pursuit control; whereas, the small squares are related to saccadic control. Note that in the topmost picture the end-point and fixation position almost coincide.



4 Conclusions

We are at the end. The previous chapters presented a proposal for a novel approach aimed at the design and comprehension of complex systems. This approach arose by observing how biological systems solve the problem of learning and adaptation during the early stages of their lives. We tried to isolate those aspects, which may be relevant both for the construction of artificial systems and for advancing our understanding of the corresponding brain functions. An important point worth stressing is that the brain cannot be seen as a monolithic structure, but rather we need to look at it as a developing system, where many subparts optimally interact. This internal organization might indeed facilitate learning and in this sense it is worth copying when one goes through the design of an “artificial adaptive agent”. Other aspects have been discussed, for instance, the presence of “innate” behaviors, which later disappear. This is an open question: do they really disappear? It might very well be that those initial modules get embedded into more complex control structures. In this sense, voluntary control can be seen as learning to combine the initial reflexes, in order to solve a particular task.

We are conscious that we did not provide any formal justification, but at least we provided hints on what aspects might be relevant. These were partially introduced in chapter 1, and detailed further in chapter 2, where the “learning problem” was described on the light of biological findings.

Finally, by using a “learning by doing” philosophy, we built a humanoid robot, and “programmed” it following some of the biological aspects we denoted as “relevant” for artificial development. The robot indeed faced problems, such as moving many degrees of freedom by employing many different cooperating controllers. This is exactly the point, how should we connect all these modules together? Consider that they are not separated because all of them act on the same non-linear physical plant. Consequently, interactions must be explicitly taken into account. We devised a solution, where the timing of adaptation is carefully (but not too much) programmed. That is, the solution works by creating a proper time slot for each subpart (slots do not need to be temporally separated one from another). Inside this “critical period”, adaptation can effectively take place without disturbing the other modules excessively. This is important, especially in the early phases, when plasticity must be high (i.e.

exploration) in order to quickly acquire a consistent behavior. Yet another type of interaction occurs: modules that develop first influence modules that develop later. Consequently, the “explored state space” depends very much on how these early controllers behave. Each module can function as a “bootstrap” procedure for other subsystems. This is exactly “constructive learning” on a coarse scale, where entire streams, areas, controllers can be considered as “basis modules”. Constructive learning is thought to be superior to other learning techniques (pruning based), as mentioned in the previous chapters.

So, the spotlight moved from learning itself to the process of learning: i.e. development. What and how could be learned is determined by the learner’s developmental stage, that is, by what the state of the whole system is in terms of the other subparts (e.g. the robot could not move the neck without controlling the eyes first).

Of course, most of these conjectures need to be verified, from either the theoretical side (e.g. learning theory) or the biological point of view (e.g. by designing new experiments, for example to determine how gazing correlates with reaching). In this light, the most sensible prosecution of this work would be that of investigating all these open questions, in order to formalize a theory about developing systems. On the other hand, many hypotheses arose, which might be worth testing on “real brains”.

5 Appendix

5.1 Log-polar images

Studies on primate's visual pathways from the retina to the visual cortex have shown that the geometrical layout follows an almost regular topographic arrangement (Daniel & Whitteridge, 1961), (Cowey, 1964), (Allman & Kaas, 1971), (Hubel & Wiesel, 1977). The initial analytical formulation based on this data is due mainly to Schwartz (Schwartz, 1977); his model can be roughly summarized as follows:

- The distribution of the photoreceptors in the retina is not uniform. They lay more densely in the central region called fovea, while they are more sparse in the periphery. Consequently, the resolution also decreases moving away from the fovea toward the periphery. It has a radial symmetry, which can be approximated by a polar distribution.
- The projection of the photoreceptors array into the primary visual cortex can be well described by a logarithmic-polar (log-polar) distribution mapped onto a rectangular-like surface (the cortex).

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:

$$\begin{cases} \eta = q \cdot \theta \\ \xi = K_{\xi} \ln_a \frac{\rho}{\rho_0} \end{cases} \quad (28)$$

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 characteristics). These are related to the conventional Cartesian reference system by:

$$\begin{cases} x = \rho \cos \theta \\ y = \rho \sin \theta \end{cases} \quad (29)$$

A graphical example is shown in Figure 27, where the upper row (a) shows a log-polar or cortical image on the left, and a remapped Cartesian (or retinal) image on the right. The lower panel (b) shows how a simplified retinal layout maps to a log-polar mesh. For instance, the innermost circle (fovea) maps to the first column in the log-polar layout, and radii in the retinal image map to rows in the cortical one. The “flower” picture shows this polar mapping effect more clearly – see Figure 28.

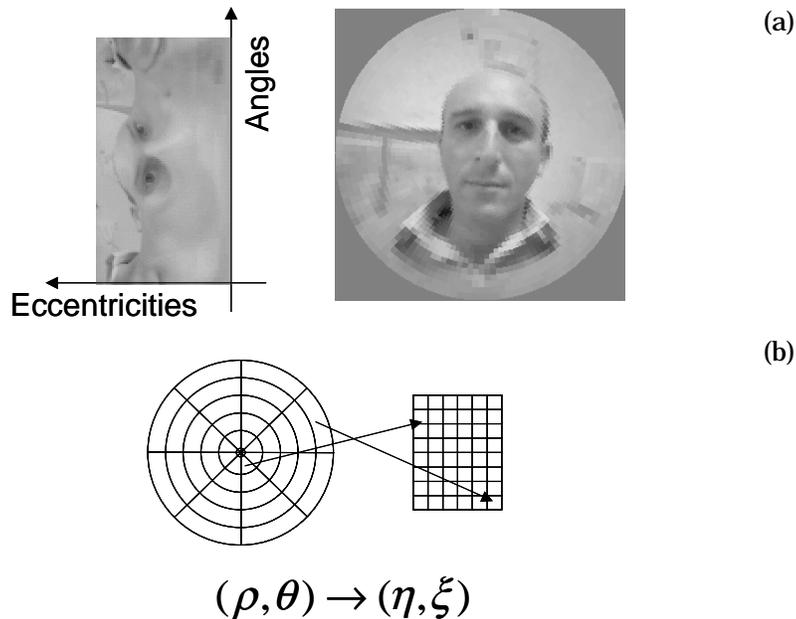


Figure 27 The log-polar mapping. The upper row (a) shows a log-polar or cortical image as acquired through the Giotto Camera (Sandini et al., 1998) a CMOS realization of the log-polar mapping at the sensor level (this image is 76×128 pixels). The image on the right is the corresponding remapped retinal image. The lower panel (b) shows – albeit simplified – how retinal pixels map onto cortical pixels. For instance, the innermost circle (on the left) map to the leftmost column (on the right) representing the fovea, on the contrary radii map to rows in the cortical image.



Original image



Log-polar image

Figure 28 An example of log-polar mapping, note as radial structures in the flower (petals) map to horizontal structures in the log-polar image. Circles, on the other hand, map to vertical patterns. Furthermore, note as the central part of the flower occupies about half of the corresponding log-polar image.

5.2 Optical flow

Optic flow, by definition, is the apparent motion of luminance patterns in the images (retinas). Under not too restrictive assumptions it can be assimilated to the motion of physical objects in the environment or to the self-movement of the cameras (eyes) – see (Horn & Shunck, 1981). The determination of the optical flow is an ill posed problem; consequently, either regularization techniques or integral methods have been proposed in the related literature (Barron, Fleet, & Beauchemin, 1994). In general terms by assuming brightness constancy on moving pixels, it is possible to derive the well-known Horn's equation (Horn, 1986):

$$\frac{dE}{dt} = 0 \Rightarrow \frac{\partial E}{\partial x} \dot{x} + \frac{\partial E}{\partial y} \dot{y} + \frac{\partial E}{\partial t} = 0 \quad (30)$$

where E is the image intensity, (\dot{x}, \dot{y}) the flow field.

It is evident that from equation (30) alone, the flow field cannot be uniquely determined for each pixel on the image – i.e. only the component along the gradient of the image intensity can be computed (Horn, 1986). Again, generally speaking, it is necessary to employ either a further constraint or a suitable *a priori* model about the resulting flow field. Depending on the application either technique will be suitable. Our choice was to employ an affine model, which easily allows recovering the first order flow field differential invariants, although it might be inaccurate if a planar condition is not met. The first formulation of this algorithm is due to Koenderink (Koenderink & Van Doorn, 1991), and it is based on the following equation:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} + \begin{bmatrix} D + S_1 & S_2 - R \\ R + S_2 & D - S_1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \quad (31)$$

which depends on four quantities: translation, rotation, divergence and shear. The first two components u_0 and v_0 represent a rigid two-dimensional translation, and D , R , S_1 , S_2 are the four first-order vector field differential invariants: divergence, curl, and shear (S_1 , S_2).

By combining equation (30) and (31), and solving for the six unknown parameters, the global first-order approximation of the flow field can be uniquely determined using a least square technique. Note that the usual hypotheses about the conditioning of the resulting over-constrained system must be satisfied in order to get a solution. In order to solve the system of equations, at least six points are necessary.

Moreover, considering that the optic flow we are looking for has to be computed in the log-polar¹³ (cortical) plane rather than in the traditional Cartesian image (retinal), Horn's equation becomes:

$$-\frac{\partial E}{\partial t} = \begin{bmatrix} \gamma_1 & \gamma_2 & g_\xi & g_\eta & \gamma_3 & \gamma_4 \end{bmatrix} \cdot \begin{bmatrix} u_0 & v_0 & D & R & S_1 & S_2 \end{bmatrix} \quad (32)$$

where $g_\xi = \frac{\rho_0 \cdot K_\xi}{\ln a} \frac{\partial E}{\partial \xi}$, $g_\eta = q\rho_0 \frac{\partial E}{\partial \eta}$, and

¹³ See section 5.1 for a complete description of the log-polar mapping.

$$\begin{aligned}
\gamma_1 &= \frac{g_\xi \cos \vartheta - g_\eta \sin \vartheta}{\rho} \\
\gamma_2 &= \frac{g_\xi \sin \vartheta + g_\eta \cos \vartheta}{\rho} \\
\gamma_3 &= g_\xi \cos 2\vartheta - g_\eta \sin 2\vartheta \\
\gamma_4 &= g_\xi \sin 2\vartheta + g_\eta \cos 2\vartheta
\end{aligned} \tag{33}$$

ρ_0 , $1/q$, K_ξ being the log-polar layout parameters, ρ , θ the polar coordinates. A least square approach is used, as before, and six points are necessary in order to estimate the model parameters. Usually the system is solved considering all the available points.

Actually, Horn's equation tells something more about the problem – that is, optic flow cannot be determined on the basis of local information alone. A suitable approach is that of extending the spatial domain by applying the same equation to a small neighborhood of pixels: by having more constraints a solution can be found, if we assume the flow field locally constant. This observation has a link to receptive field size of the neural pathway devoted to optic flow estimation. In fact, though motion is first sensed through large arrays of visual responsive neurons (Borst & Egelhaaf, 1993), each of them tuned to a particular direction, areas MT and MST that are known to participate in optic flow “interpretation” have broad RFs. This might reflect the necessity to overcome the ill-posed nature of the problem itself – local information alone is not enough to extract velocity information. Graziano and colleagues characterized the tuning curves of such neurons in the rhesus monkey MSTd area and showed response to diverging, rotating and spiraling flow fields (Graziano, Andersen, & Snowden, 1994).

5.2.1 An alternative approach based on the RF concept

The approach described above can be applied for any flow model, although it would not always be clear when and where the model itself fails to be a good description of the actual optic flow – i.e. for the affine model, the requirement is that the moving scene must be roughly planar. Another possible choice is:

$$\dot{X} = \sum_{i=1}^n C_i \Phi(x, t_i) \tag{34}$$

where \dot{X} is the flow model, which is linear in its parameters C_i . $\Phi(x, t_i)$ are vector functions parameterized by t_i , which are determined *a priori* – $\Phi(x, t_i)$ are sometimes called basis fields bearing a resemblance with the more traditional basis functions used in standard approximation problems. As before, a combination of equation (30) and (34) yield a scalar equation:

$$\left(\sum_{i=1}^n C_i \Phi(t_i) \right) \cdot \left[\frac{\partial E}{\partial x} \quad \frac{\partial E}{\partial y} \right] = - \frac{\partial E}{\partial t} \quad (35)$$

Note that notation has been simplified by dropping the image coordinates x and y . Pre-multiplying by the log-polar Jacobian matrix converts equation (35) to the log-polar domain:

$$\left(\sum_{i=1}^n C_i \Phi(t_i) \right) \cdot J_{lp} \cdot \left[\frac{\partial E}{\partial x} \quad \frac{\partial E}{\partial y} \right] = - \frac{\partial E}{\partial t} \quad (36)$$

and,

$$J_{lp} = \begin{bmatrix} \frac{K_\xi}{\rho \ln a} \cos \vartheta & \frac{K_\xi}{\rho \ln a} \sin \vartheta \\ \frac{-q}{\rho} \sin \vartheta & \frac{q}{\rho} \cos \vartheta \end{bmatrix} \quad (37)$$

One might wonder why this approach is RF based. By analyzing the solution – the coefficient C_i – we can note that each value is actually a weight factor of the corresponding vector field. In practice, a C_i represents the response of a particular vector feature detector, tuned for optimally detecting a particular flow pattern in a particular image region – i.e. each basis field represents a receptive field. By combining many basis fields, the total flow can be reconstructed, though other interpretations are still possible. For instance, some particular detectors can be directly linked to reflex-like actions as, for example, a fast avoidance behavior to looming stimuli (Gandolfo, Sandini, & Bizzi, 1996).

5.3 Stabilization index

Stabilization performances can be evaluated either by estimating image motion through optic flow processing or by using a sort of correlation measure over time. In the latter case, the image stabilization index (ISI) could be defined as follows: $ISI=1-NC_i$,

$$NC_i = \frac{\sum_{\eta, \xi} (I_i(\eta, \xi) - \mu_i)(I_{i-1}(\eta, \xi) - \mu_{i-1})}{\sqrt{\sum_{\eta, \xi} (I_i(\eta, \xi) - \mu_i)^2 \sum_{\eta, \xi} (I_{i-1}(\eta, \xi) - \mu_{i-1})^2}} \quad (38)$$

The correlation measures the degree of similarity between two subsequent log-polar images (I_i and I_{i-1}). The symbols μ_i and μ_{i-1} indicate the corresponding image mean values. Better stabilization performance is mapped to lower values of ISI.

5.4 Color segmentation

Color segmentation allows the robot to extract the position of objects from visual information. Though very simple, it offers several advantages in terms of robustness. Object position itself can be used in position-based feedback control loops, as shown in section 3.1.

In general terms, a color segmentation procedure, should identify the “principal” object color and separate it from that of the background – imagine for the moment that the color of the object is sufficiently different from that of the whole scene. An appropriate color representation provides a more efficient way of dealing with color information. This is to say that the first step of the color processing is the conversion of RGB information to some other representation – in our case hue, saturation, and value (HSV). The HSV transform allows separating brightness information – contained in the V component – from color related information (mostly hue and saturation).

A cueing procedure is applied to initially locate the object of interest; in our case, a motion detection procedure has been used. In practice, with the robot initially still, a temporal differencing processing detects a potential target for tracking; a histogram of moving pixels in HS space is then constructed in order to group the pixels belonging to the object.

That is, the region of the histogram representing the moving object can be located around the histogram maximum, and isolated by a region growing procedure. This basically determines which part of the color space

represents the target, under the assumptions that the object has a dominant color with a sufficient number of pixels. All these conditions are checked at run time by the algorithm.

A second histogram regarding the background pixels is built, and updated online for segmentation purposes – if the two histograms differ enough, that is the object can be reliably distinguished from the background, the proper color segmentation is started. This last stage is carried out by simply checking whether a pixel belongs to the object histogram. Once all pixels are checked, the position of the object is computed by estimating the center of mass of such pixels belonging to the object (i.e. those which passed the checking phase).

Figure 29 shows an example of the color segmentation procedure; the upper panel (a) shows the result of the processing, applied over a 32×64 (=2048) pixel log-polar image: the left image is the original image remapped into the Cartesian space; the right image illustrates the segmented region. The lower panel (b) shows the corresponding HS histograms: the background on the left and the object on the right.

All the computation can be easily carried on in real-time (i.e. 25 Hz) without resorting to any specialized hardware. From the software point of view, the update of the background histogram over time allows the robot to adapt rapidly to changes in the background color and, consequently, adds generality to the approach. As can be noted in Figure 29, there are no particular requirements in terms of background texturing, i.e. the systems performs equally well whether or not the background is cluttered.

5.5 The learning module: a growing neural network

In this section we introduce an incremental algorithm to train a class of networks that are interesting from both the biological plausibility and the statistical viewpoint. It inherits aspects from both the Growing Neural Gas (GNG) model (Fritzke, 1995), and the SoftMax basis function networks (Morasso & Sanguineti, 1995). The GNG is an unsupervised network model, which learns topologies. A set of units connected by edges is distributed in the input space (a subset of \mathfrak{R}^N) with an incremental mechanism, which tends to minimize the mean distortion error. For this reason the distortion error is locally accumulated and a new unit is inserted near the unit with maximum error. At every learning step a subset of units (the winner and its neighbors) is moved following a Hebbian learning rule. Among its properties, GNG is also able to adapt to locally varying dimensionality depending indeed on the input data set. A detailed description of the algorithm can be found in (Fritzke, 1994).

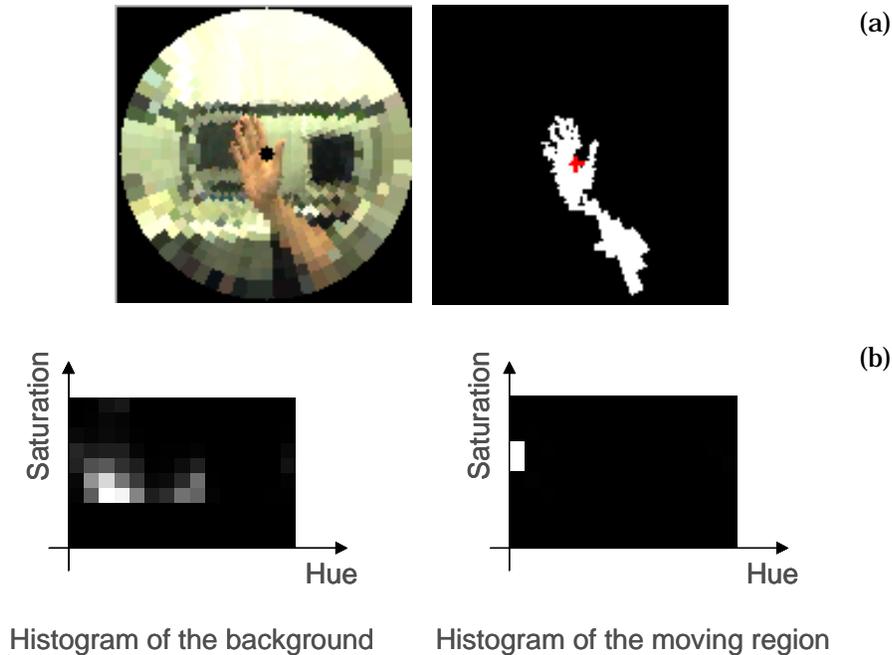


Figure 29 Color processing. The upper row (a) shows a typical image from the robot's point of view: original image (left) and color segmented image (right). All the processing is carried out in the log-polar domain; images are mapped back to the Cartesian space for visualization purposes. The lower row (b) contains the HS histograms: background (left) and object (right). Note that, as the histograms do not overlap, segmentation can be performed reliably. Hue values range from 0 (red) to 360. Saturation ranges from 0 (gray level) to 1 (full color). The V component has not been used for segmentation purposes in order to enhance robustness to changes in illumination.

A SoftMax function network ∇ consists of a single layer of *processing elements* (PEs) characterized by a receptive field centered on a preferred vector. Each unit's activation function U_i is a SoftMax function whose analytic expression is:

Deleted: (Morasso & Sanguineti, 1995)

$$U_i(x, c_i) = \frac{G(\|x - c_i\|)}{\sum_j G(\|x - c_j\|)} \quad G(x) = e^{-\frac{x^2}{2\sigma^2}} \quad (39)$$

where $G(\bullet)$ is a Gaussian function and c_i is the center of the activation function. Here the function U_i has its maximum. Benaim and Tomasini (Benaim & Tomasini, 1991) proposed a Hebbian learning rule for the optimal placement of PEs:

$$\Delta c_i = \eta_1 (x - c_i) U_i(x, c_i) \quad (40)$$

where $x \in \mathfrak{R}^N$ is an input pattern and η_1 the learning rate. Indeed, a SoftMax network can learn a smooth non-linear mapping $z = z(x)$. The reconstruction formula in this case is:

$$z(x) \cong \sum_i v_i U_i(x, c_i) \quad (41)$$

where the parameters v_i are the weights of the output layer and $z \in \mathfrak{R}^M$. In particular, considering an approximation case, this formula can be interpreted as a minimum variance estimator (Specht, 1990). The learning rule is:

$$\Delta v_i = \eta_2 (z - v_i) U_i(x, c_i) \quad (42)$$

This schema has been used to model cortical maps (Durbin & Mitchison, 1990). The normalizing factor of U can be seen as a lateral inhibition mechanism (Morasso & Sanguineti, 1995).

It is now evident how to combine the two self-organizing map previously illustrated to obtain an incremental and plastic network model with the best features of both techniques. The resulting model will be characterized by the effectiveness, typical of the GNG, in the distribution of the units within the n-dimensional input space and by the strength approximation and interpolation properties of SoftMax functions networks. We devised a heuristic criterion in order to self-tune variances. At each

learning step the variance of the winning unit and its neighbors is updated using the following rule:

$$\sigma_i^2 = k \cdot \hat{d}_i^2 \quad (43)$$

where \hat{d}_i^2 is the mean squared distance from the unit i and its neighbors, and k is a positive constant. It is worth noting that the formula is largely similar to that suggested by Fritzke in (Fritzke, 1994). However, a scaling factor (represented by k) has been added to guarantee a substantial overlap of the basis function tails.

The complete algorithm is the following (supervised learning):

1. Start with two units a, b at random positions c_a, c_b and with two associated output vectors v_a, v_b . Let \mathbf{A} be the set of units and $\mathbf{C}=\mathbf{A}\times\mathbf{A}$ the set of connections between them. \mathbf{A} is initialized as $\mathbf{A}=\{a,b\}$ and consequently \mathbf{C} contains the connection between a and b .
2. Generate an input data pair (ξ, ζ) , with $\xi \in \mathcal{R}^N$, $\zeta \in \mathcal{R}^M$, and $\zeta = f(\xi)$.
3. Find the nearest unit (winner) and the second nearest unit: s_1, s_2 .
4. If a connection between s_1 and s_2 does not exist already, create it, and set its age to zero (i.e $\mathbf{C} = \mathbf{C} \cup \{(s_1, s_2)\}$).
5. If $g(\xi)$ is the output of the network, add the squared distance between the actual output and the desired one to the accumulated error of the winner.

$$g(\xi) = \sum_i v_i U_i(\xi, c_i) \quad (44)$$

U_i described by equation (39)

6. Move s_1 and its neighbors towards (ξ, ζ) and update the respective variances using equation (43).

$$\Delta c_i = \eta_1(\xi - c_i) U_i \quad (45)$$

$$\Delta v_i = \eta_2(\zeta - v_i) U_i \quad (46)$$

7. Increment the age of all the connections (in \mathbf{C}) emanating from s_1 .

8. Remove connections with an age larger than a_{max} . Remove all units with no emanating connections.
9. If the overall number of input samples is an integer multiple of a parameter λ and an error criterion is satisfied (see description below) insert a new unit r between the one with maximum accumulated error q and its neighbors with maximum accumulated error f with the appropriate variance.

$$c_r = \frac{(c_q + c_f)}{2} \quad (47)$$

$$v_r = \frac{(v_q + v_f)}{2} \quad (48)$$

10. Decrease the accumulated error of q and f by a fraction α .
11. Set the accumulated error of the new unit equal to half of the sum of accumulated errors of q and f .
12. Decrease accumulated error of all units by a fraction β .
13. If a stopping criterion is not yet fulfilled, continue from step 2.

The estimation of the error and the relative condition (step 9) needs special mention. In fact, one of the most challenging problems concerning these networks is that of tuning the network growth. If we were able to estimate the error on-line, we could tailor the units insertion/removal mechanisms to the current approximation request. We propose the following estimation rule:

$$e_t = \psi_1 e_{t-1} + (1 - \psi_1) \|\zeta - g(\xi)\| \quad (49)$$

and,

$$\dot{e}_t = \psi_2 \dot{e}_{t-1} + (1 - \psi_2) \cdot (e_t - e_{t-1}) \quad (50)$$

The behavior of e_t follows that of the instantaneous error $\|\zeta - g(\xi)\|$ although with a sort of memory. In fact, it is a low pass filtered version of the raw error signal. $\psi_{1,2}$ are positive constants, which determine the filters cut-off frequency. \dot{e}_t is an estimate of the derivative of the error. The on-line measures have been used to block the insertion of new units according to the following conditions:

$$e_i < threshold_1 \quad (51)$$

and,

$$\|\dot{e}_i\| > threshold_2 \quad (52)$$

The derivative of the error was checked and insertion was allowed if, and only if, it was approximately zero – i.e. the error itself does not decrease anymore. We would like also to point out that e_i can be computed over a test set bigger than and/or different to the training set, hence providing an independent estimate of network performances. It could be used to validate the network model itself and it makes sense to tune the network growth rate on the basis of it (if used in this sense, it provides a sort of cross-validation mechanism).

5.6 The force field approach to motor control

Isolated skeletal muscles act like non-linear visco-elastic actuators whose length-tension properties are modulated by neuromuscular activation (Rack & Westbury, 1969). For the scope of the present work, however, a simplified model (Kandel et al., 1991) has been used to express the torque exerted by a muscle on each joint:

$$\tau = -a\kappa(q - q_0) \quad (53)$$

where q_0 is the actuator's resting position, a the activation value which modulates the overall stiffness κ (i.e. the spring constant of the muscle).

Assuming this model, a possible procedure for coding motor commands is the so-called force field approach proposed by Mussa-Ivaldi and Bizzi (Gandolfo & Mussa-Ivaldi, 1993), (Mussa-Ivaldi & Giszter, 1992), (Mussa-Ivaldi et al., 1993), (Mussa-Ivaldi, 1997). According to this theory, the neuromuscular torque exerted by each actuator can be described by means of a torque field:

$$\tau(\mathbf{q}, a) \quad (54)$$

where \mathbf{q} is the vector of generalized coordinates, a is the activation value and τ is the generalized torque field. In the case of a multi-joint structure (such as a limb) the overall torque is expressed by:

$$\boldsymbol{\tau} = \sum_i \boldsymbol{\tau}_i(\mathbf{q}, a_i) \quad (55)$$

where a_i are the control parameters.

From the mechanical point of view, the system controlled by these actuators is passive. Consequently, it has a stable Equilibrium Point (EP) in its state space $(\mathbf{q}, \dot{\mathbf{q}})$. The EP is i) the point where the torque field described by equation (55) is zero; ii) the intersection of the actuators' angle-tension curves. If we apply the torques described by equation (55) to the multi-joint structure, its state will eventually reach the EP (at equilibrium). Thus, the EP can be thought of as the point toward which the configuration is aiming at each instant of time.

In theory, specification of the EP suffices in driving the system to a given configuration. On the other hand, experimental results in animals and humans (Mussa-Ivaldi et al., 1993) support a rather different view. In fact, it seems that shifting the EP smoothly from the start to the end, rather than suddenly moving it to the target position, causes the limb to move. The sequence of EPs defines what is called a "virtual trajectory" (Hogan, 1985). It is worth noting that, the arm trajectory is different from the virtual trajectory – in other words it is like pulling a toy car with a rubber band: the trajectory in space of the pulling hand is different from the trajectory of the car, because of the stretching of the rubber band.

The simplification – and in some sense the feasibility – of this schema comes from the experimental observation that any position of the EP in the arm configuration space (and consequently its motion) can be obtained by a linear combination of a small number of motor primitives each represented as a torque field (the so-called "basis fields" (Mussa-Ivaldi, 1992), (Mussa-Ivaldi et al., 1993)). In our model, each motor primitive is a structure, which activates a single or a group of actuators. It is actually a synergy, which combines (linearly) the effect of a set of actuators by activating them synchronously by means of only one control parameter. The following torque field describes the primitives:

$$\mathbf{T}_j(\mathbf{q}, C_j) = \sum_i \mathbf{I}_{ji} \boldsymbol{\tau}_i(\mathbf{q}, C_j) \quad (56)$$

where $\boldsymbol{\tau}_i$ is the i^{th} actuator field, C_j the activation value and:

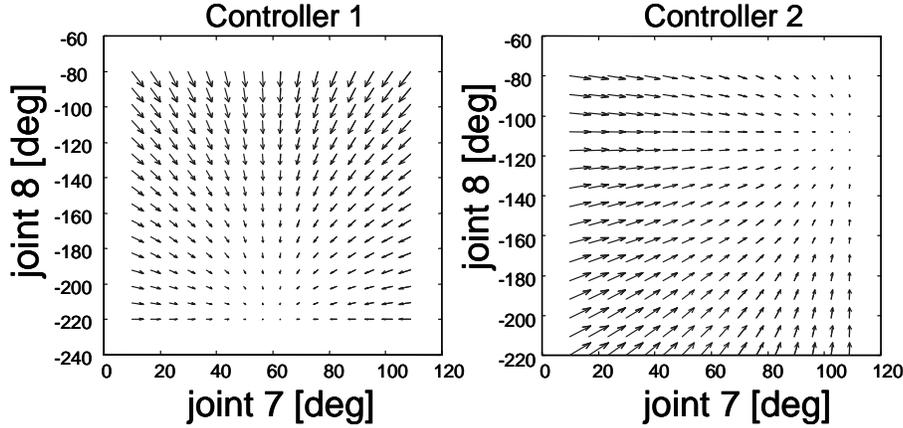


Figure 30 Two exemplar basis fields represented as torque fields in joint coordinates. Ordinate and abscissa show joint position (Babybot's joint 7 and 8 – shoulder and elbow respectively) in degrees. Arrows point to the common EP of the two joints. The actual resting position of each actuator q_0 was defined beforehand.

$$\mathbf{I}_{ij} = \begin{cases} 1 & \text{if the } j^{\text{th}} \text{ controller activates the } i^{\text{th}} \text{ actuator} \\ 0 & \text{otherwise} \end{cases} \quad (57)$$

\mathbf{T}_j are exactly the basis fields as shown in Figure 30. The total field \mathbf{T} is expressed by the following:

$$\mathbf{T}(\mathbf{q}) = \sum_j \mathbf{T}_j(\mathbf{q}, C_j) = \sum_j \sum_i \mathbf{I}_{ji} \boldsymbol{\tau}_i(\mathbf{q}, C_j) \quad (58)$$

We designed the connections between actuators and primitives (through \mathbf{I}_{ij}) *a priori*. In our case, the basis fields are constant and embedded into the system. Given this assumption, the task of the controller is to combine the basis fields by providing, for each point of the configuration space, a set of control parameters C_j . A schematic diagram of

the controller was shown in Figure 23 (section 3.7) in the case of 4 basis fields and 2 joints.

A further simplification, allowed by the force field approach, comes from the fact that control parameters are not dependent on any particular frame of reference (Mussa-Ivaldi & Giszter, 1992). This is easily shown converting equation (58) into extrinsic coordinates. Let $\mathbf{x}=\Lambda(\mathbf{q})$ be the direct kinematics mapping of the arm and \mathbf{J}_Λ its Jacobian. For any configuration where \mathbf{J}_Λ is not singular we can write:

$$\mathbf{J}_\Lambda^{-T} \boldsymbol{\tau} = \mathbf{F} \quad (59)$$

where \mathbf{J}_Λ^{-T} is the transposed inverse Jacobian, $\boldsymbol{\tau}$ the torque vector and \mathbf{F} the corresponding force vector in extrinsic coordinates. Substituting equation (59) in equation (58) and considering linear actuators (as in equation (53)) yields:

$$\mathbf{J}_\Lambda^{-T} \mathbf{T} = \sum_j C_j \mathbf{J}_\Lambda^{-T} \mathbf{T}_j \quad (60)$$

where $\mathbf{J}_\Lambda^{-T} \mathbf{T} = \mathbf{F}$ is the total force field and $\mathbf{J}_\Lambda^{-T} \mathbf{T}_j = \mathbf{F}_j$ are the basis fields in extrinsic coordinates. Substituting yields:

$$\mathbf{F} = \sum_j C_j \mathbf{F}_j \quad (61)$$

Equation (61) shows that the control coefficients C_j are invariant under coordinates transformation (for a discussion of the underlying conditions, see (Mussa-Ivaldi & Giszter, 1992)). A similar result applies for the redundant case (where \mathbf{J}_Λ is not invertible), depending on the motor primitives considered (Gandolfo & Mussa-Ivaldi, 1993). Given these results, it is correct to freely exchange torque fields generated by actuators with force fields applied to the arm end-point because the two representations are indeed equivalent. Furthermore, this approach allows us: i) to embed the kinematic parameters in the resulting force fields (Hogan, 1985); ii) to represent the activation of a synergy of muscles as a force field, consequently several joints can be controlled by using only one control parameter – it is clear that in this case a multi-joint motion is still behaving as a single degree of freedom, though it is then easy to represent

“innate” motor synergies and reflexes. Figure 30 and Figure 31 show two exemplar basis fields as used in our experiments; Figure 30 shows two torque fields in joint coordinates, while Figure 31 plots the same two fields converted in Cartesian coordinates.

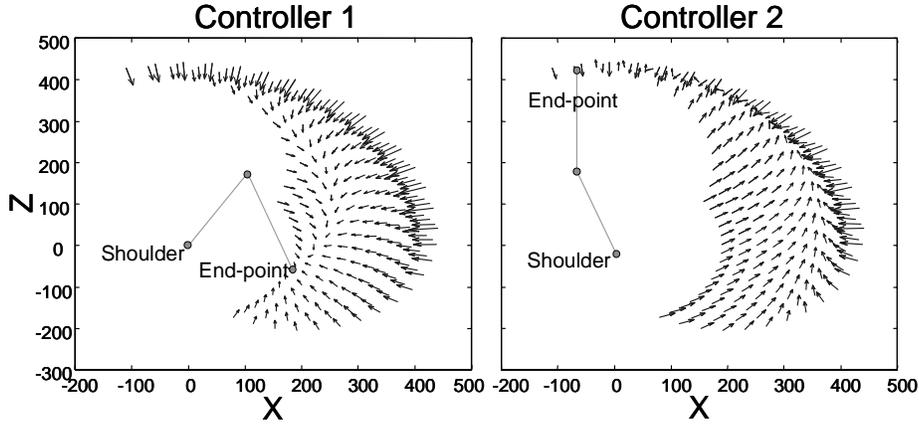


Figure 31 Two exemplar basis fields represented in Cartesian coordinates (see also Figure 30). Ordinate and abscissa represent the plane where the arm motion has been constrained – see experimental session on section 3.7.

5.6.1 From the EP to the activation values

Up to now we have not addressed the problem of generating the appropriate activation values, so that the resulting force field converges to a desired equilibrium point. Following the procedure suggested by Mussa-Ivaldi and colleagues (Mussa-Ivaldi & Giszter, 1992), the problem can be formulated as a function approximation problem by using a finite set of basis functions:

$$\mathbf{T} \approx \sum_j C_j \sum_i \mathbf{I}_{ji} \boldsymbol{\tau}_i \quad (62)$$

where \mathbf{T} is the desired resulting total field, and $\boldsymbol{\tau}$ is the basis fields. The continuous problem can be converted into a discrete one by sampling the input space Q in k points. In other words, if we know the desired total field in k points, it is possible to compute the best approximation of that field by

using a linear combination of basis fields. Equation (62) can be rewritten as:

$$\mathbf{T}(\mathbf{q}_k) \approx \sum_j C_j \sum_i \mathbf{I}_{ji} \boldsymbol{\tau}_i(\mathbf{q}_k) \quad (63)$$

where \mathbf{q}_k is the set of sampling points. If the number of sampling points, K , times the number of joints, N , is greater than the number of unknowns, J , a least squares approach allows us to determine the vector \mathbf{C} that best approximates the desired total field.

A particular desired force field is the converging pattern with a single EP. This represents a position controller; by varying the resting point position, it is possible to smoothly control the motion of the robot's end-point as discussed before. An example of the converging pattern, as used in our implementation, is shown in Figure 32.

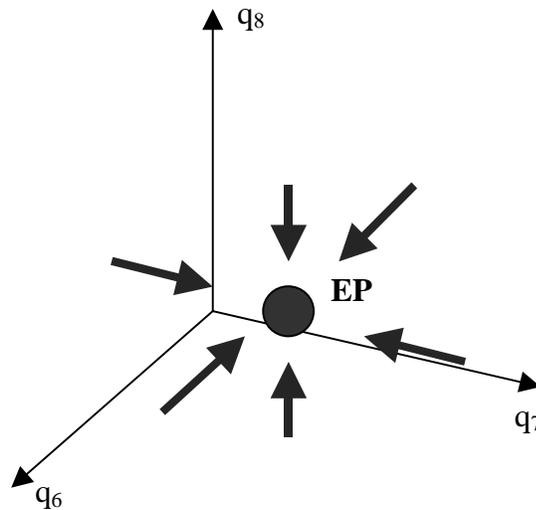


Figure 32 Exemplar converging force field as employed by Babybot's controller. The 7 vectors together with the knowledge of the basis fields, allow us to recover the activation values, from there a simulation of the spring-like actuators allow computing, for each time step, a corresponding torque vector. The torque vector eventually is converted to the appropriate currents, which drive the motors.

5.7 The technological corner-shop

This section, deals with the software architecture of the Babybot. The robot code is based on a standard known as DCOM (acronym for Distributed Component Object Model), which allows working with “components” in an object-oriented framework and, furthermore, allows running the objects across a network. The key aspect of the standard is the “componentization”. Just to have an idea, imagine you have to build a circuit, what you do nowadays is to buy an off-the-shelf implementation – that is an IC – and use it without worrying much about its actual implementation (e.g. an NAND gate). You just plug it to whatever application as far as it adheres to a standard in terms of electrical compatibility.

With software components you basically apply the same sort of paradigm. DCOM components have to adhere to a “binary standard” and, in doing so they guarantee they can be plugged in whatever software application. Of course, the price to pay is that the standard is even stricter than a usual programming language, in the sense that the interface toward the external world must be strictly determined.

The main concept behind a DCOM object is that of “interface”. An interface is the specification of how the object presents itself to the external world. The interface has to be standardized, and in fact, there is a descriptive language to write interfaces – i.e. IDL (Interface Description Language). Once the interface is defined, the object functionality can be implemented by using whatever programming language (the standard is independent of the language). Just to clarify, consider that an interface is not a class, it is not an object, it is immutable, and that clients interact only through pointer to interfaces. There is no way for a client to access the real object implementation – in DCOM the interface is the only access port.

Another peculiarity of the DCOM architecture is the identification of objects. In practice, each interface is signed with a “globally unique identifier” (GUID). This signature is what matters when instantiating an object, so when requesting a particular interface, we are assured that the right object will be created. Furthermore, this feature deals also with the problem of versioning; its “unique” signature allows recognizing an older version of the same object. Consequently, clients can appropriately handle this situation, and even recognize a newer version of the same kind of object.

Because, the standard is at the binary level, location transparency is guaranteed. For the client there is no way to distinguish between a local

object and a remote one. Of course, if the object is on the local machine, it could be faster – a local object can be instantiated as in-process. Code reusability comes with the framework; it is possible to reuse code and, as in object-oriented practice, to use objects inside other objects. Finally, though unusual, the transport layer can be customized and data of any type can be efficiently sent through different network mediums.

How does this influence Babybot's architecture? From the software point of view, the use of components allows layering of code, and creating a separation from the "low level handling of hardware resources" (e.g. frame grabbers, control boards, etc) and the "high level implementation" (e.g. control loops, learning, etc). In this sense, once a component, for instance the neural network algorithm, is designed it may be reused throughout the system. Furthermore, we can really design a "distributed system" with a uniform programming environment (in spite of the OS – DCOM is an open standard). This can be seen as a "coarse grain" parallel system, where the smallest unit is a single PC. Those PCs can be easily connected through a fast Ethernet, or even a Gigabit network if this were the requirement. Flexibility and expandability are easily obtained by connecting a new machine to the network.

Objects, in this scenario, are distributed across machines and, consequently, at any moment any object can access every other object in the system with ease. This development platform takes, in some sense, the opposite approach to the current engineering practice. Where many control systems are centralized – either on a single machine or on a single bus system – the Babybot's architecture is distributed and connected through a "relatively slow" (if compared to a bus) network. Where others used real-time OS, we used a soft real-time OS. Where synchronicity and tight scheduling is an issue, we employed an asynchronous control system, where nothing about timing is granted.

The schematics below (Figure 33) show how the system is put together and what the machine roles are within the control structure. In particular, Babybot is presently controlled by four PCs, one mounting frame grabbers for visual processing, a second one dedicated to the head control, and a third to the arm control. The fourth machine is employed for monitoring purposes and does not carry out any relevant processing.

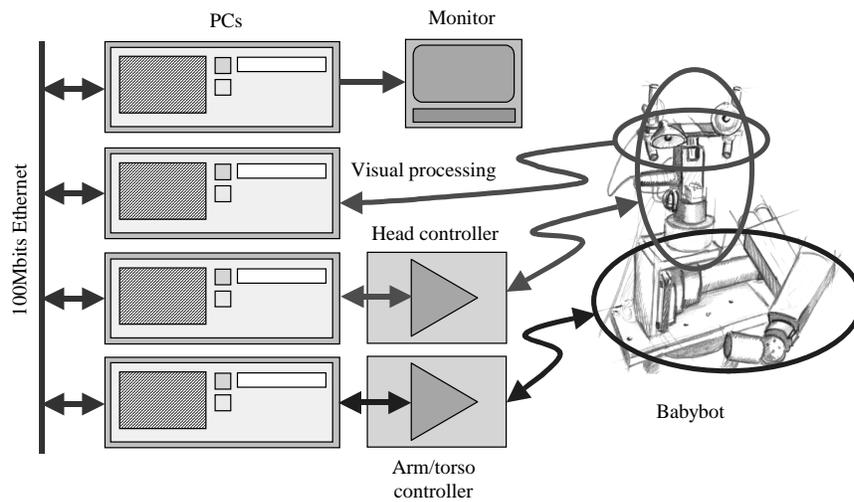


Figure 33 The Babybot distributed architecture. Presently, it consists of four PCs connected through a 100Mbit/s Ethernet link. Some of them are equipped with frame grabbers, axis control boards, and ADCs. From the software point of view, the adopted framework allows uniform programming and object distribution across machine boundaries. In practice the whole system can be seen as a unique processing machine where internal boundaries are actually as thin as possible.

Bibliography

- Abend, W., Bizzi, E., & Morasso, P. (1982). Human Arm Trajectory Formation. *Brain*(105), 331-348.
- Adolphs, R., Tranel, D., & Damasio, A. (1998). The human amygdala in social judgement. *Nature*(393), 470-474.
- Allman, J. M., & Kaas, J. H. (1971). Representation of the visual field in striate and adjoining cortex of the owl monkey (*Aotus trivirgatus*). *Brain Research*, 35, 89-106.
- Aloimonos, J., Weiss, I., & Bandyopadhyay, A. (1988). Active Vision. *International Journal of Computer Vision*, 1(4), 333--356.
- Atkinson, J. (1998). The 'Where and What' or 'Who and How' of Visual Development. In F. Simion & G. Butterworth (Eds.), *The Development of Sensory, Motor and Cognitive Capacities in Early Infancy* (pp. 3-24). Hove, East Sussex: Psychology Press Ltd.
- Atkinson, J., & Braddick, O. (1981). *Acuity, contrast sensitivity, and accommodation in infancy*. New York: Academic Press.
- Baarsma, E. A., & Collewijn, M. (1974). Vestibulo-Ocular and Optokinetic Reactions to Rotations and Their Interaction in the Rabbit. *Journal of Physiology*(238), 603-625.
- Bajcsy, R. K. (1985). *Active Perception vs. Passive Perception*. Paper presented at the Third IEEE Workshop on Computer Vision: Representation and Control, Bellaire (MI).
- Ballard, D. H., & Brown, C. M. (1992). Principles of Animate Vision. *Computer Vision Graphics and Image Processing*, 56(1), 3-21.
- Barron, A. R. (1993). Universal Approximation Bounds for Superpositions of a Sigmoidal Function. *IEEE Transaction on Information Theory*, 39(3), 930-945.
- Barron, J. L., Fleet, D. J., & Beauchemin, S. S. (1994). Performance of optical flow techniques. *International Journal on Computer Vision*, 12(1), 43-77.
- Beer, R. D., Chiel, H. J., Quinn, R. D., & Ritzmann, R. E. (1998). Biorobotic approaches to the study of motor systems. *Current Opinion in Neurobiology*, 8(6), 777-782.
- Bekoff, A., Kauer, J. A., Fulstone, A., & Summers, T. R. (1989). Neural control of limb coordination. *Experimental Brain Research*, 74, 609-617.

- Bellman, R. E. (1956). *Dynamic Programming*. Princeton: Princeton University Press.
- Benaim, M., & Tomasini, L. (1991). Competitive and Self-organizing Algorithms Based on the Minimization of an Information Criterion. In T. Kohonen, K. Makisara, O. Simula, & J. Kangas (Eds.), *Artificial Neural Networks* (pp. 391-396). Amsterdam, North-Holland: Elsevier Science Publisher.
- Benson, A. J., Guedry, F. E., & Melvill Jones, G. (1970). Response of Semicircular Canal Dependent Units in Vestibular Nuclei to Rotation of a Linear Acceleration Vector Without Angular Acceleration. *Journal of Physiology*(210), 475-494.
- Berthouze, L., Bakker, P., & Kuniyoshi, Y. (1996). *Learning of Oculo-Motor Control: a Prelude to Robotic Imitation*. Paper presented at the IEEE-RSJ -IROS 1996.
- Berthouze, L., & Kuniyoshi, Y. (1998). Emergence and Categorization of Coordinated Visual Behavior Through Embodied Interaction. *Machine Learning*(31), 187-200.
- Biguer, B., & Prablanc, C. (1981,). Modulation of the Vestibulo-Ocular Reflex in Eye-Head Orientation as a Function of Target Distance in Man.
- Bizzi, E. (1974). The Coordination of Eye-Head Movements. *Scientific American*, 231(4), 100-106.
- Borst, A., & Egelhaaf, M. (1993). Visual Motion and its Role in the Stabilization of Gaze, *Visual Motion and its Role in the Stabilization of Gaze* (pp. 3-27). Amsterdam: Elsevier.
- Bower, T. G. R., Broughton, J. M., & Moore, M. K. (1970). The coordination of visual and tactual input in infants. *Perception and Psychophysics*, 8, 51-53.
- Brooks, R. (1986). A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics & Automation*, 2, 237-244.
- Brooks, R. (1989). A robot that Walks: Emergent Behaviour from a Careful Evolved Network. *Neural Computation*.
- Brooks, R. (1996). *Behavior-Based Humanoid Robotics*. Paper presented at the IEEE/RSJ IROS'96.
- Bushnell, E. W. (1981). The Ontogeny of Intermodal Relations: Vision and Touch in Infancy. In R. Walk & H. Pick (Eds.), *Intersensory Perception and Sensory Integration* (pp. 5-37). New York: Plenum Press.
- Caldwell, D. G., Medrano-Cerda, G. A., & Bowler, C. J. (1997, April, 20-25). *Investigation of bipedal robot locomotion using pneumatic*

- muscle actuators*. Paper presented at the IEEE International Conference on Robotics and Automation, Albuquerque, USA.
- Capurro, C., Panerai, F., Grosso, E., & Sandini, G. (1993, July). *A Binocular Active Vision System Using Space Variant Sensors: Exploiting Autonomous Behaviors for Space Application*. Paper presented at the International Conference on Digital Signal Processing, Nicosia, Cyprus.
- Capurro, C., Panerai, F., & Sandini, G. (1995, April). *Space Variant Vision for an Active Camera Mount*. Paper presented at the SPIE AeroSense95, Orlando, Florida.
- Carpenter, G., & Grossberg, S. (1986). *Adaptive Resonance Theory: stable self-organization of neural recognition codes in response to arbitrary list of input patterns*. Paper presented at the Eighth Annual Conference on the Cognitive Science.
- Carpenter, R. H. S. (1988). *Movements of the Eyes*. (Second ed.). London: Pion Limited.
- Carpenter, R. H. S. (1999). A Neural Mechanism that Randomises Behaviour. *Journal of Consciousness Studies*, 6(1), 13-22.
- Cioni, G., Favilla, M., Ghelarducci, B., & La Noce, A. (1984). Development of the Dynamic Characteristics of the Horizontal Vestibulo-Ocular Reflex in Infancy. *Neuropediatrics*, 15, 125-130.
- Clifton, R. K., Muir, D. W., Ashmead, D. H., & Clarkson, M. G. (1993). Is visually guided reaching in early infancy a myth? *Child development*, 64, 1099-1110.
- Collewijn, H., Erkelens, C. J., & Steinman, R. M. (1995). Voluntary Binocular Gaze-shifts in the Plane of Regard: Dynamics of Version and Vergence. *Vision Research*, 35(23-24), 3335-3358.
- Cowey, A. (1964). Projection of the Retina on to Striate and Prestriate Cortex in the Squirrel Monkey (*Saimiri Sciureus*). *Journal of Neurophysiology*(27), 266-293.
- Crane, B. T., Viirre, E. S., & Demer, J. L. (1997). The Human Horizontal Vestibulo-Ocular Reflex During Combined Linear and Angular Acceleration. *Experimental Brain Research*(114), 304-320.
- Crowell, J. A., Banks, M. S., Shenoy, K. V., & Andersen, R. A. (1998). Visual Self-Motion Perception During Head Turns. *Nature Neuroscience*, 1(8), 732-737.
- Crowley, J. L., Bobet, P., & Mesrabi, M. (1992, May). *Gaze Control with a Binocular Camera Head*. Paper presented at the European Conference of Computer Vision, Santa Margherita Ligure, Italy.

- Daniel, M., & Whitteridge, D. (1961). The Representation of the Visual Field on the Cerebral Cortex in Monkeys. *Journal of Physiology*(159), 203-221.
- Dennet, D. C. (1997). *L'idea pericolosa di Darwin - Orig. Darwin's dangerous idea. Evolution and the meaning of life.* (Frediani, S., Trans.). (First edition ed.). Torino: Bollati Boringhieri.
- Diamond, A. (1981). Retrieval of an Object from an Open Box: The Development of Visual-Tactile Control of Reaching in the First Year of Life. *Society of Research in Child Development (Abstracts)*(3), 78.
- Distler, C., Vital-Durand, F., Korte, R., Korbmacher, H., & Hoffmann, K. P. (1999). Development of the optokinetic system in macaque monkeys. *Vision Research*(39), 3909-3919.
- Durbin, R., & Mitchison, G. (1990). A dimension reduction framework for understanding cortical maps. *Nature*(343), 644-647.
- Eibl-Eibesfeld, I. (1970). *Ethology: the Biology of Behavior (E.Klinghammer trans.)*: Holt, Rinehart and Winston.
- Evgeniou, T., Pontil, M., & Poggio, T. (1999). *A unified framework for Regularization Networks and Support Vector Machines (AI Memo 1654)*. Boston, MA: MIT.
- Finocchio, D. V., Preston, K. L., & Fuchs, A. F. (1991). Infant Eye Movements: Quantification of the Vestibulo-Ocular Reflex and Visual-Vestibular Interactions. *Vision Research*, 31, 1717-1730.
- French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends in cognitive science*, 3(4), 128-135.
- Fritzke, B. (1994). Fast learning with incremental RBF Networks. *Neural Processing Letters*, 1(1), 2-5.
- Fritzke, B. (1995). A growing neural gas network learns topologies. In G. Tesauro, D. S. Touretzky, & T. K. Leen (Eds.), *Advances in Neural Information Processing Systems* (Vol. 7,). Cambridge MA, USA: MIT Press.
- Gandolfo, F., & Mussa-Ivaldi, F. A. (1993). *Vector summation of end-point impedance in kinematically redundant manipulators*. Paper presented at the IEEE/RJS International Conference of Intelligent Robots and Systems, Yokohama, Japan.
- Gandolfo, F., Sandini, G., & Bizzi, E. (1996). *A field-based approach to visuo-motor coordination*. Paper presented at the Workshop on Sensorimotor Coordination: Amphibians, Models, and Comparative Studies, Sedona, Arizona USA.

- Gandolfo, F., Sandini, G., & Tistarelli, M. (1991, June). *Towards vision guided manipulation*. Paper presented at the International Conference on Advanced Robotics, Pisa, Italy.
- Gdowski, G. T., & McCrea, R. A. (1999). Integration of Vestibular and Head Movement Signals in the Vestibular Nuclei During Whole-Body Rotation. *Journal of Neurophysiology*, *82*, 436-449.
- Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, *4*, 1-58.
- Ghez, C., Gordon, J., Ghilardi, M., & Sainburg, R. (1996). Contribution of vision and proprioception to accuracy of limb movements. In M. S. Gazzaniga (Ed.), *The cognitive neuroscience*. Cambridge: MIT Press.
- Gilmore, R. O., & Johnson, M. H. (1998). Learning What is Where: Oculomotor Contributions to the Development of Spatial Cognition. In F. Simion & G. Butterworth (Eds.), *The Development of Sensory, Motor and Cognitive Capacities in Early Infancy* (pp. 25-47). Hove, East Sussex: Psychology Press Ltd.
- Gomi, H., & Kawato, M. (1997). Human arm stiffness and equilibrium-point trajectory during multi-joint movement. *Biological Cybernetics*, *76*, 163-171.
- Goodale, M. A. (1989). Modularity in Visuomotor Control: from Input to Output. In Pylyshyn (Ed.), *Computational Processes in human vision: an interdisciplinary perspective* (pp. 262-285). Norwood, NJ: Ablex Publishing.
- Goossens, H. H. L. M., & Van Opstal, A. J. (1997). Human Eye-Head Coordination in Two Dimensions Under Different Sensorimotor Conditions. *Experimental Brain Research*(114), 542-560.
- Gould, J. L. (1982). *Ethology: the Mechanism and Evolution of Behavior*. Norton.
- Graziano, M. S. A., Andersen, R. A., & Snowden, R. J. (1994). Tuning of MST Neurons to Spiral Motions. *The Journal of Neuroscience*, *14*(1), 54-67.
- Grosso, E., Manzotti, R., Tiso, R., & Sandini, G. (1995, November). *A Space-Variant Approach to Oculomotor Control*. Paper presented at the IEEE International Symposium on Computer Vision, Coral Gables, Florida.
- Grosso, E., Metta, G., Oddera, A., & Sandini, G. (1996). Robust Visual Servoing in 3D Reaching Tasks. *IEEE Transactions Robotics and Automation*, *12*(8), 732-742.

- Hadders-Algra, M., Eykern, L. A. V., Nieuwendijk, A. W. J. K.-V. d., & Prechtl, H. F. R. (1992). Developmental course of general movements in early infancy: II, EMG correlates. *Early Human Development*, 28, 231-251.
- Held, R., & Bauer, J. A. (1967). Visually guided reaching in infant monkeys after restricted rearing. *Science*(155), 718-720.
- Held, R., & Hein, A. (1963). Movement produced stimulation in the development of visually guided behavior. *Journal of Comparative and Physiological Psychology*, 56, 872-876.
- Hennemann, E., & Mendell, L. M. (1981). . In V. Brooks (Ed.), *Handbook of physiology* (Vol. Sect. 1 Vol. 2 Part. 1, pp. 423-507). Washington D.C.: American Physiological Society.
- Hine, T., & Thorn, F. (1987). Compensatory eye movements during active head rotation for near targets: effects of imagination, rapid head oscillation and vergence. *Vision Research*(27), 1639-1657.
- Hirai, K., Hirose, M., Haikawa, Y., & Takenaka, T. (1998, May, 16-20). *The development of Honda humanoid robot*. Paper presented at the IEEE International Conference on Robotics and Automation, Leuven, Belgium.
- Hogan, N. (1985). The Mechanics of Multi-Joint Posture and Movement Control. *Biological Cybernetics*, 52, 313-331.
- Horn, B. K. P. (1986). *Robot Vision*. MIT Press.
- Horn, B. K. P., & Shunck, B. G. (1981). Determining Optical Flow. *Artificial Intelligence*, 17(1), 185-204.
- Hubel, D. H., & Wiesel, T. N. (1974). Uniformity of Monkey Striate Cortex: A Parallel Relationship between Field Size, Scatter, and Magnification Factor. *Journal of Computational Neuroscience*(158), 295-306.
- Hubel, D. H., & Wiesel, T. N. (1977). Functional architecture of macaque monkey cortex. *Proceedings of the Royal Society of London*(198), 1-59.
- Jordan, M. I. (1996). Computational Motor Control. In M. S. Gazzaniga (Ed.), *The Cognitive Neuroscience* (pp. 597-609). Cambridge MA - USA: MIT Press.
- Jordan, M. I., & Flash, T. (1994). A model of the learning of arm trajectories from spatial deviations. *Journal of Cognitive Neuroscience*, 4(6), 359-376.
- Kalveram, K. T. (1991). Controlling the dynamics of a two-joined arm by central patterning and reflex-like processing. A two stage hybrid model. *Biological Cybernetics*(65), 65-71.

- Kandel, E. R., Schwartz, J. H., & Jessel, T. M. (1991). *Principles of Neuroscience*: Elsevier.
- Kawato, M., Furukawa, K., & Suzuki, R. (1987). A hierarchical neural network model for control and learning of voluntary movement. *Biological Cybernetics*(57), 169-185.
- Keller, E. L. (1978). Gain of vestibulo-ocular reflex in the monkey at high rotational frequencies. *Vision Research*(18), 311-315.
- Koenderink, J., & Van Doorn, J. (1991). Affine Structure from Motion. *Journal of the Optical Society of America*, 8(2), 377-385.
- Konczak, J., Borutta, M., & Dichgans, J. (1997). Development of goal-directed reaching in infants: II. Learning to produce task-adequate patterns of joint torque. *Experimental Brain Research*, 113, 465-474.
- Konczak, J., Borutta, M., Topka, H., & Dichgans, J. (1995). Development of goal-directed reaching in infants: Hand trajectory formation and joint force control. *Experimental Brain Research*, 106, 156-168.
- Konczak, J., & Dichgans, J. (1997). Goal-directed reaching: development toward stereotypic arm kinematics in the first three years of life. *Experimental Brain Research*, 117, 346-354.
- Kuniyoshi, Y., & Cheng, G. (1999). Complex Continuous Meaningful Humanoid Interaction: A Multi Sensory-Cue Based Approach : Personal Communication.
- Kuperstein, M. (1988). Neural model of adaptive hand-eye coordination for single postures. *Science*, 239, 1308-1311.
- Leary, D. D. M. O. (1992). Development of connectional diversity and specificity in the mammalian brain by the pruning of collateral projections. *Current Opinion in Neurobiology*(2), 70-77.
- Ledoux, J. E. (1996). In search of an emotional system in the brain: leaping from fear to emotion and consciousness. In M. S. Gazzaniga (Ed.), *The Cognitive Neuroscience* (pp. 1049-1061). Cambridge, USA: MIT Press.
- Ljung, L. (1987). *System Identification: Theory for the User*. Englewood Cliffs, N.J.: Prentice-Hall.
- Lorentz, G. G. (1986). *Approximation of Functions*. New York: Chelsea Publishing Co.
- Majarrad, M., & Shahinpoor, M. (1997, April 20-25). *Biomimetic robotic propulsion using polymeric artificial muscles*. Paper presented at the IEEE International Conference on Robotics and Automation, Albuquerque, USA.

- Maringelli, F., & McCarthy, J. (1999). Shifting visuo-spatial attention in a three dimensional space: evidence in favour of a dissociation between attentional systems operating in the proximal and distal space : Personal Communication.
- Marr, D. (1982). *Vision*. San Francisco: Freeman W.H. and Company.
- Mathew, A., & Cook, M. (1986). The control of reaching movements by young infants. *Child Development*(61), 1238-1257.
- McCrea, R. A., Gdowsky, G. T., Boyle, R., & Belton, T. (1999). Firing Behavior of Vestibular Neurons During Active and Passive Head Movements: Vestibulo-Spinal and Other Non-Eye-Movement Related Neurons. *Journal of Neurophysiology*, 82, 416-428.
- Medendorp, W. P., Bakker, B. J., Van Gisbergen, J. A. M., & Gielen, C. C. A. M. (1999). Human Gaze Stabilization for Voluntary Off-Centric Head Rotations (Personal Communication) .
- Metta, G., Sandini, G., & Konczak, J. (1999). A Developmental Approach to visually-guided reaching in artificial systems. *Neural Networks*, 12(10), 1413-1427.
- Micheal, J. A., & Jones, G. M. (1966). Dependence of visual tracking capability upon stimulus predictability. *Vision Research*, 16, 707-716.
- Milani-Comparetti, A., & Gidoni, E. (1967). Routine developmental examination in normal and retarded children. *Developmental Medicine and Child Neurology*, 9, 631-638.
- Miles, F. A. (1997). Visual stabilization of the eyes in primates. *Current Opinion Neurobiology*(7), 867-871.
- Miles, F. A., Kawano, K., & Optican, L. M. (1986). Short-latency ocular following responses of monkey. I. Dependence on temporospatial properties of the visual input. *Journal of Neurophysiology*(56), 1321-1354.
- Milner, D. A., & Goodale, M. A. (1995). *The Visual Brain in Action*. (Vol. 27). Oxford: Oxford University Press.
- Morasso, P., & Sanguineti, V. (1995). Self-Organizing Body Schema for Motor Planning. *Journal of Motor Behavior*, 27(1), 52-66.
- Morasso, P., & Sanguineti, V. (1997). *Self-Organization, Cortical Maps, and Motor Control*. (Vol. 119): Elsevier Science.
- Mussa-Ivaldi, F. A. (1992). From basis functions to basis fields: vector field approximation from sparse data. *Biological Cybernetics*(67), 479-489.
- Mussa-Ivaldi, F. A., & Bizzi, E. (1989). Structural Constraints and Computational Problems in Motor Control. In P. Dario, G. Sandini,

- & P. Aebischer (Eds.), *Robots and Biological Systems: Toward a New Bionics?* (Vol. 102, pp. 786): Springer-Verlag.
- Mussa-Ivaldi, F. A., & Giszter, S. F. (1992). Vector field approximation: a computational paradigm for motor control and learning. *Biological Cybernetics*, *67*, 491-500.
- Mussa-Ivaldi, F. A., Giszter, S. F., & Bizzi, E. (1993). Convergent Force Fields Organized in the Frog's Spinal Cord. *The Journal of Neuroscience*, *13*(2), 467-491.
- Mussa-Ivaldi, F. D. (1997, July 10-11.). *Nonlinear Force Fields: A Distributed System of Control Primitives for Representing and Learning Movements*. Paper presented at the Proceedings of CIRA'97, Monterey, California, USA.
- Paige, G. D. (1991). Linear Vestibulo-Ocular Reflex (LVOR) and Modulation by Vergence. *Acta Otolaryngol Suppl*, *481*, 282-286.
- Panerai, F., Metta, G., & Sandini, G. (2000). Visuo-inertial Stabilization in Space-variant Binocular Systems. *Robotics and Autonomous Systems*, *30*(1-2), 195-214.
- Panerai, F., & Sandini, G. (1998). Oculo-Motor Stabilization Reflexes: Integration of Inertial and Visual Information. *Neural Networks*, *11*, 1191-1204.
- Pfeifer, R., & Scheier, C. (1997). Sensory-motor coordination: The metaphor and beyond. *Robotics and Autonomous Systems*, *20*, 157-178.
- Pfeifer, R., & Scheier, C. (1998, March, 8-12). *Representation in Natural and Artificial Agents: an Embodied Cognitive Science Perspective*. Paper presented at the Natural Organisms, Artificial Organisms, and Their Brains, Bielefeld, Germany.
- Piaget, J. (1952). *The Origins of Intelligence in the Child*. Norton.
- Poggio, T., Torre, V., & Koch, C. (1985). Computational Vision and Regularization Theory. *Nature*, *317*, 314-319.
- Posner, M. (1980). Orienting of attention. *Quarterly Journal of Experimental Psychology*(32), 2-25.
- Purves, D., & Lichtman, J. W. (1980). Elimination of Synapses in the Developing Nervous System. *Science*(210), 153-157.
- Quartz, S. R., & Sejnowski, T. J. (1997). The neural basis of cognitive development: A constructivist manifesto. *Behavioral and Brain Sciences*(20), 537-596.
- Rack, P. M. H., & Westbury, D. R. (1969). The effects of length and stimulus rate on tension in the isometric cat soleus muscle. *Journal of Physiology*, *204*, 443-460.

- Rucci, M., Wray, J., Tononi, G., & Edelman, G. M. (1997, April, 20-25). *A robotic system emulating the adaptive orienting behavior of the barn owl*. Paper presented at the IEEE International Conference on Robotics and Automation, Albuquerque, USA.
- Rueckl, J. (1993). *Jumpnet: a multiple-memory connectionist architecture*. Paper presented at the 15th Annual Conference of the Cognitive Science Society.
- Sandini, G. (1997, April). *Artificial Systems and Neuroscience*. Paper presented at the Proc. of the Otto and Martha Fischbeck Seminar on Active Vision.
- Sandini, G., Alaerts, A., Dierickx, B., Ferrari, F., Hermans, L., Mannucci, A., Parmentier, B., Questa, P., Meynants, G., & Sheffer, D. (1998, May). *The Project SVAVISA: a Space-Variant Color CMOS Sensor*. Paper presented at the AFPAEC'98, Zurich.
- Sandini, G., Braccini, C., Gambardella, G., & Tagliasco, V. (1981). *Computational Model of the Retino-Cortical Transformation of the Peripheral Visual Field*. Paper presented at the 4th European Conference on Visual Perception, Gouvieux (France).
- Sandini, G., Metta, G., & Konczak, J. (1997, November). *Human Sensori-motor Development and Artificial Systems*. Paper presented at the AIR&IHAS '97.
- Sandini, G., & Tagliasco, V. (1980). An Anthropomorphic Retina-like Structure for Scene Analysis. *CVGIP*, 14(3), 365-372.
- Sauvan, X. M. (1998). Early Integration of Retinal and Extra-Retinal Information: Recent Results and Hypothesis. *Reviews in the Neurosciences*, 9, 291-299.
- Schaal, S., & Atkeson, C. G. (1998). Constructive Incremental Learning from Only Local Information. *Neural Computation*(10), 2047-2084.
- Schwartz, E. L. (1977). Spatial Mapping in the Primate Sensory Projection: Analytic Structure and Relevance to Perception. *Biological Cybernetics*, 25, 181-194.
- Schwarz, U., Busettini, C., & Miles, F. A. (1989). Ocular Responses to Linear Motion are Inversely Proportional to Viewing Distance. *Science*, 245, 1394-1396.
- Scott, D. W. (1992). *Multivariate density estimation*. New York: Wiley.
- Snyder, L. H., & King, W. M. (1992). Effect of viewing distance and location of the axis of head rotation on the monkey's vestibuloocular reflex. I. Eye movement responses. *Journal of Neurophysiology*, 67, 861-874.

- Specht, D. F. (1990). Probabilistic Neural Networks. *Neural Networks*(3), 109-118.
- Stein, B. E., & Meredith, M. A. (1993). *The merging of the senses*: MIT-Press.
- Streri, A. (1993). *Seeing, Reaching, Touching - The Relations between Vision and Touch in Infancy*. MIT Press.
- Sutton, R. S., & Barto, A. (1998). *Reinforcement Learning: an Introduction*. Cambridge: MIT Press.
- Telford, L., Seidman, S. H., & Paige, G. D. (1998). Canal-otolith interactions in the squirrel monkey vestibulo-ocular reflex and the influence of fixation distance. *Experimental Brain Research*(118), 115-125.
- Thelen, E., Corbetta, D., Kamm, K., JP, J. P. S., Schneider, K., & Zernicke, R. F. (1993). The transition to reaching: mapping intention and intrinsic dynamics. *Child Development*, 64(1058-1098).
- Trevarthen, C. (1980). *Neurological development and the growth of psychological functions*. Paper presented at the J. Sants (Ed.) *Developmental Psychology and Society*, MacMillans, London.
- Trevarthen, C. (1984). *How control of movement develops*. Paper presented at the H.T.A. Whiting (Ed.), *Human motor actions: Bernstein reassessed*, Amsterdam, North-Holland.
- Vaidyanathan, R., Chiel, H. J., & Quinn, R. D. (1997, May, 12-14). *A hydrostatic robot for marine applications*. Paper presented at the 11th VPI & SU Symposium on Structural Dynamics and Control, Blacksburg.
- Van Essen, D. C., Anderson, C. H., & Olshausen, B. A. (1992). Information processing in the primate visual system: An integrated systems perspective. *Science*(255), 419-423.
- Van Essen, D. C., & Deyoe, E. A. (1995). Concurrent processing in the primate visual cortex. In M. S. Gazzaniga (Ed.), *The cognitive neurosciences* (3rd edition, 1996 ed., pp. 383-400). Cambridge, Massachusetts: Bradford Book.
- Van Hopstal, A. J., & Kappen, H. (1993). A Two-Dimensional Ensemble Coding Model for Spatial-Temporal Transformation of Saccades in Monkey Superior Colliculus. *Network*(4), 19-38.
- Vapnik, V. N. (1998). *Statistical Learning Theory*. New York: Wiley.
- Viirre, E., Tweed, D., Milner, K., & Vilis, T. (1986). Re-examination of the gain of the vestibulo-ocular reflex. *Journal of Neurophysiology*(56), 439-450.

- Von Hofsten, C. (1979). Development of visually directed reaching: the approach phase. *Journal of Human Movement Studies*, 5, 160-178.
- Von Hofsten, C. (1991). Structuring of early reaching movements: a longitudinal study. *Journal of Motor Behavior*, 23(4), 280-292.
- Von Hofsten, C., & Rosander, K. (1997). Development of Smooth Pursuit Tracking in Young Infants. *Vision Research*, 37(13), 1799-1810.
- White, B. L., Castle, P., & Held, R. (1964). Observations on the development of visually guided reaching. *Child Development*, 35, 349-364.
- Williamson, M. M. (1996, 1996). *Postural primitives: interactive behavior for a humanoid robot arm*. Paper presented at the From Animals to Animats: 4th International Conference on Simulation of Adaptive Behavior, Cambridge, Massachusetts USA.
- Wilson, V. J., & Jones, G. M. (1979). *Mammalian vestibular physiology*. Plenum Press.
- Wurtz, R. H., & Munoz, D. P. (1996). Role of Monkey Superior Colliculus in Control of Saccades and Fixation. In M. S. Gazzaniga (Ed.), *The Cognitive Neurosciences* (3rd printing ed., pp. 533-548). Cambridge, Massachusetts: MIT Press.

Table of illustrations

Box 1 The experimental setup. The experimental setup consists of a five degrees of freedom robot head (designed and realized at LIRA-Lab), and an off-the-shelf six degrees of freedom robot manipulator (an Unimation Puma260), both mounted on a rotating base: i.e. the torso. The kinematics resembles that of the upper part of the human body although with less degrees of freedom. From the sensory point of view, the Babybot is equipped with two space-variant cameras (Sandini & Tagliasco, 1980), (Sandini, Braccini, Gambardella, & Tagliasco, 1981), microphones for acoustic localization, an inertial sensor simulating the vestibular system (Panerai & Sandini, 1998), and proprioceptive information through motor encoders. The robot is controlled by a set of PCs – ranging from Pentium II to Pentium III processors – each running Windows NT and connected by a fast Ethernet link. In order to provide the necessary interface with the hardware (i.e. sensors and motors) some machines are equipped with motion control boards, frame grabbers, AD converters, etc. In particular, one machine controls the robot arm and the torso, another one the head, and a third computer performs the visual processing. The software adheres to DCOM, a standard, which allows running objects among the various machines. The Babybot kinematics is shown on the right panel of the picture below. The dashed lines indicate joint's axes numbered from q_1 to q_{12} respectively. 13

Box 2 Konczak and coworkers (Konczak, Borutta, Topka, & Dichgans, 1995) followed nine young infants longitudinally from 4 to 15 months of age. They analyzed arm kinematics and dynamics in order to determine which learning procedure might underlay the acquisition of goal-directed reaching. The main results of this study can be summarized as follows: i) the amplitude of joint torques do not vary systematically with age – thus early reaching kinematics is not conditioned by the inability to generate appropriate torques; ii) external forces exploitation emerges only at about 9 months of age, that is an important component of proper limb control is acquired with experience; iii) there is a clear trend in the evolution of torque timing, which might reflect yet another kind of learning process – important for proper trajectory generation; iv) it seems conceivable

- that learning is “unsupervised”, which might also reflect a sort of optimization based learning. The latter point is a crucial one; in fact, the economy of the movement could be a suitable “cost function”, whose minimization would lead to the observed hand kinematics in adults. As an example, the picture above shows the progression towards a stable kinematic pattern and the straightening of the trajectory. Trajectories are projected onto a vertical plane: the starting point is on the bottom left corner of the image, and the stationary target is toward the upper right corner – see bottom row (adapted from (Konczak & Dichgans, 1997))..... 28
- Box 3 Learning the closed loop Jacobian matrix as described in section 3.1. The relevant components of the two matrices controlling the eyes are plotted with respect to time. The point here is that the learning process is convergent. In general, considering a discrete-time case:.. 58
- Box 4 Emergence of goal directed eye movements. We recorded several eye movement trials, with the robot working unrestrained – i.e. the task was simply to foveate some visually identified target – targets appeared randomly within the robot’s field of view. The results from the first stage are plotted below, where abscissa and ordinates represent the image plane, and different graphical signs mark trajectories (the target position at each control step – 40ms period). As expected all the trajectories are converging to the fovea. This plot was obtained after the first stage of the development process as described in section 3.1. It is worth noting that in this case the movements are still quite slow, and the number of “points per trajectory” is high..... 59
- Box 5 An exemplar trajectory after learning. Note as the first three steps are enough to reduce the retinal error to less than five pixels, afterwards the target remains in the fovea. This plot has been obtained after learning of the closed loop controller was completed and the saccade maps almost converged to a stable configuration. The two sets of points are relative to the left and right eye. In this case, it is clear that, the target appeared on the left side of the robot. 62
- Box 6 Animal vestibulo-ocular reflex. In animals with fixed eyes, like many insects and some birds, compensatory head or body movements produce retinal image stabilization. Primates and many vertebrates with an efficient oculo-motor apparatus, rely mostly on compensatory eye-movements. As a matter of fact, the “hardware” triggering compensatory motor responses is common to many biological species. A wide range of mechano-neural transducers, functionally equivalent

to rotation- and translation-sensitive mechanisms, are found in many species (Wilson & Jones, 1979). One can speculate about the advantages of these particular motion sensing “transducers”, but nevertheless it remains that such a particular design solution has been naturally selected to deal with the image stabilization problem. In primates, the mechanism controlling the direction of gaze on the basis of inertial information is called Vestibulo-Ocular Reflex (VOR). It is subdivided into angular VOR (AVOR) – generating oculo-motor responses to angular head motion – and translational VOR (TVOR) – generating responses to linear head motion (Paige, 1991), (Schwarz, Busetini, & Miles, 1989). In the case of the AVOR, three ring-shaped sensors (called semi-circular canals) sense angular velocities along three perpendicular directions. In the case of the LVOR, the sensing is performed by the otoliths organs, which sense linear movements in horizontal and vertical directions, and orientation of the head with respect to gravity (Kandel et al., 1991). The vestibular reflexes are known to operate in open-loop, are very rapid and work best for high frequency movements of the head (Keller, 1978), (Benson, Guedry, & Melvill Jones, 1970), (Wilson & Jones, 1979). On the other hand, the visual reflexes, like the Opto-Kinetic Reflex (OKR), operate in closed loop, they are slower and respond better for lower frequencies of head movements (Baarsma & Collewijn, 1974), (Micheal & Jones, 1966). The human sensory apparatus is shown below..... 79

Box 7 Learning the vestibulo-ocular compensation. The plots below show how the compensatory eye commands are acquired. The robot is stimulated manually by random rotation of the torso. This generates both inertial and visual information. The neural network combines the two sources of information, and a compensatory command is generated in order to minimize the optic flow. The first plot shows the inertial sensor signal and the generated motor command (eye velocity) just at the beginning of training – both signals are normalized in the range ± 1 ; abscissa is time expressed as control cycles (40ms). The motor command in this first stage steadily increases because compensation is not yet effective. On a second plot, the same quantities are displayed after some training has taken place. In this case, compensatory motor response is still growing, but at a slower rate..... 87

Box 8 Learning the vestibulo-ocular compensation: the VOR map. The graph below shows the VOR network output after about 10000 learning steps. Note that because of the dependence on the optic flow,

the map allows taking into account the visual input appropriately. In practice, the system generates a higher (amplitude) motor command in situations where the optic flow is higher – i.e. when the retinal slip is high, it is reasonable to use a higher stabilization command. There is also a dependence on the inertial signal as expected..... 88

Box 9 Learning the vestibulo-ocular compensation: minimizing the optic flow. In order to evaluate the learning process, we analyzed the ratio between the inertial information (i.e. the stimulus) and the optic flow (the stabilization performance). The rationale is that if the network is behaving correctly the optic flow should, on average, decrease as learning progresses. The analysis consisted of plotting the optic flow versus the inertial signal. This operation was repeated for 300 sample-long portions of data, extracted at different consecutive instants of time over the training period. The slope of the linear fitting of those data gives an indication of the ratio between stimuli and stabilization. The two pictures below show this analysis graphically for an exemplar set of 300 points (top), and for different portions of data (each of them 300 points long). As learning progresses the slope decreases showing that the optic flow is effectively minimized. 89

Box 10 The Asymmetric Tonic Neck Reflex. The stimulus for the ATNR is the turning motion of the head. This head turn triggers a complex bilateral synergy. The infant's arm is extended to the side the infant is looking, effectively bringing the hand into the field of view. The contralateral arm is flexed as part of crossed extensor reflex spanning both homologous limbs. Thus, this multi-muscle synergy, coupling arm and head movements, provides an effective mean of linking visual and proprioceptive maps. The picture below shows the ATNR in two 3-month-old twins. Note the typical "fencer" position with one arm flexed and the other arm extended. The ATNR can be elicited up to the 4th postnatal month. 95

Box 11 Bizzi and colleagues' experiment. By electrically stimulating a few sites in the frog's spinal cord, Bizzi and coworkers showed that the limb movement could be coded in terms of "force fields" and an Equilibrium Point (EP). The experiment consisted in exposing the spinal cord (as shown in figure E below) in order to implant an electrode; the frog's limb was connected to a force sensor. The limb was moved passively to different positions within its workspace and the forces with and without stimulation were recorded (see panel A and B). The position of the electrode was kept constant for an entire

set of measurement. Force samples were interpolated, as shown in panel C, in order to get a continuous approximation of the underlying force field (the EP was identified as the point where the field is equal to zero). It was surprising that among all possible fields, many of the measured ones were of convergent type with a single EP. By stimulating different sites, the experimenters identified about four different fields. It has been shown also that the vast majority of limb movements can be coded by just combining these fields linearly. This is even more surprising given the non-linearity of actuators and neural responses. This led to the hypothesis that limb placement can be obtained by moving the EP, this is turn might be obtained by a linear combination of the four identified fields (which are termed “basis fields”). Panels D and E show the position of a force field with respect to the limb workspace. Adapted from (Mussa-Ivaldi et al., 1993)..... 100

Box 12 Reaching trajectories. This experiment was performed to illustrate the performance of the proposed approach. It describes the learning of ballistic reaching movements toward static visual targets. In order to test the performance of the system at different learning stages, the position in the arm’s workspace of three targets was calibrated beforehand by manually positioning the end-effector at target center and storing the corresponding joint angle values measured by the encoders. Each target consisted of a piece of cardboard about $5 \times 5 \text{ cm}$ in size. During the training the target of the reaching task was manually moved by the experimenter over the arm’s workspace while the reaching behavior was continuously activated. From time to time training was suspended and performance evaluated. During the evaluation phase, the three targets in the calibrated positions were activated one at a time and the trajectory of the arm stored. The reaching error was measured by computing the Euclidean distance between the pre-calibrated target positions and the position of the end-effector at the end of the reaching movement. At least 30 trials (10 for each target) were executed and the average error and standard deviation were computed. During this evaluation phase the map update was stopped and the noise term removed. The reaching error before and after 51 trials, and after 134 trials are reported in the following table: 105

Box 13 Reaching lookup table. The figure below shows one component of the reaching lookup table after more than 1000 reaching trials. In order to display it (see sketch at the bottom of this box), the following

procedure has been applied: i) the input domain has been divided into a regular grid (for a total of $11 \times 11 \times 12$ cubes); ii) 12 slices of varying “version angle” are plotted as 11×11 2D maps; iii) the color intensity represents the output of the map controlling joint 6 (shoulder); iv) the outputs of all units falling into the same cube have been averaged. The output is the position of the EP in joint space, which is expressed in radians. 111

- Box 14 Reaching trajectories. In the 3D case we recorded the trajectory of the arm end-point. In addition, we recorded the gaze direction, from which it is possible to recover the position of the fixation point in space. The two quantities have been plotted during a reaching movement in order to illustrate quantitatively the behavior of the robot. Moreover, the position of the fixation point provides also an estimate of the target position. In fact, whenever the retinal error is below a small threshold, we can assume that the position of the fixation point is indeed the position of the target. The simple wire-frame model represents the robot. Small circles indicate joints; solid lines are the links. Concerning the fixation point, two different marks can be distinguished: the cross marks represent the time instants when tracking was of smooth pursuit type; whereas, the small squares are instead related to saccadic control..... 112
- Box 15 Reaching trajectories: top view. As in Box 14, two trajectories are shown. Two different marks can be distinguished: the cross marks represent the time instants when tracking was of smooth pursuit type; whereas, the small squares are related to saccadic control. ... 114
- Box 16 Reaching trajectories: two representative trajectories of the robot behavior. As before, two different marks can be distinguished: the crosses mark smooth pursuit control; whereas, the small squares are related to saccadic control. Note that in the topmost picture the end-point and fixation position almost coincide. 115

Figure 1 The pattern of interconnections in the Babybot. This schematic is the block diagram of Babybot’s controller. The yellow blocks are those where learning and adaptation take place. Not all areas are active from the beginning. The blue central block is the robot mechanical plant – i.e. the system dynamics. Green blocks are processing related areas either visual or motor. Orange marked triangles are gain of PD controllers. Many of them are tuned beforehand..... 20

Figure 2 The pattern of interconnections in the macaque (adapted from (Van Essen & Deyoe, 1995)). Van Essen and coworkers pointed out

that visual processing in primates involves dozens of different areas, and both forward and backward connections, with the former, perhaps, carrying out the processing per-se, and the latter type mostly implementing a sort of “flow control” structure – this view is oversimplified anyway. Connections are both hierarchical and concurrent, so processing is both serial and parallel at the same time.

..... 21

Figure 3 The developmental stages. The diagram above approximately shows the interleaving of the developmental stages; abscissa represents time. The first step is the acquisition of the closed loop gains; reflex-like modules control the arm sub-system. After a while, learning of the saccade control begins. Whatever movement of the robot also stimulates the inertial sensor: this information is used to tune the VOR. Eventually the eye-head coordination is acquired together with a more effective head-arm coordination map..... 35

Figure 4 Gazing and reaching. Two trajectories are shown, the fixation point and the arm end-point respectively. The simple wire-frame model represents the robot. Small circles indicate joints; solid lines are the links. Concerning the fixation point, two different marks can be distinguished: the crosses represent the time instants when tracking was of smooth pursuit type, the small squares are related to saccadic control. Note that the arm end-point follows the motion of the fixation point up to the moment when the target is too far away to be reached. 36

Figure 5 Two views of the same trajectory plot shown in Figure 4. The upper panel is the lateral view; the lower plot represents the top view. As before, fixation point and arm trajectories are shown. The fixation point motion is described by either cross marks (when smooth pursuit is active) or square marks (saccadic control). 37

Figure 6 A neural network (RBF) over-fits the data. In this case, a very limited training set was used. The network has more units than training points and consequently cannot properly approximate the data. 43

Figure 7 Results of the performance test of three learning algorithms. Abscissa represents noise (i.e. the probability of taking a “greedy” action versus an explorative one), ordinates the number of successful trials out of 5000 control steps. These results endorse the hypothesis that the solution to the control problem actually lies in a sub-region of the whole control/state space – there are in practice “wrong” regions of the state space, which do not need to be explored. In this case an

algorithm such as the closed-loop or the inverse model based one performed comparatively better than a “random explorer”. In terms of performance the “inverse model” schema is running two orders of magnitude better than the “random explorer”. 46

Figure 8 A cartoon drawing illustrating the effect of constraining the state space. In this case, dimensionality is reduced because the problem itself is inherently two-dimensional. Furthermore, another principle is shown: though the initial formulation is made on \mathcal{R}^3 , the actual problem has its own precise limits – i.e. it is defined on a limited set of points. On top of this, the learner can apply a variable resolution-coding schema, thus maximally exploiting a limited amount of resources. 48

Figure 9 The developmental stages. The diagram above roughly shows the interleaving of the developmental stages; abscissa represents time. The first step is the acquisition of the closed loop gains; reflex-like modules control the arm sub-system. After a while, learning of the saccade control begins. Whatever movement of the robot also stimulates the inertial sensor: this information is used to tune the VOR. Eventually the eye-head coordination is acquired together with a more effective head-arm coordination map. 56

Figure 10 The eye control schema – simplified. It consists of a closed loop and a feed-forward secondary loop. The loop using the inverse Jacobian is derived from a classical visual servoing approach. The secondary loop consists of an inverse model (indicated by “Map”). It is activated whenever necessary – retinal error greater than a threshold – and generates a fast motion of the eyes in order to foveate the target. The goal of the network is to learn the inverse model. λ is a positive constant gain. It is tuned in order to obtain stability of the closed loop system. The input to the robot controller is a velocity command. A low level controller (PID) generates the motors’ driving torques τ . The block identified by “Saccade” is the governing logic (i.e. the threshold mechanism issuing the “start” signal for the fast motion). 61

Figure 11 A hypothetical eye-head coordinated movement. The left panel (A) represents the initial situation preceding a saccade: a target indicated by the big “A” character appears within the robot’s field of view. The middle sketch indicates that even before any actual motion is started the robot computes the final eye positions; this efferent signal is the used to determine the required neck rotation. Once the appropriate commands are computed they are fed into the head low-

level controllers, though because of the different inertias and programmed accelerations the eyes get to the target before the head motion is completed. On the other hand, because of the inertial sensor and the VOR the whole motion remains coordinated and eventually the head/eye system reaches a stable configuration as shown in panel (C)..... 63

Figure 12 The neck control schema. It employs the same working principle of the eye controller. However there are a few important differences. First, there is not direct visual feedback, on the contrary, eye positions drive the movement of the head – the PID controller has to move the head in order to maintain a symmetric vergence configuration as much as possible. Second, the saccade-like movement is based on the prediction of the eye positions at the end of the saccade – i.e. efferent copy. Δq_5 and Δq_4 are the output of the eye maps; they are combined with the actual eye positions to get a prediction of the eyes' orientation, this eventually allows estimating the required head rotation..... 64

Figure 13 Robot motor performances. The upper plot shows the moving window average of the residual retinal error (i.e. at the end of a saccade). The lower plot is the standard deviation of the same 300 samples. Abscissas represent the number of trials. Note also that the error is computed over the space-variant geometry of the retinal layout; consequently they should have been plotted on an exponential scale rather than the linear one to take into account the compression due to the logarithmic sampling..... 74

Figure 14 Two views of the left eye map. The “+” sign represents the most recent 300 samples of the training set, and the circles the position of the unit's centers. The plot has been obtained after about 90000 steps performed using the most recent 300 samples from the training set. The input space (x,y) is the image plane in Cartesian coordinates (bear in mind that the actual data are acquired in the space variant log-polar plane described in section 5.1), the output (the height of the surface plot) is the angle required to foveate a target appearing in the corresponding (x,y) image position..... 77

Figure 15 The head control map. It is obtained after about 30000 control cycles. In this case the output is already the required velocity command (approximately the angle multiplied by the control rate), the input is the predicted position of the two eyes as described in section 3.3. The bottom panel shows a 2D plot of the neural network. The “+” signs are the training samples, and circles stands for the

positions of the units; the solid lines represent the contour lines. Note also that the upper-right quadrant is empty, because it corresponds to divergent-eye configurations. 78

Figure 16 Geometry of the head-eye system showing the parameters relevant for the inertial and visual measures. This is the reference model for the kinematic analysis of the stabilization as described in text. P is the gaze point at distance d from the head rotational axis (i.e. the neck). b is the interocular distance or baseline. 80

Figure 17 The simple method for image stabilization. The angular sensor measures the rotation of the head. Vision, in parallel, senses the retinal slip. The two sources of information are linearly combined (using gain G_{AVOR} and G_{OKR}) and fed to the head control system. The inertial information is processed open loop. 82

Figure 18 Adaptive tuning of compensatory gains. The compensatory gain of the eyes is tuned according to distance of fixation. The adaptive gain module simply replaces the constant one of the previous scheme (Figure 17). 83

Figure 19 Learning compensation commands: VOR gain tuning. In order to generate an appropriate stabilization command, inertial information (from the sensor) and retinal slip (from optic flow) are combined. The neural network is responsible for building such association. The teaching signal is the optic flow itself, which has to be minimized for stabilization to be effective. It is worth stressing that for the schema to work, the robot must be interacting with a real environment; that is the system evaluates on-line performance (the residual optic flow) to update the network's parameters. 84

Figure 20 Position and velocity information during a gaze redirection experiment. From top to bottom: the left eye position, the head position, the inertial sensor output, and the gaze position are shown. The head velocity sensed by the inertial sensor is used to generate compensatory eye movements. The end of the saccadic part of the eye movement is marked with a '*' symbol. 91

Figure 21 Visual parameters computed during gaze redirection. From top to bottom: target retinal error (left eye), image stabilization index (ISI) for the left eye, target retinal error (right eye), and ISI (right eye). The overshoot is measured as the difference between the minimum ('*' symbol) and the maximum retinal error ('+' symbol) after the saccade. The time interval required for the ISI to fall below a stable threshold of 0.3 is delimited by two '*' symbols. 92

Figure 22 Parameters measuring stabilization performance during coordinate eye-head movements. Top: retinal error overshoot: the inertial and non-inertial case. Bottom: time interval required for the ISI to fall below the 0.3 threshold. 93

Figure 23 Controller structure: motor primitives, represented by torque fields are combined (weighted by C_1 , C_2 , C_3 , and C_4). The overall field “guides” the arm end-point toward the EP. 99

Figure 24 The overall arm’s control scheme. The position of the head (q_{head}) queries the map that “computes” the activation vector for the arm. This stage is followed by a trajectory generation that interpolates linearly between activation vectors. The resulting force field is then computed and used to generate the torques, which drive the arm motors. 103

Figure 25 The trajectory showed a bell-shaped velocity profile. On the left, the trajectory of the arm end-point: abscissa and ordinate represent the plane where the arm motion was constrained – bear in mind it was a 2D experiment. The dashed line is the actual trajectory sampled at 40Hz, the solid line is the “virtual trajectory”. In spite of the virtual trajectory that moves directly to the target, the actual motion showed an overshoot. On the right, the hand speed has a bell-shaped profile. Note the two bumps corresponding to the first large “transport phase” and a second corrective movement. Time is expressed in control cycles (25ms), and speed in meters per second. 108

Figure 26 The arm control schema: 3D reaching experiment. This schema roughly resembles Figure 24 apart from the two “yellow” blocks which carry on the “redundancy reduction” and the computation of the arm activation values starting from the equilibrium point position. 110

Figure 27 The log-polar mapping. The upper row (a) shows a log-polar or cortical image as acquired through the Giotto Camera (Sandini et al., 1998) a CMOS realization of the log-polar mapping at the sensor level (this image is 76×128 pixels). The image on the right is the corresponding remapped retinal image. The lower panel (b) shows – albeit simplified – how retinal pixels map onto cortical pixels. For instance, the innermost circle (on the left) map to the leftmost column (on the right) representing the fovea, on the contrary radii map to rows in the cortical image. 120

Figure 28 An example of log-polar mapping, note as radial structures in the flower (petals) map to horizontal structures in the log-polar image. Circles, on the other hand, map to vertical patterns.

- Furthermore, note as the central part of the flower occupies about half of the corresponding log-polar image. 121
- Figure 29 Color processing. The upper row (a) shows a typical image from the robot's point of view: original image (left) and color segmented image (right). All the processing is carried out in the log-polar domain; images are mapped back to the Cartesian space for visualization purposes. The lower row (b) contains the HS histograms: background (left) and object (right). Note that, as the histograms do not overlap, segmentation can be performed reliably. Hue values range from 0 (red) to 360. Saturation ranges from 0 (gray level) to 1 (full color). The V component has not been used for segmentation purposes in order to enhance robustness to changes in illumination. 127
- Figure 30 Two exemplar basis fields represented as torque fields in joint coordinates. Ordinate and abscissa show joint position (Babybot's joint 7 and 8 – shoulder and elbow respectively) in degrees. Arrows point to the common EP of the two joints. The actual resting position of each actuator q_0 was defined beforehand. 133
- Figure 31 Two exemplar basis fields represented in Cartesian coordinates (see also Figure 30). Ordinate and abscissa represent the plane where the arm motion has been constrained – see experimental session on section 3.7. 135
- Figure 32 Exemplar converging force field as employed by Babybot's controller. The 7 vectors together with the knowledge of the basis fields, allow us to recover the activation values, from there a simulation of the spring-like actuators allow computing, for each time step, a corresponding torque vector. The torque vector eventually is converted to the appropriate currents, which drive the motors..... 136
- Figure 33 The Babybot distributed architecture. Presently, it consists of four PCs connected through a 100Mbit/s Ethernet link. Some of them are equipped with frame grabbers, axis control boards, and ADCs. From the software point of view, the adopted framework allows uniform programming and object distribution across machine boundaries. In practice the whole system can be seen as a unique processing machine where internal boundaries are actually as thin as possible..... 139

Index

- 3D, 51, 64, 98, 107, 109, 110, 112
Abend, W., 107
Actuators
 elastic, 102, 131
 motors, 13, 61, 97, 101, 103, 136
 muscle-like, 16
 spring-like, 56, 94, 136
 stiffness, 26, 33, 56, 97, 106, 109, 113, 131
Adolphs, R., 50
Agent, 17, 19, 23, 39, 40, 41, 44, 51, 117
AI, 11, 12, 51, 52, 96
Allman, J.M., 119
Aloimonos, J., 15
Andersen, R.A., 74, 123
Anderson, C.H., 19
Approach phase, 29, 30
Approximation theory, 50
Arm activation, 98, 102, 103, 110
Ashmead, D.H., 27
Atkeson, C.G., 24, 39, 49
Atkinson, J., 23, 26, 51
ATNR, 95, 96, 97, 101
Baarsma, E.A., 79
Babybot, 13, 14, 16, 17, 20, 33, 55, 65, 85, 117, 133, 136, 137, 138, 139
Background, 58, 72, 75, 125, 126, 127
Behavior
 navigation, 17
Bajcsy, R.K., 15
Bakker, B.J., 16, 25
Bakker, P., 16, 25
Ballard, D.H., 15
Bandyopadhyay, A., 15
Banks, M.S., 74
Barron, A.R., 24, 121
Barron, J.L., 24, 121
Barto, A., 39, 44
Bauer, J.A., 27
Beauchemin, S.S., 121
Beer, R.D., 16
Behavior, 14, 16, 17, 18, 19, 22, 29, 31, 32, 33, 34, 39, 40, 42, 44, 52, 54, 58, 63, 64, 74, 75, 90, 96, 105, 106, 108, 110, 112, 113, 115, 118, 124, 130
 adaptive, 14, 16, 19
 adult, 12, 19, 25, 26, 30, 53
 goal-directed, 27, 28, 29, 32, 33, 40, 47, 94, 96
 manipulation, 15, 16, 51, 81
 orienting, 14, 15, 16, 18, 22, 24, 27, 31, 63, 96, 113
 reaching, 14, 15, 18, 22, 26, 28, 29, 30, 31, 32, 33, 36, 45, 70, 94, 96, 97, 98, 105, 106, 107, 109, 110, 111, 112, 113, 114, 115, 118
 saccades, 24, 25, 32, 60, 61, 75, 90
 smooth pursuit, 23, 24, 32, 33, 36, 37, 112, 114, 115
 vergence, 64, 83, 108, 109
Bekoff, A., 26, 41
Bell shaped, 29, 107, 108
Bellman, R.E., 41, 48

- Belton, T., 25
Benaim, M., 128
Benson, A.J., 79
Berthouze, L., 16
Bias, 14, 23, 40, 41, 42, 43, 44, 49, 54
Bias-variance dilemma, 14, 41, 42, 49
Bienenstock, E., 39
Biguer, B., 81
Birth, 17, 23, 24, 25, 26, 29, 30, 32, 40, 41, 53, 55, 85, 97, 113
Bizzi, E., 16, 25, 97, 100, 107, 124, 131
Bobet, P., 15
Bootstrap, 17, 41, 43, 118
Borstm, A., 123
Borutta, M., 28, 30
Bower, T., 27
Bowler, C.J., 16
Braccini, C., 13
Braddick, O., 26
Brain, 11, 12, 15, 19, 22, 23, 34, 40, 41, 47, 49, 51, 53, 74, 117
 areas, 11, 19, 23
 cerebellum, 53
 climbing fiber, 53
 cognitive, 11, 12, 17, 31
 cortex, 23, 24, 25, 40, 50, 119
 modules, 17, 23, 34, 35, 44, 47, 48, 49, 51, 52, 55, 56, 106, 107, 117
 streams, 23, 40, 51, 118
 subcortical, 23, 24, 25, 40, 50
Brain sciences, 11, 15, 34
Brooks, R., 15, 52
Broughton, J.M., 27
Brown, C.M., 15
Busettini, C., 79
Bushnell, E.W., 96
Caldwell, D.G., 16
Calibration, 95, 96
Capurro, C., 15, 16
Car, 19, 22, 132
Carpenter, G., 23, 25, 39
Carpenter, R.H.S., 23, 25, 39
Cartesian, 44, 66, 67, 72, 77, 105, 119, 120, 122, 126, 127, 135
Castle, P., 96
Central nervous system, 107
Cheng, G., 16
Chiel, H.J., 15
Cioni, G., 32
Clarkson, M.G., 27
Clifton, R.K., 27
Closed loop, 32, 35, 45, 56, 58, 59, 60, 61, 62, 64, 76, 79, 113
Closed-loop, 45, 46, 47, 57, 75
Collewijn, M., 79, 82
Compliant, 97
Connectionist, 47
Connections, 19, 21, 51, 129, 130, 133
Constraints, 64, 108, 123
Control diagram, 18, 73
Control structure, 12, 19, 26, 31, 34, 52, 55, 113, 117, 138
Cook, M., 29
Coordination, 23, 25, 35, 56, 63
Cortical take-over, 40, 60
Cost function, 28, 107
Cowey, A., 119
Crane, B.T., 81
Cross-validation, 50, 131
Crowell, J.A., 74
Crowley, J.L., 15, 16
Cue, 16, 29, 82
Curse of dimensionality, 14, 41, 48
Damasio, A., 50

- Daniel, M., 119
DCOM, 13, 137, 138
Degrees of freedom, 13, 14, 17, 18, 24, 31, 32, 33, 52, 55, 63, 64, 65, 75, 97, 107, 113, 117, 134
Demer, J.L., 81
Dennet, D.C., 12
Deprivation, 23
Development, 12, 14, 16, 17, 19, 22, 23, 24, 25, 26, 27, 29, 30, 31, 32, 33, 34, 35, 39, 40, 51, 53, 55, 56, 59, 61, 63, 76, 85, 94, 96, 97, 107, 113, 118, 138
 adaptation, 7, 11, 19, 20, 62, 117
 approach, 14, 53, 61
 artificial, 30, 34, 96, 117
 events, 34, 55, 63
 free parameters, 17, 24, 42, 55
 learning, 11, 12, 14, 15, 17, 18, 20, 22, 23, 24, 28, 29, 30, 31, 32, 33, 35, 39, 40, 41, 42, 43, 44, 45, 46, 47, 49, 50, 51, 52, 53, 56, 57, 58, 59, 61, 62, 69, 75, 76, 84, 85, 86, 87, 88, 89, 95, 96, 98, 101, 103, 104, 105, 106, 107, 108, 109, 117, 118, 126, 128, 129, 138
 rules, 12, 17, 97
 stages, 12, 17, 20, 24, 27, 31, 35, 46, 56, 86, 117
Deyoe, E.A., 12, 19, 21
Diamond, A., 30
Dichgans, J., 28, 30
Differential invariants, 122
Direct approach, 62
Distler, C., 32
Distortion error, 126
DNA, 22
Doursat, R., 39
Durbin, R., 128
Dynamics, 20, 28, 52, 85, 106, 107
Edelman, G.M., 16
Efferent copy, 24, 64, 73
Egelhaaf, M., 123
Eibl-Eibesfeld, I., 96
Embodiment, 15, 49, 50
End-effector, 98, 104, 105, 106
End-point, 33, 36, 97, 99, 101, 108, 109, 110, 112, 115, 134, 136
Environment, 11, 12, 15, 17, 19, 22, 23, 31, 32, 33, 39, 49, 50, 51, 53, 54, 84, 97, 104, 121, 138
Equilibrium point, 16, 56, 70, 97, 99, 100, 102, 103, 109, 110, 111, 132, 133, 135, 136
Erkelens, C.J., 82
Evgeniou, T., 49, 50
Exploitation, 17, 28, 30, 41, 44, 52, 54, 55
Exploration, 17, 23, 39, 40, 41, 44, 45, 46, 52, 54, 55, 59, 60, 96, 101, 109, 118
Exploration-exploitation dilemma, 41, 52
External world, 23, 25, 137
Eykern, L.A.V., 26
Favilla, M., 32
Feedback, 19, 24, 26, 40, 45, 56, 61, 64, 73, 125
Feedback-error learning, 40
Finocchio, D.V., 85
Flash, T., 96
Fleet, D.J., 121
fNMRI, 22
Force fields, 97
Fovea, 25, 32, 59, 62, 75, 119, 120

- Frame of reference, 32, 134
French, R.M., 47
Fritzke, B., 24, 126, 129
Fuchs, A.F., 85
Fulstone, A., 26
Furukawa, K., 40
Gambardella, G., 13
Gandolfo, F., 15, 124, 131, 134
Gaze, 9, 14, 15, 16, 22, 24, 26, 27, 29, 32, 33, 52, 79, 80, 81, 82, 90, 91, 92, 94, 98, 103, 104, 108, 109, 110, 112
Gaze shifting hypothesis, 27, 29
Gdowski, G.T., 74
Geman, S., 39, 42
Ghelarducci, B., 32
Ghez, C., 106
Ghilardi, M., 106
Gidoni, E., 26
Gielen, C.C.A.M., 25
Gilmore, R.O., 23, 24, 63
Giszter, S.F., 16, 97, 131, 134, 135
Globally unique identifier, 137
Gomi, H., 16, 107, 109
Goodale, M.A., 23, 51
Goossens, H.H.L.M., 25
Gordon, J., 106
Gould, J.L., 96
Gravity, 30, 79, 94, 106, 107
Graziano, M.S.A., 123
Greedy action, 46
Grossberg, S., 39
Grosso, E., 15
Growing neural gas, 76, 126, 128
Guedry, F.E., 79
Hadders-Algra, M., 26, 41
Haikawa, Y., 15
Hein, A., 27
Held, R., 27, 96
Hennemann, E., 101
Heuristic, 128
Hine, T., 81
Hirai, K., 15
Hirose, M., 15
Hoffmann, K.P., 32
Hogan, N., 132, 134
Horn, B.K.P., 121, 122, 123
HSV coding, 125
Hubel, D.H., 41, 49, 119
Hue-saturation space, 125
Image stabilization, 79, 82, 85, 92, 125
Image stabilization index, 90, 92, 93, 94, 125
Inertial sensor, 13, 16, 18, 32, 35, 55, 56, 63, 65, 73, 75, 87, 91
Infant reaching, 94
Information, 11, 13, 16, 24, 25, 27, 32, 33, 35, 39, 47, 49, 51, 52, 54, 55, 56, 65, 74, 79, 82, 83, 84, 85, 87, 89, 90, 91, 94, 98, 106, 113, 123, 125
Innate, 12, 117, 135
Integrated circuit, 137
Interface description language, 137
Inverse modeling, 40, 60
Jacobian, 58, 61, 124, 134
Jerk, 107
Jessel, T.M., 53
Johnson, M.H., 23, 24, 63
Jones, G.M., 79
Jordan, M.I., 16, 96, 107
Kaas, J.H., 119
Kalveram, K.T., 95
Kandel, E.R., 53, 79, 101, 131
Kappen, H., 25
Kawano, K., 25
Kawato, M., 16, 40, 61, 107, 109

- Keller, E.L., 79
Kicking, 96
Kinematics, 13, 24, 28, 29, 64, 65,
80, 81, 84, 98, 104, 109, 134
King, W.M., 81
Koch, C., 39
Koenderink, J., 122
Konczak, J., 16, 28, 29, 30
Korbmacher, H., 32
Korte, R., 32
Kuniyoshi, Y., 16
Kuperstein, M., 95, 96
La Noce, A., 32
Laboratory for integrated
advanced robotics, 16
Learning stages, 12, 75, 105
Leary, D.D.M.O., 27
Least-square approach, 57
Ledoux, J.E., 50
Lichtman, J.W., 53
Linear summation hypothesis, 83
Ljung, L., 42
Log-polar, 13, 55, 65, 66, 71, 72,
74, 77, 90, 119, 120, 121, 122,
123, 124, 125, 126, 127
Loops, 16, 32, 35, 40, 45, 46, 47,
55, 56, 57, 58, 59, 60, 61, 62,
64, 74, 113, 125, 138
Lorentz, G.G., 50
Low stiffness controller, 33
Low-stiffness, 107
Majarrad, M., 16
Manifold, 39
Manipulation workspace, 81
Manzotti, R., 15
Map, 18, 22, 33, 35, 45, 56, 61,
66, 67, 68, 69, 70, 71, 73, 75,
77, 78, 88, 98, 101, 102, 103,
104, 105, 110, 111, 120, 121,
128
Maringelli, F., 51
Marr, D., 51
Mathew, A., 29
McCarthy, J., 51
McCrea, R.A., 25, 74
Medendorp, W.P., 25
Medrano-Cerda, G.A., 16
Melvill Jones, G., 79
Mendell, L.M., 101
Meredith, M.A., 16, 26, 40
Mesrabi, M., 15
Metta, G., 15, 16
Micheal, J.A., 79
Milani-Comparetti, A., 26
Miles, F.A., 25, 79
Milner, D.A., 23, 51, 81
Milner, K., 23, 51, 81
Minimum variance, 128
Model, 11, 16, 18, 25, 36, 40, 42,
45, 46, 47, 50, 51, 55, 61, 73,
80, 83, 94, 96, 97, 101, 107,
112, 113, 119, 122, 123, 124,
126, 128, 131, 132, 137
Moore, M.K., 27
Morasso, P., 73, 107, 126, 127,
128
Motor act, 23, 26, 30, 31
Motor coordinates, 57, 108
Motor primitives, 96, 97, 98, 99,
101, 132, 134
Motor strategies, 31
Motor-motor coordination, 98,
101, 104
Muir, D.W., 27
Munoz, D.P., 26
Mussa-Ivaldi, F.A., 16, 97, 100,
131, 132, 134, 135
Myelin, 25, 53
Networks
learning rule, 86

- Neural networks, 15, 24
 controller, 26
 Training data, 23, 33, 41, 42, 44
- Neuroscience, 12, 25
- Newborn, 12, 19, 23, 24, 26, 27, 29, 30, 31, 34, 39, 40, 94, 95, 96
- Networks
 hebbian learning, 126, 128
- Nieuwendijk, A.W.J.K., 26
- Noise, 17, 25, 32, 34, 39, 40, 46, 47, 52, 53, 54, 55, 56, 59, 75, 103, 104, 105, 113
- Noisy, 52
- NOT, 24
- Nucleus of optic tract, 24
- Oddera, A., 15
- Olshausen, B.A., 19
- Open-loop, 79, 82
- Operative system, 138
- Optical encoders, 18, 32, 56
- Optical flow, 18, 83, 85, 97, 121
- Optican, L.M., 25
- Opto-kinetic nistagmus, 24
- Otoliths, 79
- Overshoot, 73, 90, 92, 93, 94, 106, 108
- Paige, G.D., 79, 81
- Panerai, F., 13, 15, 16, 25, 55, 73, 80
- Pathways, 19, 23, 51, 119
- PC, 13, 52, 138, 139
- PET scan, 22
- Pfeifer, R., 15, 16
- Photoreceptors, 55, 119
- Physiology, 94
- Piaget, J., 27
- PID, 20, 56, 61, 64, 75
- Plant, 20, 26, 51, 52, 53, 54, 62, 94, 98, 117
- Poggio, T., 39, 49
- Pontil, M., 49
- Posner, M., 51
- Prablanc, C., 81
- Precht, H.F.R., 26
- Preston, K.L., 85
- Principal Component Analysis, 109
- Process, 11, 12, 17, 19, 22, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34, 39, 40, 51, 53, 55, 58, 59, 60, 61, 73, 76, 85, 86, 89, 94, 96, 98, 104, 106, 118, 138
- Propioceptive, 13, 65
- Pruning, 26, 53, 118
- Purkinje, 53
- Purves, D., 53
- Quartz, S.R., 24, 27, 43
- Quinn, R.D., 15
- Rack, P.M.H., 131
- Radial basis functions, 50
- Range information, 83
- Receptive fields, 123
- Redundancy, 14, 24, 32, 33, 64, 75, 110, 113, 134
- Reflexes, 12, 17, 26, 27, 29, 30, 31, 32, 33, 34, 35, 41, 43, 55, 56, 79, 85, 95, 97, 113, 117, 124, 135
 grasping, 31, 97
 tonic neck, 29, 31, 95, 96, 97, 101
- Regularization, 50, 121
- Residual optical flow, 83, 85
- Retinal error, 57, 60, 61, 62, 74, 75, 76, 90, 92, 93, 110, 112
- RGB coding, 125
- Ritzmann, R.E., 16
- Robotics, 11, 12, 13, 14, 15, 16, 17, 18, 20, 22, 31, 32, 33, 34,

- 35, 36, 52, 55, 56, 59, 60, 61, 62, 63, 65, 72, 73, 74, 75, 81, 82, 84, 87, 90, 97, 98, 101, 103, 107, 109, 110, 112, 113, 115, 117, 118, 125, 126, 127, 136, 137
 arm, 13, 18, 56, 65, 102
 head, 15
 Rosander, K., 24, 32
 Rucci, M., 16
 Rueckl, J., 47
 Sandini, G., 13, 15, 16, 25, 55, 73, 120, 124
 Sanguineti, V., 73, 126, 127, 128
 Sauvan, X.M., 25
 Schaal, S., 24, 39, 49
 Scheier, C., 15, 16
 Schwartz, E.L., 53, 119
 Schwartz, J.H., 53, 119
 Schwarz, U., 79
 Scott, D.W., 41
 Seidman, S.H., 81
 Sejnowski, T.J., 24, 27, 43
 Semi-circular canals, 79
 Sensori-motor coordination, 14, 16, 22, 23, 25, 27, 30, 31, 32, 33, 35, 56, 74, 96, 97, 98, 113
 Sensory modality, 17
 Shahinpoor, M., 16
 Shenoy, K.V., 74
 Shunck, B.G., 121
 Snowden, R.J., 123
 Snyder, L.H., 81
 SoftMax, 126, 127, 128
 Spinal cord, 100
 Stability-plasticity dilemma, 47
 State, 12, 17, 18, 25, 34, 39, 40, 41, 44, 45, 46, 48, 49, 51, 52, 54, 55, 62, 73, 96, 108, 118, 132
 initial, 12, 17, 54, 55
 space, 17, 25, 34, 39, 40, 41, 45, 46, 48, 49, 51, 52, 54, 55, 62, 118, 132
 variables, 18
 Stein, B.E., 16, 26, 40
 Steinman, R.M., 82
 Stiffness, 26, 33, 56, 97, 106, 109, 113, 131
 Stimulus, 18, 19, 24, 33, 39, 41, 50, 52, 89, 95, 96, 103, 124
 Stimulus bound, 18, 19, 41
 Streri, A., 40
 Structural Risk Minimization, 49, 50
 Summers, T.R., 26
 Superior Colliculus, 24
 Sutton, R.S., 39, 44
 Suzuki, R., 40
 Systems, 14, 15, 16, 17, 19, 23, 29, 30, 31, 32, 34, 41, 43, 44, 45, 51, 52, 53, 56, 101, 117, 118, 126, 138
 artificial, 15, 34, 41, 51, 60, 63, 94, 117
 binocular, 80, 82
 biological, 14, 15, 17, 32, 34, 41, 43, 44, 53, 54, 56, 85, 96, 117
 control, 12, 13, 14, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 28, 31, 32, 33, 34, 35, 36, 37, 39, 40, 43, 44, 45, 46, 47, 51, 52, 53, 55, 56, 57, 58, 59, 60, 61, 62, 64, 68, 73, 74, 75, 78, 81, 82, 85, 87, 90, 94, 96, 97, 98, 101, 102, 103, 105, 107, 108, 109, 110, 112, 113, 114, 115, 117, 125, 131, 132, 133, 134, 136, 138, 139

- coordinate, 24, 25, 32, 51, 57, 81, 93
- coordinates, 57, 63
- identification, 17, 39, 40, 52, 53, 54, 55, 94, 137
- pulse response, 52
- sensory, 16, 18, 73
- stability, 57
- vestibular, 13, 16, 18, 25, 41, 55, 65, 74, 75, 79, 85
- Tagliasco, V., 13, 55
- Takenaka, T., 15
- Telford, L., 81
- Thelen, E., 29
- Theoretical pressures, 14, 39, 41
- Thorn, F., 81
- Tilt, 18, 60, 75, 98, 108, 109
- Time, 20, 23, 28, 40, 107, 117, 138
 - delay, 19, 32, 53
- Tiso, R., 15
- Tistarelli, M., 15
- Tomasini, L., 128
- Tononi, G., 16
- Topka, H., 28
- Torque, 15, 28, 97, 99, 101, 102, 103, 106, 107, 131, 132, 133, 134, 136
- Torque fields, 99, 102, 131, 132, 133, 134
- Torre, V., 39
- Trajectories, 18, 27, 28, 29, 30, 31, 33, 36, 37, 59, 62, 75, 90, 97, 98, 102, 103, 105, 106, 107, 108, 110, 112, 114, 115, 132
- Tranel, D., 50
- Transport phase, 29, 30, 108, 138
- Trevarthen, C., 27, 96
- Tweed, D., 81
- Vaidyanathan, R., 15
- Van Doorn, J., 122
- Van Essen, D.C., 12, 19, 21
- Van Gisbergen, J.A.M., 25
- Van Opstal, A.J., 25
- Vapnik, V.N., 39, 42, 43, 49
- VC dimension, 49, 50
- Version, 70, 83, 109, 111, 130, 137
- Vestibular, 18, 55
 - information, 25
- Vestibulo-ocular reflex, 25, 26, 32, 33, 35, 55, 56, 63, 69, 74, 79, 84, 85, 88, 90, 113
- Viirre, E.S., 81
- Vilis, T., 81
- Vision
 - algorithms, 15, 58, 69, 83, 84, 85, 86, 87, 88, 89, 90, 121, 122, 123, 125, 126
 - color segmentation, 125
 - cortex, 119
 - processing, 13, 19, 21, 24, 55, 138
- Vital-Durand, F., 32
- Von Hofsten, C., 24, 29, 32
- VOR, 25, 26, 32, 33, 35, 55, 56, 63, 69, 74, 79, 84, 85, 88, 90, 113
- Weiss, I., 15
- Westbury, D.R., 131
- What and where, 23
- White, B.L., 96
- Whitteridge, D., 119
- Wiesel, T.N., 41, 49, 119
- Williamson, M.M., 16
- Wilson, V.J., 79
- Windows NT, 13
- Wiring, 19, 22
- Wray, J., 16
- Wurtz, R.H., 26

Printed September 2000
at Litoprint, Genova - Italy

