Universidade Técnica de Lisboa
Instituto Superior Técnico

A Developmental Roadmap for Learning by Imitation in Robots

Manuel Fernando Cabido Peres Lopes
(Licenciado)

Dissertação para a obtenção do Grau de Doutor em Engenharia Electrotécnica e de Computadores

Orientador: Doutor José Alberto Rosado dos Santos Victor

Júri:
Presidente: Reitor da Universidade Técnica de Lisboa
Vogais: Doutor Alexandre Lemos de Castro Caldas
Doutor José Alberto Rosado dos Santos Victor
Doutor Yiannis Demiris
Doutora Estela Guerreiro da Silva Bicho Erlhagen
Doutor Mário Alexandre Teles de Figueiredo
Doutor Alexandre José Malheiro Bernardino

Maio de 2006

Contents

Abstract

There is a growing research effort for designing "robot companions" able to interact with people in a friendly way, over extended periods of time. This is a challenging endeavor for two main reasons: (i) the need to work in highly unpredictable and uncertain environments, designed for people instead of robots and (ii) the variety of the tasks to execute and the interaction with (non-technical) people over long periods of time. These problems are addressed in this thesis using the paradigms of imitation and artificial development.

With the metaphor of imitation, one could "program" a humanoid-type robot to solve a certain task simply by demonstration and have the robot imitating the task later on, avoiding the user having to write sophisticated computer programs. Although imitation is a learning mechanism massively used by human infants, it is not an easy problem. Inspired after developmental psychology, we present a developmental pathway for the robot to progressively acquire the skills necessary to imitate observed actions: (i) explore its own sensory-motor capabilities, (ii) understand its surrounding environment, (iii) become aware of people acting in the environment.

The first developmental level is devoted to sensory-motor coordination, during which the robot learns how to control its own body. Sensory-motor maps allow the robot to predict the sensory consequences of a certain action (forward model) as well as to determine the motor action necessary to produce a given effect in the world (inverse model). We present a variety of sensory-motor maps that are learned during periods of auto-observation.

In the second developmental level, the robot is attracted toward objects. It learns how to grasp them and explore their properties, based on the sensory-motor maps learned previously. In addition, the robot learns how to recognize grasp actions performed by others, an approach based on the recent findings of the mirror neurons in the pre-motor cortex of macaque monkeys.

In the last stage of development, the robot engages (to some extent) in social interaction. Its attention is drawn toward people and the robot learns how to imitate the tasks performed by a demonstrator. For that purpose, it solves the view-point transformation problem to account for the different coordinate frames of the demonstrator and its own body, chooses among different metrics giving different imitation behaviors and solves the body correspondence problem.

Experiments have been conducted with Baltazar, an anthropomorphic robot combining a binocular head and an articulated arm and hand.

Keywords: imitation learning, artificial development, humanoid robotics, human-robot interaction, vision, machine learning

Resumo

Existe um grande esfoço de investigação para a criação de "robots companheiros" capazes de interagir amigavelmente com as pessoas, durante longos períodos de tempo. Este objectivo é bastante ambicioso por duas razões: (i) a necessidade de trabalhar em ambientes bastante complexos e imprevisíveis, desenhados para pessoas e não para robots e (ii) a variedade de tarefas a executar e a complexidade da interacção com pessoas de formação não técnica. Estes problemas são abordados nesta tese usando os paradigmas de aprendizagem por imitação e de desenvolvimento artificial. Com a metáfora de imitação, é possível "programar" robots para resolver uma tarefa simplesmente demonstrando a solução. No final o robot imitará essa tarefa sem ter havido necessidade de escrever programas de computador sofisticados. Apesar da imitação ser um mecanismo de aprendizagem muito usado pelas crianças, é bastante complicado do ponto de vista computacional. Tendo como inspiração resultados de psicologia do desenvolvimento, apresentamos um roteiro de desenvolvimento para que um robot adquire progressivamente capacidades necessária para imitar acções observadas: (i) explorar as suas capacidades sensorio-motoras, (ii) compreender o mundo circundante e (iii) imitar as acções das pessoas que agem à sua volta.

O primeiro nível de desenvolvimento é dedicado à coordenação sensorio-motora, durante a qual o robot aprende a controlar o seu próprio corpo. Mapas sensoriomotores permitem ao robot predizer a consequências sensoriais de certas acções, como também determinar qual a acção necessária para um dado efeito. Apresentamos uma variedade de mapas que são aprendidos ao longo do tempo através de auto-observação.

No segundo nível de desenvolvimento, o robot é atraído pelos objectos. Aprende com manipular objectos e explorar as suas propriedades, utilizando a capacidades sensorio-motoras desenvolvidas anteriormente. O robot aprende também a reconhecer acções de manipulação realizadas por outros, através de uma abordagem baseada em recentes descobertas de neurofisiologia.

Na fase final de desenvolvimento, o robot inicia uma rudimentar interacção social. A sua atenção é dirigida para as pessoas e o robot imita as tarefas realizadas por demonstradores no ambiente. Com esse objectivo, resolve o problema de transformação de ponto de vista para ter em conta os diferentes sistemas de coordenadas, escolhe de entre várias possíveis métricas de imitação que geram comportamentos de imitação diferentes, finalmente resolve o problema da correspondência entre corpos.

Várias experiências foram realizadas no robot Baltazar, um robot antropomórfico com uma cabeça binocular, um braça e uma mão.

Palavras-chave: aprendizagem por imitação, desenvolvimento artificial, robots humanóides, interacção homem-robot, visão, aprendizagem automática

Agradecimentos

O primeiro agradecimento vai para o meu orientador científico Prof. José Santos-Victor pelo apoio, discussão, cooperação e pelas condições de trabalho e oportunidades proporcionadas no decorrer deste trabalho.

A participação em dois projectos europeus, MIRROR e ROBOTCUB, permitiu-me contactar com diversos investigadores e muitas das ideias desta tese derivaram de discussões com parceiros desse projecto, Claes von Hofsten, Kerstin Rosander, Luciano Fadiga, Giulio Sandini.

Parte deste trabalho foi realizado em cooperação com o INRIA-RENNES. Por esse apoio, oportunidade e colaboração agradeço ao Prof. François Chaumette e ao Nicolas Mansard.

Ao Alex e ao Plínio que foram sempre a primeira linha de ajuda para validar um nova ideia que surgia ou para ajudar a resolver bugs de software. Obrigado.

O desenho e construção do robot onde decorreram a maior parte das experiências não teria sido possível sem a contribuição do Ricardo Beira. Após a sua construção o Rodrigo e posteriormento o Ricardo Nunes resolveram-me todos os problemas causadas pelas fumachas e folgas.

Aos professores Pedro Lima e Luís Custódio onde iniciei as minhas actividades de investigação.

E a todos os colegas, em especial do VISLAB, com os quais trabalhei e discuti ao longo destes anos.

Ao Francisco Melo pela imagem da capa que descreve os principais conceitos deste trabalho.

Este trabalho foi suportado (parcialmente) por:
European Project (IST-2005-004370) RobotCub
European Project (FET-2000-28159) MIRROR
FCT - FSE QCA III SFRH/BD/2961/2000
FCT - Programa Operacional Sociedade de Informação no QCA III
Instituto de Sistemas e Robótica
Instituto Superior Técnico
Escola Superior de Tecnologia de Setúbal

Chapter 1

Introduction

The impressive advance of research and development in robotics and autonomous systems in the past years has led to the development of robotic systems of increasing motor, perceptual and cognitive capabilities. These achievements are opening the way for new application opportunities where these systems will be required to interact with other robots or non technical users during extended periods of time.

“Friendly” and social interaction between robots and humans is a grand challenge for robotics. Due to the diversity of actions/tasks to be performed and the range of possible interactions with objects and humans, it would be impractical (if not impossible) to explicitly pre-program a robot with such capabilities. Instead, such systems must be able to learn by themselves what tasks to execute and how they should be performed, which requires sophisticated motor, perceptual and cognitive skills.

Advances in these areas will allow the creation of new applications like robot companions, living inside people’s homes to provide assistance to daily tasks, care and entertainment. Traditional programming methodologies and robot interfaces are clearly insufficient, as such systems would need to learn to execute complex tasks and improve its performance throughout its lifetime, in an ecological manner.

One powerful method for a non-technical user to “program” a robot to accomplish a certain task is teaching-by-showing. Similarly to the ability of human infants to learn through (extensive) imitation, an artificial system could learn how to perform complex tasks, simply by observing other individuals, humans or robots working in the same area.

Endowing a system with the capability of learning through imitation is an extremely challenging endeavor. In addition to the complexity of the environment where humanoid robots should work, the computational problems in imitation are extremely vast and range from perception and action in a complex world to task inference. To tackle this complexity problem, we will adopt a developmental perspective that can balance the complexity of the system at the various levels of functional performance.

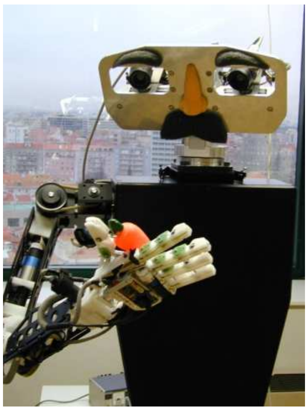

Figure 1.1: Baltazar. A 14 degrees of freedom humanoid torso.

The long-term goal pursued with this thesis is two-fold. On one hand, we want to develop methodologies whereby a system can learn how to perform complex tasks through imitation, in a developmental way, with each module building on top of another. On the other hand, our approach relies on recent findings in neuroscience and developmental psychology, aiming to contribute to a better understanding of fundamental problems of human cognition: how to learn sensory-motor coordination, properties of the objects and the world, how do humans imitate each other, recognize and understand the observed behavior and actions.

1.1 Imitation

Teaching how to solve a given task by showing the solution is a method frequently used by people and animals. It is an easy way to give information. Learning by imitation will most probably be the primary form of teaching social, cognitive robots. A huge variety of tasks can be learned through this process, like learning how to grasp objects, how to dance, cook or hunt, amongst many others. In many ways, the learner can benefit from observing the solution to the problem and imitate it afterwards.

Before discussing any further, it is important to analyze the meaning of *imitation* and how the ability to imitate is an advantage for certain species. According to the Merriam-Webster Online Dictionary [Merriam-Webster, 2005] imitation can be defined as:

- to follow as a pattern, model, or example,
- to be or appear like : resemble
- to produce a copy of : reproduce

- mimic, counterfeit

From these definitions, it is clear that there are multiple definitions of imitation, all involving the observation of actions and the replication of those actions. Before analyzing such different imitation modalities, let us first discuss hypothetical reasons why some animals or agents benefit with the ability to imitate.

In a speculative presentation [Ramachandran, 2000], it is suggested that the ability to imitate could be one major explanation for the "the great leap forward" in human evolution. Although the hominid brain kept the same size from several thousand years, the evolution of language and tool-use took many more years to develop.

The ability to imitate enables a possibility of creating a *theory of mind*, establishing an implicit level of communication between individuals, ultimately leading to language acquisition. Cultural spreading becomes possible by a Lamarckian principle where infants learn how to act by imitating others thus having the same mannerisms as their peers. With imitation, every new discovery will by learned by others very efficiently just by observation and behavior matching.

Imitation can also be a fast solution for a problem that one cannot solve on its own. The copied solution can be a first step, from which a better and more original one can be derived. The most important example concerns learning body movements. In dance classes, the trainer usually performs the gestures while the learner tries to replicate those same gestures. Later on, the dancer can evolve his or her own dance movements and develop his own style. In a more symbolic setting, [Furse, 2001b] shows imitation learning in intellectual tasks, such as learning mathematical algorithms, where teaching is traditionally done by presenting the global solution step-by-step.

Recent research in neuroscience is shedding some light as to the areas of the brain involved in imitative behaviour [Perret et al., 1989]. The so-called Mirror neurons, found in the area F5 of the macaque's premotor cortex [Fadiga et al., 2000] have the intriguing property of firing during both the execution or observation of goal-oriented actions. The ability of the Mirror system to match the behavior of others with one's own behaviours, could allow the recognition of intentions. If we see someone with a rock and a nut, it is possible to infer the intention of breaking and eating the nut.

The observation that *Mirror* neurons fire both during the execution and observation of a specific goal-oriented actions, suggests that the motor system responsible for triggering an action is also involved when recognizing that same action. In other words, recognition is possibly performed in motor terms, rather than in a purely visual space. By establishing a direct connection between gestures performed by a subject and similar gestures performed by others, mirror neurons may be connected to the ability to imitate found in some species [Ramachandran, 2000], establishing an implicit level of communication between individuals.

The Canonical neurons are a second class of visuomotor neurons [Murata et al., 1997] that respond when objects, that afford a *specific* type of grasp, are present in the scene, even if the grasp action is not performed or observed. Thus, canonical neurons may encode object affordances and help distinguishing ambiguous gestures during the process of recognition.

There are several difficulties to overcome before imitative behaviour can be elicited. The following questions need to be solved: (i) how to gather task-relevant information? (ii) how to convert the data that are valid for one agent to the other? and (iii) how to infer the important parts of the demonstration (e.g. infer the goals of the agent).

In the following section we are going to present models of imitation in biological systems, a review of artificial systems using this paradigm and some computational problem arising in imitation.

1.1.1 Models of Imitation

Imitation is not a well defined problem. As an example, let us consider an imitation game, where the demonstrator and the imitator and seated front-to-front, next to a table. When asking someone to imitate a hand movement [Bekkering et al., 2000, Gergely et al., 2002], the results may vary substantially. In the absence of points of reference on the table, people usually are very precise about the movement and the use of the correct hand, with respect to the demonstrator. If, instead, there is a point on the table that is touched by the demonstrator, the imitation is interpreted as "touch the point". If there are no points on the table the imitator will tend to closely imitate the precise arm motions and the correct arm. The relation with world objects is considered the main task. It is a sensory problem to check for relevant points in the scene that might be correlated with the demonstrator actions.

As we see in this example, *imitation* does not have a unique definition. Sometimes, imitation is associated to different processes to achieve the same final behavior. From the study of imitation capabilities in animals, several mechanism were proposed to describe behavior that can generate an "imitative behavior", [Byrne, 1995, Byrne, 2002, Schaal et al., 2003, Schaal, 1999]:

1. **Stimulus Enhancement** describes the general tendency to respond more vigorously toward those parts of the environment, with which a conspecific is seen to interact. Learning by trial and error is very tedious, hard and time consuming. However, seeing what are the important parts of the environment and which objects might be useful, can speed up learning. It is a simple yet powerful mechanism which probably explains why many animals exhibit such

behaviour. Learning about good vs bad food and safe vs dangerous places is done by Stimulus Enhancement in birds, rats and apes.

2. **Response facilitation** is described by [Byrne, 1995] as "a kind of social effect that selectively enhances responses: watching a conspecific performing an act, often resulting in a reward, increases the probability of an animal doing the same." Large flocks of birds fly in perfect synchronization. They are not imitating each other, but simply doing the same action to protect themselves against predators.

3. **Contextual Learning** describes the situation when an action is not learned, but the perception of a new object property can produce the desire to act upon it. If, for instance, an animal sees someone throwing a coconut, it will learn the possibility of throwing that. The action itself is not new, since it was already present in the existing repertoire of the animal. In the context of this work, contextual imitation would amount to learning to employ an action in different circumstances, but not learning its form.

4. **Emulation** can also lead to a behavioral match. Observing an action and the corresponding result might bring a desire to obtain the same goal. Learning that a coconut can be smashed to reach the inside, will give the desire to eat the inside and thus producing the same behavior.

Although the mechanisms just described produce imitative behaviour, they do not exactly correspond to imitation learning, in the sense that no new actions are added or learned from scratch to the existent motor repertoire.

Instead, there is a second set of processes leading to imitative behaviour where learning of new actions does actually occur. This is called Production Learning [Byrne, 2002] and, as it is the most-powerful way of imitation, some authors [Schaal, 1999] call it "true-imitation". Byrne distinguishes two cases inside of Production Learning which are: **Action-Learning** and **Program-Level**. In the first case, imitation corresponds to the "exact" copy of the observed movement, the latter case is described to "acquire not only the surface form but also the underlying organization appropriate to the task"[Byrne, 2002].

1. **Action-Learning** is defined as: "The indiscriminate copying of the actions of the teacher without mapping them onto more abstract motor representation.", [Schaal, 1999]. This is a perfect copy of the motions, if the kinematics of the systems are the same, even the joint level trajectories are the same.

2. **Program-Level** Byrne [Byrne, 1999] introduced the *string parsing* mechanism describing, in a computational way, program-level. Consider an animal

observing a behavior unknown to him but consisting in known component actions. If the observer learns the task after being presented several times and without any kind of reinforcement, then program-level imitation is the only mechanism that can explain how it learned. In a macro-behavior consisting in smaller ones, the agent can decompose the overall behavior in known component actions, as a string parsing algorithm, to infer the task. This gives a sequence of basic elements, A, B, C,... and so on. After several presentations of the task a general grammar explaining the task can be inferred as the underlying structure of the behavior, including an hierarchical organized one.

1.1.2 Imitation in Artificial Systems

Having discussed the meaning, the variety and the utility of imitation in natural systems, we will now discuss how imitation has been implemented in artificial (robotic) systems. A thorough review can be seen in [Schaal, 1999, Breazeal and Scassellati, 2000, Meltzoff and Prinz, 2002, Schaal et al., 2003].

One first example of using imitation for programming is Tinker [Lieberman, 1993]. In a blocks world, the user can solve a task by describing the situation with Lisp commands. In this way, the user can program the system to solve the task, simply by showing how to do it. This is mainly used to teach the user how to program Lisp. Some degree of interaction with the user is necessary in order to disambiguate some situations. Several commercial products have evolved to facilitate programming with Java applets [Furse, 2001a].

One of the first works in imitation with robots was proposed in [Kuniyoshi et al., 1994]. It consists of a system capable to learn how to imitate an assembly task by extracting a hierarchical description of the task. A discussion about the advantages of learning by imitation is made by [Hayes and Demiris, 1994]. Imitation is used to transfer programs from one robot to the other. If a robot travels through a maze, a robot following it can associate its percepts at locations where some important action was performed by the teacher. The setup of maze following for imitation was further developed in [Billard and Hayes, 1997], in the context of grounding symbols to some specific perceptions, a neural network was also created to solve this problem [Billard and Hayes, 1999].

Another early work is presented in [Demiris and Hayes, 1996], where the orientation of a mobile robot is controlled according to the observed position and orientation of a human head. Usually, such tele-operation modality includes a filter to eliminate sluggish motion of the user and correct errors, e.g. see [Yang et al., 1994] in controlling an Orbit Replaceable Unit. From the observation of an operator doing a task, the robot generates an Hidden Markov Model describing the task. Another work using a similar approach for assembly tasks is described in [Hovland et al., 1996].

Learning by imitation has also been proposed in the context of humanoid robots [Schaal, 1999], where the number of degrees of freedom is very large. In [Matarict́, 2002], imitation is achieved by first classifying the observed motion and then eliciting the corresponding movement. When the matching error is large, a new skill is learned. In [Maistros and Hayes, 2001], a similar approach is done but the metric includes terms for the task goal as well as the task solution. In the work described in [D'Souza et al., 2001], the imitator can replicate both the demonstrator's gestures and dynamics. Nevertheless, it requires the usage of an exoskeleton to sense the demonstrator's behavior.

Regarding the imitation of facial expressions, the Kismet robot head [Breazeal, 1999] has achieved great popularity by being able to imitate emotions like happiness and sadness. The Infanoid project [Kozima, 2000] has been dealing with gesture imitation [Kozima et al., 2002], interaction with people [Kozima and Yano, 2001], and joint attention [Nagai et al., 2002]. Also developed at the National Institute of Information and Communications Technology in Japan, Keepon [Kozima et al., 2003] is a small robot with simple kinematics of four degrees of freedom. It can imitate gestures and interact with people. Both setups have been using to do "attention coupling" or "joint attention", that is spatio-temporal coordination of each otherŠs attention. It is claimed that this mechanism is a prerequisite for human-robot social interaction by allowing the human to attribute mental states to the robot.

Robota [Billard, 2003] is a mini-humanoid doll-shaped robot, created to interact with (normal and disabled) children. This robot can imitate arm and head gestures in an educational, entertaining game. Several positive effects result from the interaction of this robot with autistic children, as presented by [Robins et al., 2004]. In [Lopes and Santos-Victor, 2003, Lopes and Santos-Victor, 2005] a system is presented where a robot, equipped with a single camera is able to imitate arm gestures by assuming similar kinematics between demonstrator and imitator and including a "perspective-taking" algorithm that changes the viewpoint.

In the system described in [Zöllner and Dillmann, 2003], bimanual tasks are imitated using information about the functionality of each object and handling temporal task restrictions, in a symbolically manner. Since one object can have different functional roles, dependent on different contexts, multiple hypotheses are considered. In this way, the system can detect more reliably the goals and sub-goals of a human demonstrated task. The experiment was an household task like laying a table, pouring a glass of water and handling work tools.

In many works, imitation is presented as a goal in itself or some mean to interact with others. In [Price, 2003] imitation is viewed as a way to improve the speed of learning in multiagent environments. Agents share information by showing others their behaviors. The reinforcement learning theory, combined with imitation allows agents to combine prior knowledge, learned knowledge and knowledge extracted from observations. Some extra study is made about the use of different bodies, reward functions and multiple demonstrators. A system where a humanoid robot develops until acquiring the capability of imitating interactions with objects is presented in [Lopes et al., 2005].

The problem of inferring the important parts of the task is addressed in [Billard et al., 2004], and it is cast as an optimization framework. Motor theories of perception are used in [Demiris and Khadhouri, 2006] to build better imitation systems. The basic structure is a forward-backward model capable of prediction and/or reconstruction. Note that the backward model is not a simple inverse kinematics but includes a controller for a task solution. The problem of limited computation is also dealt with by controlling the attention.

In the following paragraphs, we will review some of the computational problems that need to be addressed to build an artificial system able to learn by imitation.

1.1.3 Computational Problems

Is Imitation a new learning method?

Mathematically speaking what is the difference between imitation learning and other kinds of learning? Most of the learning methods fall in the category of function fitting, the main difference between them is the way how the information is gathered to perform this fitting. Considering a function to learn:

$$y = f(x).$$

What information is usually provided to learning systems? For supervised learning a dataset consisting in samples (x, y) needs to be given. The learner must estimate a function that explains the data. For reinforcement learning, the dataset $(x, \hat{y}, r)$ consists of input (x), response $(\hat{y})$ and evaluation of this response r. In unsupervised learning, only the inputs, (x), are given. In learning by imitation, the dataset consists of observations of the corresponding function through highly non-linear functions, $(g(x), v(y))$.

There are several other reasons why imitation learning really differs from other mechanisms. Before doing a function fitting, a view-point transformation must be made to “align” the learner and demonstrator perceptual viewpoints. Then, some feature extraction or task abstraction should be guessed, as in the case of children imitating arm motions versus point reaching. Finally, the motor commands should be extracted from the demonstration and linked with one’s own body. It is also necessary to have a “perspective taking” to infer the goals of others . Finally to evaluate the quality of imitation some “metric” should also be guessed.

Imitation Metrics and Bodies's Correspondence

As we have seen, imitation either be used to improve task learning or as a goal by itself. During the first steps when learning-by-showing, the produced and observed actions need to be comparde continuously for evaluating the quality of imitation. We can thus formally define a **metric** to evaluate the success of an imitation task, see [Nehaniv and Dautenhahn, 2001] for a complete discussion about this subject.

Defining this metric is very difficult for the general case, since it must encompass the task goal, sub-goals, context, interaction with objects, etc. In addition, the demonstrator and imitator bodies may have differences in terms of kinematics, dynamics or size. As a consequence, the imitated movements cannot be exactly the same as those of the demonstrator and the metric must take all these into account.

The context of the task may depend not only on the object in the scene but also on the agent acting upon them. Using an object as if it were a hammer depends on the strength of the user. Something that is useful for the demonstrator can be irrelevant for the imitator.

As we have already discussed, the goal of imitation is often ambiguous. For dancing, the correct imitation is a direct association between the position of the teacher body and the students. For the case of dances in couples, the student should "mirror" the teacher, because the starting feet will be the opposite. In other cases of imitation, when the task is not so mechanic, the appropriate metric will be different.

Imitation (and thus the metric) can be defined at very different levels. One can consider the level of the actuators or object trajectories or an abstract task level. The metric can be dependent on trajectory matching, on states or on events. A metric will judge two actions with similar effects to be "close" in a quantitative way.

The metric need not necessarily be explicitly computed by the imitator, but can be present as a reward from the environment, in an ecological perspective. For instance, it can be implicit in the interaction with the environment via reinforcement feedback, verbal commands of a teacher, observed reward or feeding, etc.). It can also be evaluated by a closeness measure, where similar results can suggest similar executions [Nehaniv and Dautenhahn, 2001].

Some studies have been made to characterize the quality of imitation done by humans. In [Pomplun and Matarict́, 2000], subjects were asked to perform imitation tasks and quantitative results were obtained to assess the effect of rehearsal during observation and of the repetition of the task.

A linear combination of probabilities was used as a metric in [Billard et al., 2004]. The goal of this work was to discover which are the best methods for feature extraction both in manipulation and movement tasks. In this way, a more general imitation metric was obtained.

In [Lopes and Santos-Victor, 2003], two different metrics are presented for a humanoid robot to imitate arm movements. In the "full-arm" modality, the positions of the wrist, elbow and shoulder are required to be exactly copied. Instead, in the "free-elbow" version, only the position of the wrist and shoulder are considered relevant.

In [Schaal et al., 2003] a review of methods and techniques is given for motor imitation but it relies on the use of exoskeletons to acquire (3D) motion data of a demonstrator. In [Price, 2003], several metrics were defined for evaluating the improvement in learning using imitation.

The JABBERWOCKY system [Alissandrakis et al., 2005] imitates commands of a human demonstrator in a virtual world. Metrics such as absolute/relative angle and displacement aspects and the overall arrangement/trajectory of manipulated objects are used to describe the task.

When we observe someone imitating someone else, we implicitly assume some *correspondence* between the body of the demonstrator and that of the imitator. The *correspondence problem* is defined as the mapping between the actions, states and effects of the model and imitator, [Nehaniv and Dautenhahn, 2002] and it is particularly relevant whenever where actions performed by a specific body should be made by a different body. Even in similar bodies, contextual knowledge or training may imply that the demonstrator and the imitator can use objects in different ways.

Even when dealing with similar bodies, there are always (small) differences kinematics, size, dynamics or context in such a way that the correspondence problem must be solved. Of course, the choice of the imitation metrics and a specific solution to the correspondence problem are very intertwined process and, in many cases, there is not a clear distinction between them.

Imitation and skill transfer between systems with different bodies (kinematics, dynamics and skills) was addressed in [Nehaniv and Dautenhahn, 1998] using an algebraic formulation. Given an observed behavior of the model, starting from an initial state and moving toward a final state, it is necessary to map each action to the imitator embodiment, leading to a final state that is in correspondence with the observations.

A system for the transfer of traditional Japanese dances, from a human performer to a robot, is described in [Nakaoka et al., 2003]. As the mass distribution and dynamics of the two are not the same, it is necessary to adapt the observed trajectories in order to guarantee the correct balance during task execution. Also in [Demiris and Hayes, 1996] and [Nehaniv and Dautenhahn, 1998] work has been done in trying to map task between different bodies.

Quite often, there are situations where the correspondence and the metric are equivalent. In [Lopes and Santos-Victor, 2003], the metrics were defined as a function of the kinematics, i.e. either imitate the full-arm configuration or the wrist position only. The choice of the metric then imposes the use of a compatible sensory-motor map which, in turn, solves the correspondence problem.

Visual Transformations

Almost all imitation mechanisms require some degree of ability of the imitator to recognize or characterize the gestures or actions of the demonstrator. The approach can be as complex as a complete kinematic reconstruction, [Wu and Huang, 1999], or coarser representation of the gesture, as the match against some set of motor behaviors, or appearance based methods, [Bobick and Davis, 1996].

In this thesis we will rely extensively on vision as the primary perceptual clue for imitation. Vision allows us (and robots) to see objects, the task execution and the interaction between the demonstrator objects and the environment. However, the use of vision for imitation can be quite challenging because of the problem of "seeing the world from another's viewpoint" [Bruner, 1972].

Animals know their bodies from an ego-perspective (a perspective obtained from someone looking to its own body) but the demonstrator is seen in quite a different viewpoint, the allo-perspective (the view of other people's bodies). Hence, for imitation we need to consider a mechanism of *view-point transformation* (VPT) [Lopes and Santos-Victor, 2003, Asada et al., 2000] between the ego and allo-images.

The learner must perform a "mental rotation" to place the demonstrator's arm (*allo-image*) in correspondence with the learner's own body (ego-image). This *View-point Transformation* (VPT) is illustrated in Figure 1.2. Although the observed and observer's arms are in the same configuration, their image appearance is quite different in size, orientation, occlusions (the demonstrator's hand palm is visible but only the back of the observer's hand is visible).

In the absence of image descriptors invariant to view-point changes, the VPT is needed to map the gestures of a demonstrator to the (*ego*-)image, that would be obtained if those same gestures were performed by the system itself. Surprisingly, in spite of the importance given to the VPT in psychology [Bruner, 1972], it has received very little attention from other researchers in the field of visual imitation.

In [Gardner, 2002], two experiments are made to investigate neurophysiological responses to transformations of one's egocentric perspective. This test evaluated the response times of subjects when asked about which hand a manikin was using to grasp an object. Whenever there was a contrasting perspective (front view), the response was delayed. If the frame of reference was the same, no difference was observed. Similar results were obtained when the manikin figure was replaced by an abstract visual pattern.

One work that explicitly deals with the VPT is described in [Asada et al., 2000].

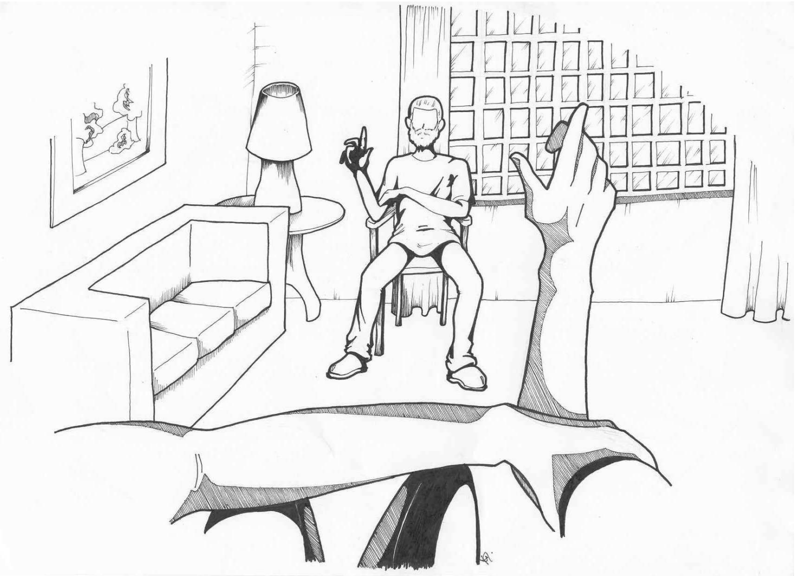

Figure 1.2: Similar gestures can be seen from very distinct perspectives. The image shows one's own arm performing a gesture (ego-image) and that of the demonstrator performing a similar gesture (allo-image).

However, instead of considering the complete arm posture, only the mapping of the end-effector position is done. The VPT is performed using epipolar geometry, based on a stereo camera pair.

In [Sauser and Billard, 2005], the VPT is learned and represented by means of a neural network. The network input is a vector encoding the direction and distance to a target, expressed in head centered coordinates. The network transforms this vector to body centered coordinates. A multilayer neural network is used to perform arbitrary three-dimensional rotations and translations.

Other studies address this problem only in an implicit or more superficial way. A mobile robot, capable of learning the policy followed by another mobile vehicle, is described in [Billard and Hayes, 1999]. Since the system kinematics is very simple, the VPT corresponds to a transformation between views of the two mobile robots. In practice, this is achieved by delaying the imitator's perception, until it reaches the same place as the demonstrator, without explicitly addressing the process of VPT. The work described in [Mataricć, 2002] allows a robot to mimic the "dance" of an Avatar. However, it does not address the VPT at all, and a special invasive hardware is used to perform this transformation.

1.1.4 Remarks

Imitation may allow the system to interact with other robots and users in a very simple way, to learn about new tasks. We have seen several aspects that need to be addressed for and artificial system to be able to learn by imitation. The question that remains is how can an artificial system develop such a capability? One approach can be cognitive artificial development, where the system skills would increase over time, through the interaction with the environment, people and objects. For inspiration, we will look at the development of sensory-motor and cognitive skills in human infants.

1.2 Developmental Robotics

In living beings, development from conception till adulthood is guided by a genetic program and the particular surrounding environment. The program is responsible to guide learning from the simplest to the most complex skills, the physical and cognitive capabilities developing from the interaction with the world, other people. Evidence in neuroscience suggests that the development mechanisms in our brain allow for a very powerful adaptation to the environment or even to injuries (brain plasticity). We will review some aspects of cognitive development in human infants and discuss how similar developmental principles can help building complex robotic systems, able to learn how to perform complex tasks by imitation.

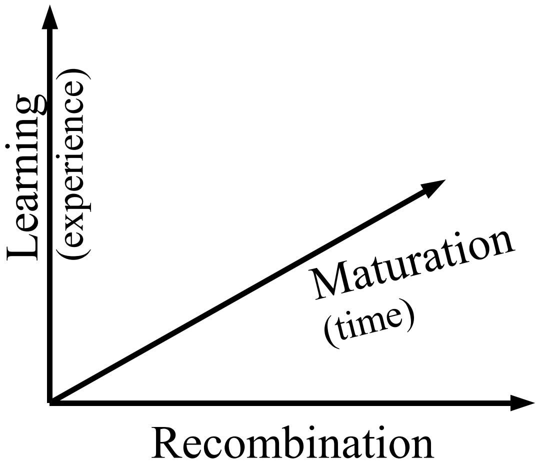

Figure 1.3: Main axis of development: learning, maturation and recombination

Broadly speaking, the process of development leads to an improvement of the skills of an individual or a machine, but this can be accomplished in several different ways. Figure 1.3 shows several axes of development that can be identified: learning, maturation and recombination.

Learning describes the improvement in the solution of a task by means of experience, e.g. improved posture control, as a result of parameter tuning. A different process, that can also result in improvement in solving a task, is the natural maturation of perceptual or motor mechanisms, e.g. due to physiological changes. This happens by means of a genetic program that improves neuronal connections resulting in finer perceptual and motor control, e.g. the stereo acuity of newborn grows over time until reaching a adult acuity at around 24 months [Banks, 1980, Birch et al., 1998, Birch et al., 2005]. The recombination component occurs when several mechanism able to solve different tasks, are integrated and put to work together. This can result in a temporary reduction in performance but, in the long term, more complex tasks can be executed. One example is the integration the skills of body posture control and leg control to acquire the capability of walking. When a newborn evolves from grasping and object from ballistic motion to visual controlled motion, there are two intervening mechanisms, one built on top (or with information from) the other.

In this thesis, there are several examples of learning but, recombination will be the main developmental mechanism adopted. The hypothesis is that a developmental strategy whereby existing mechanisms elicit the development of other mechanisms of higher complexity can be a very efficient way towards achieving sophisticated (visuo-motor or cognitive) capabilities. We will see such examples in the development of human infants before discussing how such lessons may be useful in robotics. At the same time, building robotic artifacts in a biologically plausible manner may help us understand how human cognition develops over time.

During the first months of life, infants have limited visual and motor capabilities. Both systems evolve side by side, with the visual system feeding information to "calibrate" hand/arm movements and arm movements providing stimuli to train and improve visual acuity . "*Several reflexes enable a good development of head and body control. During the last four months and the first four months post-natal, reflexive movements are so dominant that the human being has been labeled a "reflex machine". For nourishing and protection, the primitive reflexes are critical for human survival. The postural reflexes are believed to form a basis to more complex, voluntary movement of later infancy,*" [Payne and Isaacs, 1999].

Reflexes can be divided in primitive and postural reflexes. Primitive reflexes include palmar grasp, sucking, search, moro, startle, asymmetric tonic neck, symmetric tonic neck, plantar grasp, babinski, palmar mandibular and palmar mental. Postural reflexes consist of stepping, crawling, swimming, head-righting, body-righting, parachuting down/side/back, labyrinthine and pull-up. As one example, the "sucking reflex" enables a sucking action when there is a lip stimulation. Clearly, without this reflex, babies would not be able to eat or to learn how to eat.

For the case of the head-eye system, voluntary control appears very early. Several reflexive movements are evident from birth (head-righting reflex [Payne and Isaacs, 1999]), but voluntary control becomes apparent only at the end of the first month. A five-month old child already shows a good control. This control of the head will enable the tuning of the vision system to start looking at (and understanding) the environment. In [van der Meer et al., 1995] there is a discussion about the significance of neonate's arm movements. Usually, these motions are considered as unintentional, purpose-less, or reflexive. Some experiments indicate that newborns can purposely control their arms in the presence of external forces and that the development of visual control of arm movement is underway soon after birth. In these experiments, where a newborn could see its own arm in two conditions: direct observation of the arm or observation of the opposite arm, displayed on a video monitor. In a control condition, the newborn arms were not visible. Some small forces were applied to pull their wrists. The babies managed to oppose to the perturbing force so as to keep an arm up and moving normally. This only happened when they could see the arm, either directly or on the video monitor.

There are several reflexes that allow newborns to look at their hands. The "Asymmetric Tonic Neck Reflex" can be elicited when the baby is prone or supine. When the head is turned to one side or the other, the limbs on the face side extend while the limbs on the opposite side flex. This reflex is believed to facilitate the development of an awareness of both sides of the body as well as help develop eye-hand coordination by learning the relationship between motor actions and the corresponding visual stimuli.

Recombination allows the execution of complex behaviours through the integration of simpler ones, developed previously. For object grasping, there are two very distinct phases [Bower, 1977]. In Phase I there is simultaneous reaching and grasping. Reaching is visually initiated, while the grasp is visually controlled. In Phase II, there is a differentiation between reaching and grasping, the initiation and guidance of reaching is visually controlled, the grasp becomes tactile controlled.

It is interesting to see that different perceptual modalities are used in different developmental phases, ranging from visual control of grasp to tactile control. Table 1.1 shows how reaching and grasping develop in various steps, until achieving a very robust performance.

If we look at human infants, that the ability to solve very complex tasks is acquired after a number of developmental stages. Each stage is facilitated by the existence of a number of skills already in place, as a result of previous development steps. In turn, each stage allows for the improvement of specific skills which will facilitate the acquisition of new skills in more complex developmental phases. The process at different stages, is modulated by the interaction with the environment, objects or other people.

These observations suggested the adoption of a similar paradigm in robotics *The Developmental Approach to Robotics* [Weng, 1998, Asada et al., 2001, Lungarella et al., 2003, Lopes et al., 2005]. The developmental perspective, as proposed by e.g. [Weng, 1998], is a new paradigm aiming at overcoming the complexity problem, of learning and properly integrating many perceptual, motor or cognitive skills.

Age	Characteristics
Birth	Phase I reaching
1 month	Phase I reaching disappears
4 month	Phase I reaching reappears
4 to 5 month	Unable to receive multiple toys
5 to 6 months	Thumb used to oppose fingers in grasping
6 month	Phase II reaching appears
6 to 8 months	Receives two toys while storing one toy in opposite hand
9 months	Adjusts arm/hand tension to object's weight after grasping
18 months	Releases object with relative ease Anticipates arm/hand tension for same object repeated presentation Expects unknow long objects to weigh more than short objects

Table 1.1: Reaching and Grasping phases [Payne and Isaacs, 1999, chap. 11].

The robot should "start" with a minimal subset of core capabilities (as newborns do) to bootstrap learning mechanisms and the acquisition of new cognitive and behavioral skills. This is done in a progressive manner, through self-experimentation, interaction with the environment and humans, adaptation to particular contexts, and by integrating all the learning methods internally.

The tasks that the robot will be able to learn in this way should not be limited beforehand. Instead, the robot's behavior will be shaped by rewards or punishment obtained through the interaction with object and people. Reinforcement learning can be a powerful tool for this aim.

The development cannot be divided in components such as perceptual and motor. Both modalities are mutually necessary. The absence of one may make the task of the other impossible. In babies, we can see how arm movements can provide information for the eye system and vice-versa.

The main principles/requirements for a developmental machine can be summarized in seven points [Weng, 1998]:

1. Environmental openness

2. High-dimensional sensors

3. Completeness in using sensory information.

4. Online processing

5. Incremental processing

6. Perform while learning

7. Scale up to muddy tasks

In [Breazeal, 1998], a robot develops artificial emotions by interacting with people acting as a caretaker. The approach takes advantage of the social interactions for constraining learning.

In the system presented in [Lopes et al., 2005], a robot develops in several stages: eye-hand coordination, object localization and grasping and, finally, imitation of the interaction with objects.

A developmental approach is used in [Metta, 1999] for a robot that successively acquired vergence, saccade and vestibular control, as well as head-arm coordination. The system described in [Hotz et al., 2003] consists of a binocular head controlled by a neural network whose input and output resolution is improved with time (mimicking physiological maturation, another axis of development).

For reviews about this topic see [Weng, 1998, Metta et al., 2000b, Asada et al., 2001].

How to proceed next?

We have seen that development uses several mechanisms to deal with much of the complexity. The existence of reflexes allow the systems to tune sub-parts, by providing the training data and experience necessary. When these mechanisms are in-place, reflexes can disappear and the system can have its own motivation. The first development stage will thus be learning to coordinate its perception-action systems. In a subsequent phase, the motivation for interacting with objects and people in the environment will be the drivers of further development.

1.3 Approach of this work

Our main goal is to give a robot the capability of learning how to interact with the world by observing other people. We want the system to be able to learn all the necessary modules autonomously from sensory-motor coordination to high-level tasks. The systems should be autonomous from the beginning, certainly with some wired developmental strategies to allow him to increase its capabilities.

To solve this highly complex system and environment combined with the friendly interaction and robot teaching necessary will call for a very robust approach. We follow two major programming paradigms: *Imitation* as a way to let people show the robot the desired task and *Development* as a way to deal with the complexity.

The development of imitation capabilities requires an appropriate definition of the sequence of learning steps to reach that goal, as well as adequate performance evaluation methods to decide when to switch to higher developmental levels. In other words, it is important to define the overall hierarchy of developmental stages and the skills that must be acquired at each level. Table 1.2 shows the structure we adopt for the main developmental stages the robot will go through: (i) Learning about the self; (ii) Learning about objects and the world and (iii) Learning about others and imitation.

Table 1.2: Developmental pathway for the Perceptual and Motor capabilities (in *italic* the modules that are learned by the robot)

Time line	Perceptual/Motor Capabilities
sensory-motor coordination	eye vergence body recognition and tracking "random" movements *Arm-head coordination* *Hand-eye coordination*
world interaction	near-space mapping object recognition and tracking *acquisition of object affordances* *visually initiated reaching* *visual control of grasp*
imitation	task interpretation view-point transformation detection of other's actions *imitation of goal directed actions* *imitation of gestures* imitation metrics body correspondence

For each stage in this "developmental pathway", the set of skills to be acquired is presented, and the time line explaining the restrictions governing the system. It is not claimed any distinguish between innate versus learned behavior in biological systems ("the nature versus nurture" question). Instead, all the modules necessary to be present before the system can develop to the next level, are discussed. The sequence of learning stages is biological inspired but the specific division and implementation was a pragmatic option for having an autonomous performing robot. Even for artificial systems several mechanism are almost the same across levels. It is important to note that this division does not obliges that the levels are only incremental. Even when learning a module at a higher level it is possible to continue to adapt lower level modules.

In the first developmental level, the robot acquires very simple and yet crucial capabilities: vergence control and object foveation. Then, by executing "random" arm movements, in a self exploratory mode, it begins to coordinate perception and action capabilities, by creating sensory-motor maps (SMM). A first map, relating head with arm positions, is accurate enough to allow reaching for objects in easy positions. At this stage, it also recognizes its own hand and is able to relate the image of the hand with the correspondent motor inputs.

In the second developmental stage, the robot builds a map of the surrounding area (object positions and identification), studies object properties and how they can be used by others. Primarily driven by attentional cues, the robot engages in more challenging grasping tasks, for which the previously learned arm-head map is not sufficiently accurate. For that reason, a novel method for visually controlled grasping is presented, which improves over time and ensures the necessary robustness.

Two modalities are combined to allow for error correction in the learned maps, an open-loop phase moving the effector to the field of view followed by a closed-loop method with the precision necessary to put the effector in contact with the object. Special care is taken about the redundancy present in these complex robotic systems.

At the final developmental stage, the presence of a demonstrator will elicit imitation behaviors. Human gestures will be imitated, at an action-level, by using the learned maps. Goal directed imitation behaviors will decompose the actions and then replicate with a given metric. For this purpose, the system must be able to decompose the observed action into the relevant key elementary actions that must be executed for performing a task.

The imitation process consists of the following steps: (i) the system observes the demonstrator's arm movements; (ii) the VPT is used to transform these image coordinates to the *ego-image*, (iii) an action recognition is made to abstract the observed motion (if necessary), and (iv) the SMM generates the adequate joint angle references to execute the same arm movements. A schematic representation of this processing is present in Figure 1.4.

This roadmap is implemented in the humanoid robot Baltazar, shown in Figure 1.5 and described in [Lopes et al., 2004].

The remaining part of this chapter presents each level of this developmental architecture, following the structure given in Table 1.2. At each level, we describe both the main principles that guide development and the developed behaviors. At the end conclusions and future work is presented.

1.3.1 Sensory-motor coordination

Humans take a long time before becoming self-sufficient. Knowing how to walk, recognize objects, understanding how to solve a task, interacting with objects, are all very challenging problems. It takes several years, before all the necessary mechanisms to accomplish such goals, are available and working reliably. Infants have

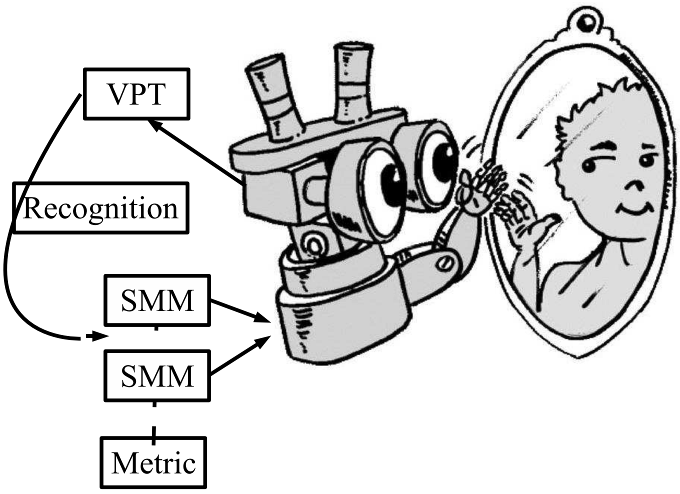

Figure 1.4: Imitation architecture. Observed actions are first transformed to a ego frame of reference (VPT) where some segmentation and recognition is made. After that an imitation metric and body correspondence is chosen (by selecting the corresponding SMM). In the end the imitation is performed

several mechanism guiding their development.

The eye-hand interaction allows a newborn to "tune" his eyes, distinguish depth and recognize touchable objects. For a baby, exploring the motion and visual appearance of its own hand, is a very interesting thing to do during the first few months of life. Through a complex learning process, the baby will gradually learn how to control his own body to do a specific task. In the end, the reward will be tremendous. The baby will be able to predict hand movements and to control his arm/hand to grasp and explore nearby objects.

Similarly to human infants, a robot must have a way to predict what happens in the world as a consequence of its own actions (*forward model*). Conversely, it is important to know which action will change the world in a pre-defined manner (*backward model*). Usually, this correspondence between perception and action is called a *Sensory-Motor Map* (SMM) and it can be interpreted in terms of forward/inverse kinematics of robotic manipulators. In this work, the SMM is used to predict the image (or the image transformation) resulting from the robot moving the arm to a certain posture, or the inverse association, by determining which motor command causes the arm to reach a specified appearance.

These models can be multimodal, including visual, motor or sound information. It is possible to predict the shape of the hand corresponding to a certain pattern of activation of the hand/arm muscles. In the inverse case, we know which muscles to activate in order to control the hand to a given (observed) configuration. With an image of an object, it is possible to predict the muscle force necessary to hold it.

In some cases, these sensory-motor maps relate motor information in different parts of the "system" (or body). For bipedal locomotion, the whole body must be balanced, so that a change in the arms posture is compensated by some hip change. A bidirectional *Arm-Head Map*, mapping between arm and head position, can be used in two ways. If the head position is fixed, moving the arm to the mapped position will drive the hand into the center of the field of view of the two eyes. Instead, if the arm is fixed, we can visually locate it by moving the head to the mapped position.

A solution for learning perception-action maps in the presence of redundant degrees of freedom was developed, which occurs frequently for humanoid robots. Our approach does not restrict the output of the system at learning time and, as a consequence, the extra degrees of freedom remain free to be used online in a secondary task.

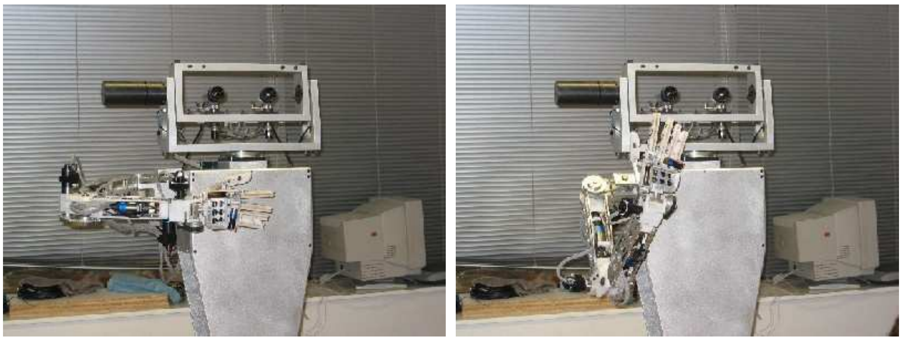

Figure 1.5: Sensory-Motor Coordination learned by self-exploration. With redundant robots, it is much harder to associate perception to actions, as multiple actions may yield the same percept.

A final aspect worth mentioning is that all these sensory-motor maps are learned during an initial phase, when the system (natural or artificial) performs arm/hand gestures and observes the (visual) consequences of such gestures. Both proprioceptive (motor) and visual data are present and the association can be established. An additional comment is that self-observation may allow the system to search and tune the most interesting visuo-motor features, such that a more compact representation of the visual space could be used.

1.3.2 World interaction level

As the robot gains control over its own perceptual and motor capabilities, it gets more and more interested in exploring the surrounding world. This exploratory motivation will call for the development of more advanced manipulative capabilities as opposed to the rudimentary skills available during phase one.

When exploring the world one realizes that objects may have different properties and functions. Some are graspable, some can be combined, some can be eaten and there are others that can move by themselves. To a great extent, our ability to interact with the world and all its entities, is intimately connected with the knowledge and understanding we have about the world and object properties.

To be able to predict changes in the observed world, we need to understand the dynamics of objects with different motions and speeds. Every system needs a model of the environment, even if it is not very explicit, that describe the world dynamic evolution. The development of an autonomous agent is strongly dependent on this ability to model and predict events in the world. The process of acquiring this knowledge requires interacting and probing the world. Then, the richness of our models and our ability to understand and predict world events, will improve over time, in an ecological developmental process, by observing and interacting with the world. Figure 1.6 shows Baltazar reaching for objects to explore their properties.

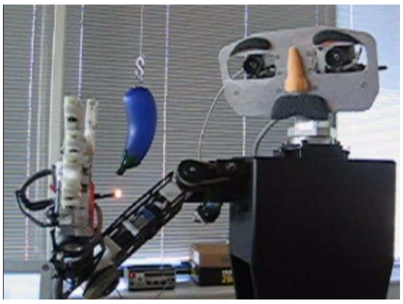

Figure 1.6: World level behavior. Grasping and interacting with objects to learn their properties and recognize similar actions performed by others.

In this level, our robot will learn about object properties by interacting with them and recognizing similar actions performed by others. However, in the beginning of this stage, all the robot can do is to fixate salient objects and approach them in a primitive form of grasping. At this stage, the development path requires the following new skills:

1. Detect object's positions in the nearby space and store this information in body centered coordinates (near space map).

2. Track moving objects.

3. Learn how to use objects (affordances) by observation and/or interaction.

4. Learn how to reach objects (using visual feedback).

5. Learn how to preshape the hand to grasp objects.

6. Recognize similar actions performed by others.

It is very useful to know where an object is and whether it can be grasped or not. After all the time spent interacting with its own hand, the system can already distinguish objects at different depths and search for the desired one. The robot thus creates a mental image of the surrounding space. The position of the objects are memorized in terms of proprioceptive (head-centric) coordinates. In a case of a moving robot, this map would need to be updated with the ego-motion.

Several motor programs are necessary for object grasping: the arm must be able to approach the object (reaching), correct possible errors with visual feedback and finally grasp it; the hand must be able to have a stable grasp and pre-shaping can be necessary for faster movements or moving objects. In addition, the system must know what is the best direction to approach the object, because specific objects have specific ways of grasp, e.g. consider the difference between grasping a cup or a ball.

This grasping mechanism is constructed upon three different sensory-maps learned in the previous stage of development. First, the head moves in order to have the eyes gazing at the object. After that, the head-arm sensory-map is used to move the arm into the field of view, as near as possible to the object. As soon as the hand is detected in the image, a visual servoing loop is responsible for the final part of the grasp movement. Upon contact with the object, the hand finally closes.

We also present a methodology that allows the robot to recognize grasping actions performed by other individuals. The approach is inspired by the finding of the mirror neurons in the macaque monkeys brain by neuroscientists, leading to the hypothesis that the ability of recognizing someone's gestures is facilitated by the fact that the system knows how to perform those same gestures. The methodology relies on knowledge about a repertoire of actions the system (or animal) can perform, knowledge about the demonstrator's and its own body, all mapped into motor information. In the perspective of development, observing the way objects are grasped is useful in two ways. One can learn successful ways to grasp objects and also, possible ways to use those objects. Therefore, recognizing grasping actions performed by others, is important to learn about objects and how to interact with them.

1.3.3 Imitation level

The final developmental stage addressed in this thesis endows the robot with the ability to imitate a task performed by a demonstrator. The overall process is represented in Figure 1.4, already discussed.

The first problem the system has to solve for an imitative task is the View Point Transformation (VPT), a mental rotation to align the image of the demonstrator with its own body coordinate frame. The structure of the chosen VPT is determinant to the class of imitation behaviors that can be generated. We consider two different cases. In the first case - 3D VPT - a complete three-dimensional imitation is intended. In the second case - 2D VPT - the goal consists in achieving coherence only in the image, even if the arm (3D) pose might be different. While the 3D VPT allows the robot to replicate the demonstrator arm movements in a faithful manner, the 2D VPT may only allow to accurately reproduce the hand movement (but not the elbow). Ultimately, the choice of the VPT is determined by the so-called imitation metric that identifies the exact goal of the task to imitate (in other words, what to imitate).

After recognizing the demonstrator and having translated the observed action to a canonical description by using the VPT, it is necessary to transform it to valid robot action. This *body correspondence* is solved using the machinery learned in previous levels of development, namely the learned visuo-motor maps and knowledge of the human body kinematics.

With these behaviors, it is possible to achieve two types of imitation: action-level and program level. The goal of action level imitation is the exact reproduction of the gestures performed by the demonstrator. If someone waves “goodbye”, the system will repeat the same gesture, with the same speed, amplitude, etc. Instead, in program level imitation, the goal consists in reproducing the observed changes in the world state (objects positions, etc), but not necessarily the specific gestures to achieve that. This aspect is particularly important when doing object interaction. The system observes someone else’s actions and movements and will need to abstract a symbolic description of the task to be executed later on. For illustrative purposes, a result of action-level imitation is shown in Figure 1.3.3.

1.4 Thesis Contributions

One main contribution of the thesis is the proposal of a developmental roadmap for imitation, allowing the robot to acquire competencies through several incremental stages. It starts by exploring his own body, continues to learn and develop by interaction with the world and finally is able to learn by observing people.

We have as target giving a humanoid-type robot the possibility to learn how to perform a task by observing a person executing that task, involving arm motions and interaction with objects. With this goal in mind, we identified all necessary perceptual and motor skills organized in three levels, according to widely accepted stages in developmental psychology: individual, world and social. Every level builds on top of the previous levels. At each level, the robot is able to solve a set of problems of a given complexity. Since the beginning, the system is autonomous, in the sense that learning and acting occur simultaneously, and all the training stimulae is self

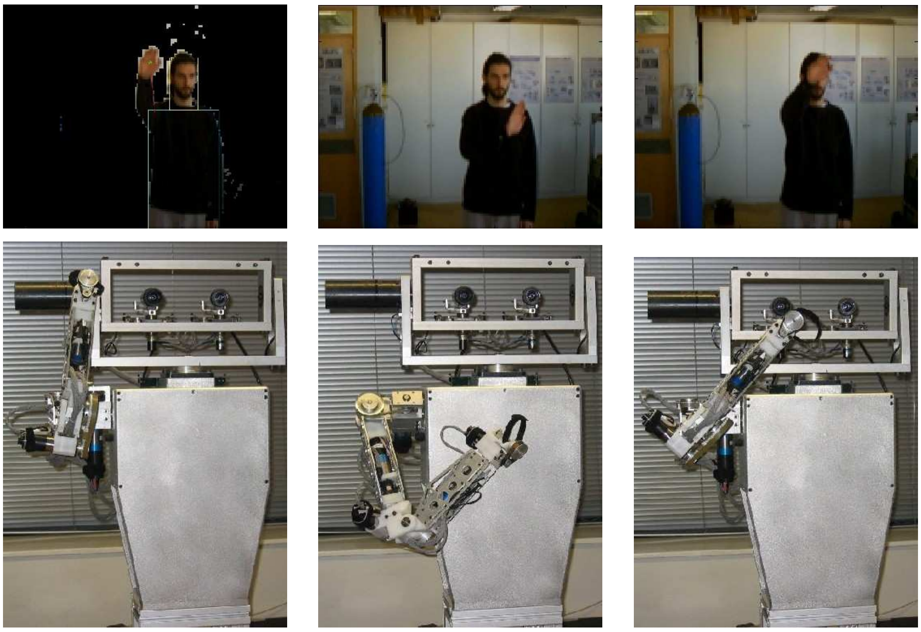

Figure 1.7: Robot imitating the gestures of a person.

generated.

In more detail, we can highlight a number of contributions in the different parts of this thesis:

In Chapter 2 - Sensory motor coordination - the main contributions consist of the use of statistical learning methods for estimating Sensory-motor maps in redundant robots. Also, we proposed improvements to uncalibrated visual servoing through the decomposition of a complex redundant task into a series of simpler, non-redundant ones.

In Chapter 3 - World interaction - the main contribution is a Bayesian model for the canonical and mirror neurons for gesture recognition. This process involves learning object affordances and recognition, by considering iconic and motor representations of the grasping process. In addition, we proposed a two-step visually-guided grasping process where learning is driven by the motivation to precisely grasp objects, that continuously adapts over time (open ended learning).

In Chapter 4 - Imitation - our main contribution is to show that the information learned beforehand, can indeed allow the system to imitate observed gestures or tasks, as a consequence of the proposed developmental process. Another contribution is the use of vision to acquire the demonstrator's data, which involved the definition of several structures of View-Point Transformation. The correspondence problem is implicitly solved through an adequately chosen sensory-motor map and several metrics for imitation are discussed, particularly for action-level and program-level imitation.

1.5 Organization of the thesis

We presented a developmental route for creating an humanoid robot [1] able to learn by imitation. As in general, both the system and the task can be very complex, this strategy allows the system to acquire increasingly more complex skills, as the tasks difficulty are increased slowly. We described results, implemented in a robotic system, of the various developmental stages of the system. This developmental roadmap is organized in three main levels, each of which corresponding to a chapter in the thesis.

Chapter 2 is dedicated to sensory-motor coordination, where robot first learns about the control of its own body through self-exploration, to build a variety of sensory-motor maps. In the end of this stage the coordination (e.g. between the head and the arm) achieved is sufficient to ensure that the hand always remains in the image and that a primitive form of grasping can be used.

Interaction with objects, exploration of object properties and affordances, learning and understanding actions exerted upon objects are dealt with in Chapter 3. In this phase, the system is driven by the motivation to interact with and explore nearby objects. The system develops the ability of precise grasping, resorting to closed-loop control loop driven by visual feedback. The grasp method consists in two phases: an open-loop controller putting the hand close to the object, and a closed-loop vision-based controller for precisely touching the object. This method does not need calibration and can be learned on-line in a very efficient way. It also creates a map of the interesting objects in the surrounding space. Finally, the system learns how to recognize gestures provided by others, a process facilitated by its own ability to perform those gestures and actions.

The final stage of development where the system becomes able to do imitation is presented in Chapter 4. The system's attention is driven toward people acting in the environment. Different metrics and types of imitation are introduced, namely the imitation of gestures (action-level) and goal directed tasks (program level). For program-level imitation, the system first builds an abstract description of the observed task (object manipulation), that is used later on for imitation.

Finally, conclusions and future work are presented in Chapter 5. In Annex A we present the kinematics of our robot, Baltazar, used for the experiments.

[1] see http://vislab.isr.ist.utl.pt/baltazar for videos showing some of the experiments in this work

Chapter 2

Sensory-Motor Coordination

As we have seen in Chapter 1, the first level of development, both for animals or robots, involves learning how to coordinate the perception and action systems, referred to as *sensory-motor coordination*. Coordination between perception and action is a fundamental skill for animals and robots to "operate" in the world. One example is the coordination between the eyes and the arm/hand to reach for an observed object. In addition to using each sub-system individually, it is important to learn how to use them in a coordinated way.

Usually, coordination between perception and action is described in terms of *Sensory-Motor Maps* (SMMs) that establish a bidirectional association existing between sensory perception and motor action. On one side, there is a transformation from the space of motor actions to the space of resulting sensory situations (the forward association). On the other side, one needs to know how to generate motor commands to reach a target sensory perception (the inverse map).

In this thesis, the forward SMM is used to predict what will happen in the world (e.g. the observed configuration of the arm), if some motor action is executed by the robot. The inverse association is useful to determine which motor commands drive the arm towards a specified appearance or image configuration, typically for control purposes. In the context of robotics, these maps can be interpreted at the forward/inverse kinematics of articulated chains with the difference that sensor signals (e.g. vision) are also considered.

From a methodological point of view, the study of robot sensory-motor coordination (both forward and inverse) may be addressed in two different ways. The classical approach consists of assuming a *prior model* of the interaction between sensors and actuators, i.e. a model built on the basis of physical laws. In this way, the forward association can be expressed as a direct kinematics equation and the inverse map would correspond to the computation of a solution to the inverse kinematics equation, needed to position the end effector.

The alternative approach, that we adopt in this thesis, consists of extracting directly from data a model describing the association between sensory perceptions and motor commands. First, <perception, action> pairs are collected by having the system operating and observing the consequences of those motor actions, i.e. through auto-observation. Then, a learning method is used to estimate the model. With this framework, no fixed model is assumed to be valid before experimenting with the real robot. This approach contrasts with the use of a prior model, that describes ideal interactions which are only met in simulated environments.

This adoption of an initial "calibration" process, relying on auto-observation, follows the general developmental guidelines and should be as autonomous as possible. The system creates its own excitation actions which, in turn, allow it to gather enough information to coordinate its own body. With this framework, an artificial system is able to start solving tasks, in a limited way, very early in the learning process. Then, as time goes by, solutions for certain tasks can be improved by exploiting the availability of more data. This process of learning by means of self-exploration will be frequently used in this thesis.

Since an SMM associates sensory perceptions and motors commands, we can define a large variety of maps, as a function of the sensing/actuation modalities and the structure of the input/output data. Motor commands can be joint torques, velocities or positions and sensor signals can be shapes extracted from vision, sounds or proprioception about the body state (tactile) or motor actuation. Figure 2.1 represents a sensory-motor map:

$$\begin{array}{ccc} \text{Motor} & & \text{Sensor} \\ \text{commands} & \overset{SMM}{\longleftrightarrow} & \text{signals} \end{array}$$

Figure 2.1: General diagram of a Sensory-Motor Map.

Depending of the sensing modalities involved, we can consider visuo-motor, auditory-motor or motor-motor maps. A *visuo-motor* map determines the motor program necessary to elicit a desired image configuration of the arm. In some situations, a *Motor-Motor Map*, relating the head/arm coordinate frames, may be necessary to place the arm in front of the head. Here, the proprioception of the head position is mapped to the action of the arm. Auditory-motor maps are very important for speech and language acquisition. The babbling is a way to generate data pairs with the motor commands of the vocal tract and the produced sounds.

In addition to the different sensory modalities involved in a SMM, we can further distinguish different types of SMMs, according to the nature of the mapped information:

- static vs incremental. An SMM can describe a static relation between the input and output or it can relate input variations to output variations. The static version is useful for positioning while the incremental map is necessary for closed loop control.

- full vs partial. In a full map, we consider that the input completely determines the output, meaning that the task is non-redundant. If there is some degree of redundancy, either in the actuation or in the task itself, the number of admissible solutions will be infinite. In such a case, the map determines the output only partially and an extra optimization process is needed to identify a unique solution.

- geometric vs radiometric. In most cases, the SMMs we have discussed describe the geometry of observation and actuation. However, in some cases, we can consider radiometric maps that describe the visual appearance of an object (e.g. the hand) in addition to its coordinates in the field of view.

In the remaining part of the thesis we will see how these different sensory-motor maps are selected and used in different stages of the robot's development. Sensory-motor coordination will be a core skill to enable more complex behavior.

Static maps will be used whenever the goal consists in positioning the robot arm in some configuration, expressed in retinal coordinates. One example is the reaching movement, whereby the arm must reach a neighborhood of an object of interest. However, the final movement must be controlled with an incremental SMM to produce relative motions of the arm.

One challenging problem in humanoid robots is that the robot often has more degrees of freedom available than those strictly necessary to accomplish a certain task. For example, Figure 2.2 shows several positions of our humanoid robot (Baltazar), where the wrist position is always the same, but the posture of the arm changes. In terms of Sensory-Motor Maps, this redundancy translates into the fact that several different motor configurations yield the same observation. Then, the backward model cannot be obtained by inverting the forward model, because multiple (possibly infinite) solutions exist and the map is non-invertible. As a consequence, common algorithms will fail to learn the inverse model directly, because of incoherencies in the dataset.

One approach to overcome this problem consists of adding extra assumptions involving the redundant degrees of freedom, e.g. that they minimize some sort of energy criterion. This strategy constrains the space of solutions so that sensory-motor maps in redundant systems can be learned. The main drawback is that the "frozen" degrees of freedom cannot be used for any additional "useful" task.

We took a different approach by introducing the concept of a "partial" SMMs. The rationale is to partition the existing degrees of freedom between redundant and non-redundant, with respect to a certain task. Then, the partial SMM will only

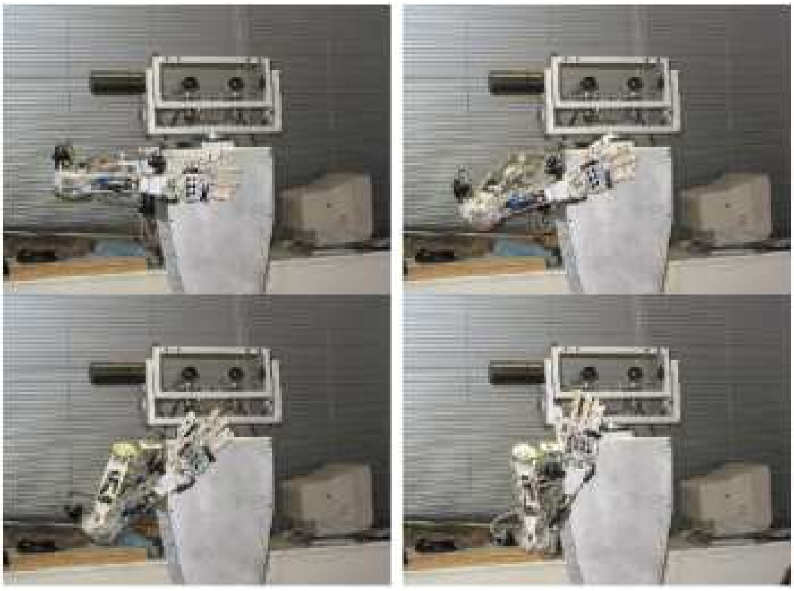

Figure 2.2: Redundancy of the robotic system, the $3D$ position of the wrist is the same but the arm configuration is different.

map the observations to the non redundant degrees of freedom. The remaining (redundant) degrees of freedom can be used to execute a secondary motion, along a direction in the null space of the first motion [Rosen, 1960, Samson et al., 1991]. For instance, one can move the hand toward a point while, at the same time, the elbow avoids some obstacles.

Several criteria can be used to choose the secondary task. This formalism is very well described by [Baerlocher and Boulic, 2004], applied to humanoid animations. In humanoid robots, several constraints can be conflicting, like the position of the hands, feet and head in a dance posture. The use of the redundancy formalism and the partial SMMs can take all these constraints into account.

We applied this paradigm for dealing with redundant robots in two situations. In the static case, we use a partial static map. An analogous solution is proposed for closed loop visual control (visual servoing), with a partial incremental sensory-motor map.

The problem of head/arm coordination for robots has been addressed in various works in the literature. An example of a full static map used is presented by [D'Souza et al., 2001]. A statistical method is used to learn an inverse kinematic function, for a highly redundant humanoid robot engaged in imitation tasks. In this case, the map relates articular joint positions and velocities to image velocities, enabling the robot to mimic observed arm gestures. The trajectories for learning were hand-coded. The authors claim that their approach solves the problem of robot singularities, because infinite velocities are not represented in the training data. The disadvantages arise from the lack of extrapolation capabilities and by not having an explicit Jacobian estimation, thus needing more time to gather the information, and preventing the use of well studied visual servoing control algorithms.

The system in [Rougeaux and Kuniyoshi, 1998] uses the assumption that the eyes are always looking at the target. It uses a motor map between 4 eye/head joint angles and 3 arm parameters. The mapping is learned during a training phase, in which the eyes observed the hand. After the training phase, no error correction mechanism is used.

In [Metta et al., 1999], a single motor map was used to obtain 2 arm control parameters from 2 eye/head parameters. Eye vergence is used to control depth [Metta et al., 2000a]. If the reaching is done in a pure open-loop manner, there will be errors that need to be corrected later on with a closed loop mechanism.

A neural-network architecture was used in [Blackburn and Nguyen, 1994] to coordinate a binocular head with a three degree of freedom arm for the reaching task. In addition to static information, his work also included error correction methods and closed loop visual servoing. Similarly, the work described in [Marjanović et al., 1996] relies on a combination of a ballistic map for the open-loop initial command of the hand, followed by a minimum jerk control to precisely point the hand to an object.

This chapter is organized as follows. First we derive a SMM based on a kinematic model of our robot. This study enables us to understand the structure of the sensory-motor map and provides insight as to how learning can be used. In addition, we use this map to exemplify the concept of partial/full SMM, for redundant robots.

We then present a sensory-motor map for a robotic hand, based on a neural network, that relates the hand appearance to the corresponding motor commands. The relevant issue here is that the SMM generates image (irradiance) information directly instead of geometric information only.

We move on with the analysis of static sensory-motor maps, focusing on the problem of redundancy. We present our approach based on partial SMMs and detail how such maps can be learned on-line during the initial developmental stages of the system. The final part of the chapter is dedicated to dynamic (incremental) SMMs, looking at the problems of redundancy, learning and control.

2.1 Arm (Kynematic) Sensory-Motor Map

In a first approach we are going to build a SMM by considering explicitly the robot's kinematics. This map allows the system to relate the geometric position of wrist, elbow and shoulder from an image and determine the motor commands that yield the same arm appearance.

It is assumed that the observation is done with a single orthographic camera. With a single image of the arm, it is not possible to unambiguously determine the arm 3D configuration, because a whole class of positions can give the same image appearance. Because of this, extra assumptions must be made in order to choose among the possible alternatives. The kinematic restrictions of the robot reduce the set of possible solutions.

In this section, two approaches (a partial and a full map) for this problem are presented. A first solution, using information about elbow position in the image and a scale factor, relies on a partial euclidean reconstruction of the arm. A second solution defines just an image reconstruction letting the depth information and the elbow position free. It is simpler to construct because there is no need to track, over time, the elbow position in the images.

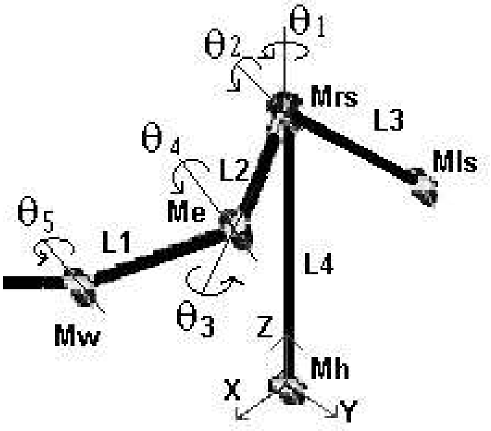

Figure 2.3: Kinematic model of the robot used in the experiments. For more details refer to Annex A.

2.1.1 Full-Arm SMM

We denote the elbow and wrist image coordinates by $\mathbf{m_e} = [v_e\, u_e]^T$ and $\mathbf{m_w} = [v_w\, u_w]^T$, corresponding to the cartesian coordinates $\mathbf{M_w}$ and $\mathbf{M_e}$, the forearm and upper arm image lengths by l_1 and l_2, with corresponding cartesian values of L_1 and L_2 and $\theta_{i=1..4}$, the various joint angles (quantities defined in Figure 2.3). We assume the camera is positioned in the shoulder facing down. We then have:

$$[\theta_1, \cdots, \theta_4] = \mathcal{F}_1(\mathbf{m_e}, \mathbf{m_w}, l_1, l_2, L_1, L_2, s)$$

where $\mathcal{F}_1(.)$ denotes the full-static SMM, L_2/L_1 represents the (known) upper/forearm ratio and s is the camera scale factor (refer to Annex A for the kinematic description of the robot).

The computation of this function can be done in successive steps, where the angles of the shoulder joint are determined first and used in a later stage to simplify the calculation of the elbow joint's angles.

The inputs to the SMM consist of features extracted from the image points of the shoulder, elbow and wrist; the outputs are the angular positions of every joint. The shoulder pan and elevation angles, θ_1 and θ_2 can be readily obtained from image data as:

$$\begin{aligned} \theta_1 &= f_1(\mathbf{m_e}) = \arctan(v_e/u_e) \\ \theta_2 &= f_2(l_2, L_2, s) = \arccos(l_2/sL_2) \end{aligned}$$

Once the system has extracted the shoulder angles, the process is repeated for the elbow. Before computing this second set of joint angles, the image features undergo a set of transformations so as to compensate the rotation of the shoulder:

$$\begin{bmatrix} u'_w \\ v'_w \\ \xi \end{bmatrix} = \mathcal{R}_{zy}(\theta_1, \theta_2) \left(\begin{bmatrix} u_w \\ v_w \\ \sqrt{s^2 L_1^2 - l_1^2} \end{bmatrix} - \begin{bmatrix} u_e \\ v_e \\ 0 \end{bmatrix} \right) \tag{2.1}$$

where ξ is not used in the remaining computations and $\mathcal{R}_{zy}(\theta_1, \theta_2)$ denotes a rotation of θ_1 around the z axis followed by a rotation of θ_2 around the y axis, the image lengths are transformed to l'_1 and l'_2.

With the transformed coordinates of the wrist, we can finally extract the remaining joint angles, θ_3 and θ_4:

$$\begin{aligned} \theta_3 &= f_3(\mathbf{m'_w}) = \arctan(v'_w/u'_w) \\ \theta_4 &= f_4(\mathbf{m'_w}, L_1, s) = \arccos(l'_1/sL_1) \end{aligned}$$

The approach just described allows the system to determine the joint angles corresponding to a certain image configuration of the arm. In the next section, we will address the case where the elbow joint is allowed to vary freely.

Rather than coding these expressions directly we can have a learning approach whereby the system learns the SMM by performing arm movements and observing the effect on the image plane. Learning the SMM can be done sequentially: estimating the first angle, which is then used in the computation of the following angle and so forth. This fact allows the system to learn the SMM as a sequence of smaller learning problems.

In all cases, we use a feed-forward *Multi-Layer Perceptron* (MLP), with an hidden layer consisting of five neurons, to learn the SMM, i.e. to approximate functions $f_{i,i=1..4}$. We have a MLP for each joint, after estimating θ_1 and θ_2 we can use the given transformation and then have the input values for evaluate θ_3 and θ_4. Table 2.1 presents the learning error and illustrates the good performance of our approach for estimating the SMM. The value 3.6 corresponds to the threshold for the training

θ_1	θ_2	θ_3	θ_4
$3.6e^{-2}$	$3.6e^{-2}$	3.6	3.6

Table 2.1: Mean squared error (in deg.^{2}) for the each joint in the *full-arm SMM*

algorithm. The order of magnitude is 100× bigger in the last 2 degrees of freedom because they depend on the previous ones in a non-linear way.

Ideas about development can be further exploited in this construction. Starting from simpler cases, de-coupling several degrees of freedom, interleaving perception with action learning cycles are developmental "techniques" found in biological systems.

2.1.2 Free-Elbow SMM

The *free-elbow* SMM is used to generate a given wrist position, while the elbow is left free to reach different configurations. This partial map is different to the previous full map where all degrees of freedom were restricted.This map can thus achieve the desired goal but is able to use the redundancy to improve posture control and/or to avoid obstacles. It is motivated by the fact that many tasks do not need a very precise depth control, e.g. saying goodbye.

The input features consist of the hand image coordinates and the depth between the shoulder and the hand ${}^{r}dZ_w$.

$$[\theta_1, \theta_2, \theta_4] = \mathcal{F}_2(\mathbf{m_w}, {}^{r}dZ_w, L_1, L_2, s)$$

The elbow joint, θ_3, is set to a comfortable position. This is done in an iterative process aiming at maintaining the joint positions as far as possible from their limit values. The optimal elbow angle position, $\hat{\theta}_3$ is chosen to maximize:

$$\hat{\theta}_3 = \arg\max_{\theta_3} \sum_i (\theta_i - \theta_i^{limits})^2$$

while the other angles can be calculated from the arm features (a more general approach will be presented in latter sections). Again, the estimation process can be done sequentially, each joint being used to estimate the next one. Considering the coordinates of the wrist in a robot-centric frame $\mathbf{X} = [{}^{r}x_h \; {}^{r}y_h \; {}^{r}z_h]^T$:

$$\begin{aligned}
\theta_4 &= \arcsin\left(\frac{{}^{r}x_h^2 +{}^{r} y_h^2 +{}^{r} z_h^2}{2} - 1\right) \\
\theta_1 &= 2\arctan\left(\frac{b_1 - \sqrt{b_1^2 + a_1^2 - c_1^2}}{a_1 + c_1}\right) \\
\theta_2 &= 2\arctan\left(\frac{b_2 - \sqrt{b_2^2 + a_2^2 - c_2^2}}{a_2 + c_2}\right) + \pi
\end{aligned}$$

where the following constants have been used:

$$\begin{aligned} a_1 &= \sin\theta_4 + 1 \\ b_1 &= \cos\theta_3 \cos\theta_4 \\ c_1 &= -{}^r y_h \\ a_2 &= \cos\theta_4 \cos\theta_2 \cos\theta_3 - \sin\theta_2 (1 + \sin\theta_4) \\ b_2 &= -\cos\theta_4 \sin\theta_3 \\ c_2 &= {}^r x_h \end{aligned}$$

2.1.3 Conclusions

We have used the robot kinematics to derive a *Sensory-Motor Map* that maps observed actions into motor data. Two different types of SMM were proposed, depending on whether the task consists of imitating the entire arm or the hand position only. These maps are instances of full/partial SMMs, as discussed in the previous chapter. We have seen how the structure of the SMM is amenable to a sequential estimation process, with resemblances biological development. Finally, we have demonstrated how this map can be learned automatically from data obtained during a period of self-observation.

2.2 Hand (Appearance) Visuo-Motor Map

For the case of hand postures, it is also possible to define a relation between an image and the corresponding motor variables. In a way, it is easier to do it because during self-observation, the system can generate a large variety of hand visual stimuli like palm and back view. Instead, the arm is rarely seen in its full extent. However, the hand is frequently subject to occlusions and the number of degrees of freedom is much larger.

The learning strategy consists in estimating a subspace spanning hand images taken from a variety of view-points. The hand Visuo-motor Map (VMM) relates the hand image (normalized for orientation and scale) directly to the finger joint angles (motor data).

As the transformation from the visual space to the motor space is quite complex to model, it was learned with a Multi-Layer Perceptron, for each joint angle. For each network, i, the input consists of a 15-dimensional vector $\mathbf{F}^V$, which are the Principal Components Analysis (PCA) components of the imaged hand appearance. The output consists of a single unit, coding the corresponding joint angle, $\mathbf{F}_i^M$. There are 5 neurons in the hidden layer.

We assume that $\mathbf{F}^V$ is captured across many different view-points. This is possible to generate during self-observation, since a huge variety of hand configurations can be easily displayed. Otherwise, a view-point transformation is needed to pre-align the visual data [Lopes and Santos-Victor, 2003].

If this map is used in real-time, it can lead to impossible (temporal) trajectories, as errors in input frames can cause discontinuities in the motor space. To overcome this problem, continuity is imposed in the motor data through a first-order dynamic filter.

Each neural network was trained with momentum and adaptive *back-propagation* with the data pre-processed to have zero mean and unitary variance. It converges to an error of 0.01 in less than 1000 epochs. Figure 2.4 shows trajectories (solid-line) for a joint angle (real and estimated) of the little finger when performing several precision grips.

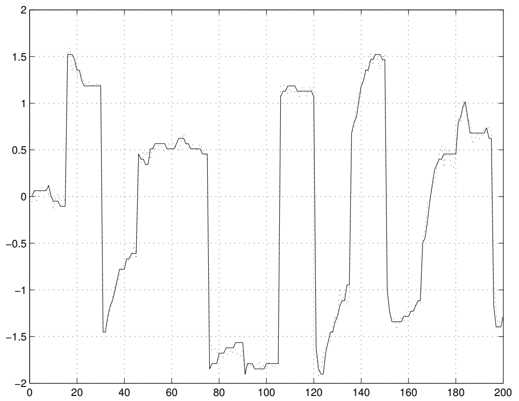

Figure 2.4: Several trials of precision grip experiment. Solid line: original motor information. Dotted Line: reconstructed motor information using the Visual-Motor Map (VMM). The two curves are almost undistinguishable.

It is noticeable that, even inside each grasp class, the variability is very large. This is due to the differences between the grasped objects, and illustrates how the observed features depend not only on the "grasp" type but also on the manipulated object (see Section 3.2.3 for discussion). The dashed-line in the figure shows that the trajectory reconstructed through the neural-VMM is remarkably close to the "true" values. The accuracy of the VMM may degrade when more complex gestures are included.

The use of this map can be represented as in figure 2.5 where it is shown one hand image and after using the map the corresponding reconstruction.

In spite of the encouraging results in this setting, there are a number of remaining drawbacks. The learning method we used does not consider any type of redundancy. For the task at hand this is not a problem because when grasping, humans always choose the same strategy, in such a way that the dataset can be learned as a function.

A final aspect worth mentioning is that the hand-VMM can naturally be learned during an initial phase of development, when the system (natural or artificial) per-

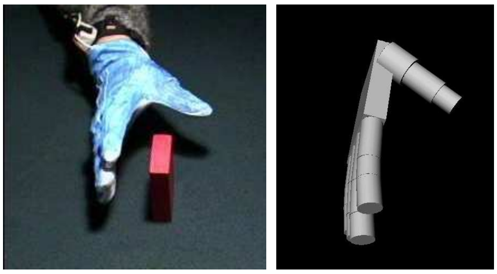

Figure 2.5: Reconstruction results obtained with the VMM

forms hand gestures and observes the (visual) consequences of such gestures. Both proprioceptive (motor) and visual data are present and the association can be established. An additional comment is that self-observation may allow the system to search and tune the most interesting visuo-motor features, such that a more compact representation could be used.

2.3 Static Sensory-motor maps

This section presents an explicit solution for learning partial static sensory-motor maps. The fact that the robotic system is highly redundant for the considered tasks is the reason why we consider the more challenging problem of partial SMMs for redundant robots.

We propose the use of a “Minimum order SMM” that takes the image configuration and redundant dofs, or degrees of redundancy (DOR), as input variables and the non-redundant dofs as outputs. As a consequence, we do not restrict the output of the system at learning time and the extra (redundant) degrees of freedom remain free for on-the-fly selection and for the execution of additional tasks. We have already seen on such example in the “free elbow map” of the previous section, where a degree of freedom was selected with a secondary criteria. Here, the framework is defined in the most general setting.

The most common control method for robots is based on the jacobian matrix [Craig, 1989]. This matrix relates the cartesian velocity of the end-effector with the corresponding joint velocities. To move the robot to a desired position, the inverse jacobian needs to be evaluated. Some other methods can be used to do this inversion when the jacobian is not square [Buss, 2004]. A well known approach is the damped least-squares where the inversion is made jointly with an energy minimization [Wampler, 1986], originally introduced to solve the problem of controlling robots near singularities.

Our approach is different from these approaches in several ways:

- Includes visual information in the loop.

- No knowledge about the system kinematics is needed, the map is learned during a self-exploratory phase.
- The map is global and not a local approximation. This means that, if needed, we can drive the robot directly to a position in an open-loop fashion.
- The learned map is independent of the control law. The same map can be used with different secondary tasks.

In the following sections a statistical learning approach will be presented that follows these lines to learn a partial static SMM.

2.3.1 Minimum order SMM

In this section we show how to define a Sensory-Motor Map that explicitly takes the DOR into consideration, thus allowing the completion of several simultaneous tasks.

Let us define a *SMM* that maps a vector of control variables (n,r) to a vector of image point features $\mathcal{I}$, where n is a minimum set of degrees of freedom that spans the full output space and r is a set of redundant degrees of freedom. Note that there are several partitions of the input space, into redundant versus non-redundant degrees of freedom, that can give this same property. It is possible to find automatically the redundancy by analyzing the correlation matrix for the jacobian estimation [Hosoda and Asada, 2000]. This forward model can thus be written as:

$$\mathcal{I} = f(n, r)$$

and allows to predict the image configuration of the robot given a set of motor commands.

In many cases, we are more interested in the inverse map, i.e. computing the motor commands that drive the robot to a desired image configuration, $\mathcal{I}$. If there were an inverse mapping $(n, r) = f^{-1}(\mathcal{I})$, this problem could be solved in a straight forward manner. However, as the dimension of the input space is larger than that of the output space, there are many input combinations that generate the same image point features. In other words, because of the DOR, $f(n, r)$ is not bijective and, therefore, not invertible.

Fig. 2.2 shows an example of redundancy, where, for this robot the $3D$ position of the wrist is controlled with 4 DOFs, thus remaining one DOR.

To put the problem in another perspective, we can say that finding the robot joint angles to move the arm to a desired image configuration $\mathcal{I}$ is an ill-posed problem when the arm has redundant degrees of freedom, [Tikhonov and Arsenin, 1977], because multiple solutions exist.

One approach to solve ill-posed problems, [Poggio et al., 1985, Bertero et al., 1988], consists in using additional constraints that restrain the set of admissible solutions, in such a way that the solution sought becomes unique in this reduced solution space. In our case, this corresponds to recast the original problem to that of moving the robot to a desired image position $\mathcal{I}^*$ while, at the same time, minimizing some auxiliary criterion, $c(n, r)$.

We built a cost function, l, with two terms: one weighting the error in the position of the end effector (data fitness) and another one corresponding to the weights on the control (regularization term).

$$\mathcal{K}(\mathcal{I}^*, n, r) = \lambda \left\|\mathcal{I} - \mathcal{I}^*\right\|^2 + c(n, r) \tag{2.2}$$

This cost function expresses that we are willing to accept some error in the position if another task can be solved at the same time, in this case control costs. Examples of control cost criteria c can be “Comfort” (e.g. distance to joint limits), Energy minimization (e.g. the position with lower momentum) or Minimum motion (i.e. minimize total motion from current to desired position), posture control, amongst others.

The regularized solution can be found by minimizing the cost defined in Equation (2.2), as follows:

$$(\hat{n}, \hat{r}) = \arg\min_{n,r} \left(\lambda \left\|\mathcal{I} - \mathcal{I}^*\right\|^2 + c(n, r)\right) \tag{2.3}$$

where $\mathcal{I}$ can be computed with the forward model $\mathcal{I} = f(n, r)$. Similarly to [Khatib et al., 2004], this formula integrates two terms: one describing the task part and another related to posture control.

There are two important observations to this formulation. Firstly, the optimization is done with respect to all control variables, which translates into a significant computational cost. Secondly, the DORs are not treated as such, since they undergo exactly the same process as the non-redundant DOFs.

The consequence of this approach is that the extra degrees of freedom are frozen from the beginning and can no longer be used for a different purpose during execution. In a way, redundancy is lost.

Instead, in our approach, we would like to keep the redundant degrees of freedom free for solving additional tasks online. In essence, we split the problem in two steps. Firstly, we define a “Minimal Order Sensory Motor Map”, $g(\mathcal{I}, r)$, that relates n and $(\mathcal{I}, r)$:

$$n = g(\mathcal{I}, r) \tag{2.4}$$

By taking the DORs as input (independent variables) instead of output signals, the problem of computing the non-redundant DOFs becomes well posed. The DORs,

r, are left unconstrained and can be fixed during runtime, when a secondary task or optimization criterion is specified.

The definition of the "Minimum Order SMM" allows us to use the redundancy to meet additional criteria or task-constraints, that can be changed online. The DORs can be determined as the solution of a new optimization problem, with cost function $\mathcal{L}$:

$$\hat{r} \ = \ \arg\min_r \mathcal{L}(\mathcal{I}^*, r) \tag{2.5}$$

The optimization is done with a gradient-descendant method with following update step:

$$r_{t+1} = r_t - \alpha \nabla_r \mathcal{L}(\mathcal{I}, r)$$

Note that, in contrast with the previous case, this optimization is done with respect to the redundant degrees of freedom, only. The optimization complexity is thus substantially lower and lends itself to be used as an online process. In general, the solutions in the two cases are not the same, because different local minima could be reached and the criteria are slightly different.

Our approach guarantees zero prediction error, because the *Minimum Order SMM* allows us to determine the values of n corresponding to the exact image position, for the selected redundant degrees of freedom. This solution is similar to the first (regularized) problem when λ becomes large. If the *Minimum Order SMM* is not exact, then it will introduce some error in the final image configuration.

For clarity, we summarize the final algorithm.

1. Select the desired image configuration, $\mathcal{I}^*$
2. Select and initial motor command (n, r)
3. Select the secondary task optimization criterion
4. Solve the optimization of Equation (2.5) for r and use $g(.)$ to compute n.
5. Move the arm to the obtained solution, (n, r)
6. Observe $\mathcal{I}$ and possibly adjust the function $g(\mathcal{I}, n)$
7. If some extra precision is needed, go to 4

There are several important differences in our approach when compared to other methods based on the robot Jacobian. The *Minimum Order SMM* provides directly the goal position corresponding to the desired redundant joint position. It is then possible to move the robot directly (i.e. in an open-loop fashion) to the goal position, avoiding incremental steps. The posture optimization is done iteratively with the previous update rule. Therefore, the motion goes along the optimization path or directly to the convergence point. This is the case because no visual feedback is necessary to the algorithm. If extra precision is needed, then a visual feedback loop needs to be added.

An example, where the secondary goal is to maintain the control variables as near zero as possible, is presented next:

$$\begin{aligned}\mathcal{L}(\mathcal{I}^*, r) &= \|n\|^2 + \|r\|^2 \\ &= \|g(\mathcal{I}^*, r)\|^2 + \|r\|^2 \qquad (2.6)\end{aligned}$$

Differentiating this cost function yields:

$$\nabla_r \mathcal{L}(\mathcal{I}, r) = 2\left(\frac{\partial g(\mathcal{I}, r)}{\partial r} g(\mathcal{I}, r) + r\right)$$

The derivation of $\frac{\partial g(\mathcal{I},r)}{\partial r}$ is presented in the end of this section.

We have seen how the introduction of the *Minimum Order SMM* allows us to use the system redundancy to solve additional tasks online, as opposed to freezing the DORs in a regularized solution to the initial ill-posed problem. In the next section, we will see how to estimate the *Minimum Order SMM* $g(\mathcal{I}, n)$ online.

2.3.2 Learning the Minimum Order SMM through Local Regression

In the previous section we have seen how to partition the redundant and non-redundant degrees of freedom to build a Minimum Order SMM, $g(\mathcal{I}, r)$ that allows for the computation of the non-redundant dofs leaving the redundant dofs unconstrained. We will now see how such a map can be estimated online. Without loss of generality, let us assume that we want to estimate the following non-linear function:

$$y = f(\mathbf{x}) \qquad (2.7)$$

Since there is little information available about this function, the usual approach consists in approximating $f(\mathbf{x})$ by a set of models that are good local approximations of the original global non-linear function, see [Hastie et al., 2001].

In this work, $f(\mathbf{x})$ will be approximated by a mixture of locally linear models. Obviously, a single linear approximation would fail to provide the necessary degree of fitting accuracy. Each model will have a “confidence” region, called the kernel, and represented by K_j.

$$y = f(\mathbf{x}) \approx \frac{\sum_{j=1}^{M} K_j \mathbf{B}_j^T \mathbf{x}}{\sum_{j=1}^{M} K_j}$$

for some regression matrices, $\mathbf{B}_j$ to be estimated. Several kernel shapes can be defined [Atkeson et al., 1997], leading to different properties of the approximating function. We have adopted a Gaussian kernel with mean μ and variance $\mathbf{W}$:

$$K_j = K_{\mathbf{W_j}}(\mu_\mathbf{j}, \mathbf{x}) = \frac{1}{det(\mathbf{W_j})} e^{-(\mathbf{x}-\mu_\mathbf{j})^T \mathbf{W_j} (\mathbf{x}-\mu_\mathbf{j})} \tag{2.8}$$

Let us assume for the moment that the number and the parameters of each Kernel are known in advance. Each model will be fitted by minimizing the following criteria:

$$\hat{\mathbf{B}} = \arg\min_{\mathbf{B}} \sum_{i=1}^{t} \lambda^{(t-i)} K_i \left\| y_i - \mathbf{B}^T \mathbf{x}_i \right\|^2 \tag{2.9}$$

where K_i weights points according to the kernel measure and λ provides a time forgetting factor.

The model can be estimated by:

$$\hat{\mathbf{B}} = \mathbf{Q}\mathbf{R}^+ \tag{2.10}$$

with

$$\begin{aligned} \mathbf{Q} &= \sum_{i=1}^{t} \lambda^{(t-i)} K_W(\mu, \mathbf{x_i}) y_i^T \mathbf{x_i} \\ \mathbf{R} &= \sum_{i=1}^{t} \lambda^{(t-i)} K_W(\mu, \mathbf{x_i}) \mathbf{x_i}^T \mathbf{x_i} \end{aligned} \tag{2.11}$$

An advantage of writing these terms in this way is the possibility of defining an online estimator:

$$\begin{aligned} \mathbf{Q_t} &= \lambda \mathbf{Q}_{t-1} + K_W(\mu, \mathbf{x_t}) y_t^T x_t \\ \mathbf{R_t} &= \lambda \mathbf{R}_{t-1} + K_W(\mu, \mathbf{x_t}) \mathbf{x_t}^T \mathbf{x_t} \end{aligned} \tag{2.12}$$

Finally, when an input sample is present at runtime, the output will be evaluated as a combination of each model $\mathbf{B_i}$ weighted by K_j:

$$\hat{y} = \frac{\sum_{j=1}^{M} K_j \hat{\mathbf{B}}_j^T \mathbf{x}}{\sum_{j=1}^{M} K_j} \tag{2.13}$$

This function is guaranteed to be C^0 if the kernels are C^0 and have an infinite support, and C^1 if the kernels are differentiable.

The final point to discuss is related to the Kernel functions $K_W(\mu, \mathbf{x})$. How many kernels should be used and what should the parameters of each kernel be ? Usually, the number of kernels is iteratively increased during training. If the distance between a new data sample and its nearest kernel exceeds a certain threshold, a new kernel is created with center (μ) in this point. The shape of the kernels (the covariance matrix) can be automatically updated choosing a measure of reconstruction quality or others [Schaal and Atkeson, 1998, Hastie and Loader, 1993].

This is a standard formulation and several improvements have already been proposed. The *Locally Weighted Projection Regression* method, proposed in [Vijayakumar and Schaal., 2000], is linear with the number of samples and every new sample can be added easily. As the method is not capable of extrapolating, the work space must be well covered in the training set. Other implementations keep several samples in memory without estimating any explicit models. The prediction is produced online by weighting the points in memory with some kernel functions.

2.3.3 Experimental Results

We have planned several experiments with a real robot to assess the quality of the algorithms and the ability to learn the proposed SMM online. The experimental setup was the Baltazar humanoid robot torso [Lopes et al., 2004], consisting of a 4 dof head, a 6 dof arm and a 10 dof under-actuated hand. The results shown here report to a fixed position of the head and the use of 4 dof of the arm. The image features consist on the image position of the wrist.

This task requires the use of two degrees of freedom only. The non-redundant dofs are the shoulder adduction/abduction and flexion/extension, while shoulder axis rotation and elbow flexion/extension are considered as redundant degrees of freedom.

Quality evaluation of local learning method

The first experiment is designed to validate the local learning algorithm of the Minimum order SMM. The map to learn will associate the image position of the robot hand and the redundant degrees of freedom to the non-redundant dofs.

During learning, the head remains in a fixed position observing the robot hand. The arm moved to several randomly selected positions. The range of movement of each joint is in the order of $0.55\,rad$. After the robot wrist attains one such position, it remains fixed while different solution for the inverse kinematics are used. This is possible due to the redundancy of this robot's kinematics, when four degrees of freedom are used for position. For more details, see [Lopes et al., 2004] and Figure 2.2. Another redundancy exists because, with the use of a single camera, depth information is lost. Hence, for a 2 dof task we have 4 dof available.

Figure 2.6 shows the evolution of joint angles during this period of auto-observation and learning, which basically represents the dataset used for learning. It also shows the image position of the robot wrist where, due to elasticity in the robot joints, oscillations occur when the acceleration is large.

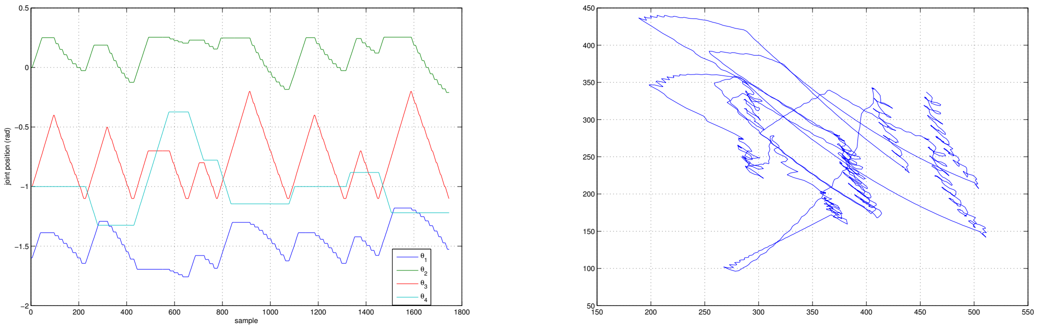

Figure 2.6: Dataset for the real robot experiments. Left: temporal evolution of the joint angles. Right: corresponding image trajectories. The oscillation is caused by elasticity in the robot joints.

Figure 2.7 shows the quality of the SMM estimation, as described in Section 2.3.2. The top plots show the true and estimated non-redundant joint angles, which are in good agreement. The histogram and cumulative distribution of the error are shown in the bottom plots, for 2000 data points. For both non-redundant joints less than 10% of the points have an error bigger than $0.05\,rad$.

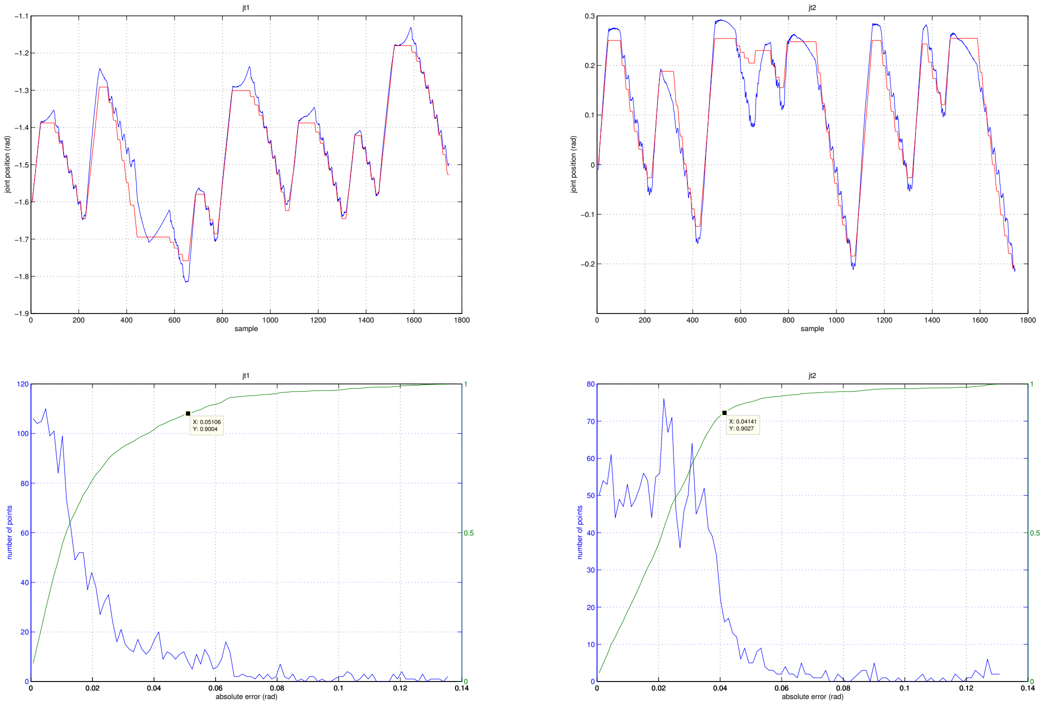

Figure 2.7: The top figures show the prediction error for (non-redundant) Joint 1 (leftt) and Joint 2 (right). The absolute error histogram and the cumulative distribution are shown in the bottom figure. Almost 90% of the samples have an error below $0.05\,rad$

These results show that the online estimation method presented in Section 2.3.2 provides a good approximation to the original Minimum order SMM. The next set of experiments show how to define a secondary task, based on an energy minimization criterion, to drive the robot to the desired position, while meeting this secondary goal.

Sensory-motor coordination with redundant dof

For a given desired image position and an initial position of the redundant degrees of freedom, our goal is to reach a certain image position, while satisfying a secondary criterion (task). This is obtained through the following optimization problem, as defined earlier:

$$l = \|\theta_{\mathbf{n}} - \mu_{\mathbf{n}}\|^2 + \|\theta_{\mathbf{r}} - \mu_{\mathbf{r}}\|^2$$

that aims to maximize the distance to joint limits, corresponding to a comfort criterion.

The optimization process relies on the estimated Minimum order SMM, as described before. Figure 2.8 presents the evolution of the cost function, l for each iteration of the Newton method. It also presents the trajectory of all 4 (redundant and non-redundant) robot joints. We can see that, for this case, the maximum for one joint was $0.5\,rad$. Most of the error is due to elasticity in the robot joints.

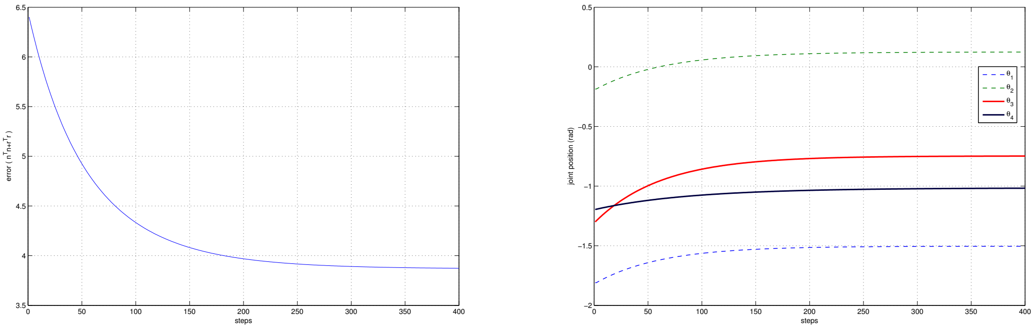

Figure 2.8: Convergence rate (left) and evolution of the position (right) for the real robot as a function of the optimization step. It is interesting to see that one joint moved $0.5\,rad$ and the final error in the image corresponds to $0.03\,rad$.

The final error in the image was as small as $0.023\,rad$. Figure 2.9 shows the robot view of the hand for an intuition for this error, about the size of the target. Due to the redundancy in the arm, it would be possible to fixate the target while changing the arm posture.

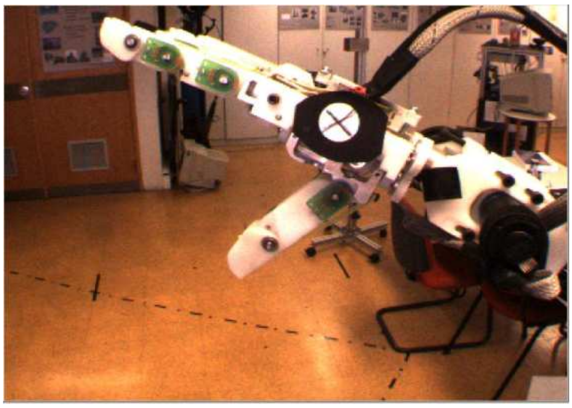

Figure 2.9: Robot view of the hand. We can see the arm, hand and the target being tracked.

Derivative of Local learning method

If redundant degrees of freedom are to be chosen with some extra criterion, it is important to evaluate the derivative of the prediction function. The prediction of the chosen local learning method is given by:

$$\hat{y} = \frac{\sum_{j=1}^{M} K_j {\mathbf{B_j}}^T \mathbf{x}}{\sum_{j=1}^{M} K_j}$$

Now we want to evaluate its derivative as a function of the inputs $\frac{\partial \hat{y}}{\partial \mathbf{x}}$:

$$\frac{\partial \hat{y}}{\partial \mathbf{x}} = \sum_{j=1}^{M} \left(\frac{\partial K_j}{\partial \mathbf{x}} \mathbf{x}^T + K_j \right) \frac{\mathbf{B_j}}{\tilde{k}} - \frac{1}{\tilde{k}^2} \sum_{l=1}^{M} \frac{\partial K_l}{\partial \mathbf{x}} \sum_{j=1}^{M} K_j \mathbf{x}^T \beta$$

with $\tilde{k} = \sum_{j=1}^{M} K_j$ and $\frac{dK_j}{d\mathbf{x}} = -W_j(\mathbf{x} - \mu_{\mathbf{j}})K_j$

After some computations, we have:

$$\tilde{k}\frac{\partial \hat{y}}{\partial x} = \sum_{j=1}^{M} \frac{\partial K_j}{\partial x} x^T \mathbf{B_j} + \sum_{j=1}^{M} \mathbf{B_j} K_j - \tilde{dk}\ \hat{y}^T \tag{2.14}$$

with $\tilde{dk} = \sum_{j=1}^{M} \frac{\partial K_j}{\partial \mathbf{x}}$

2.3.4 Conclusions

We have addressed the problem of estimating Sensory Motor Maps in redundant systems, which is often the case of humanoid robots. Because of redundancy, the inverse map cannot be estimated since the forward model is not bijective.

For a given task, we started by partitioning the robot degrees of freedom in redundant and non-redundant dofs. Then, we defined a “Minimum order SMM” that takes as the input general image configurations and redundant degrees of freedom.

In this way, this partial backward model is well defined and can be used to determine the configuration for the non-redundant degrees of freedom.

Using the "Minimum order SMM", the redundant degrees of freedom are available to meet additional online constraints, arising from secondary tasks or criteria. We presented an optimization framework to choose redundant degrees of freedom for several example secondary tasks.

The Minimum Order SMM is learned with a local learning method. Experimental results done in an humanoid torso with 10 degrees of freedom were presented, illustrating both the ability to learn the *Minimum order SMM* and how it can be used for specific tasks.

A large workspace was used of about $40\,degrees$ for each joint, a small reconstruction error was achieved, about $2.5\,degrees$. For the optimization one joint could "travel" $90\,degrees$ to be able to reduce the cost function by 40%, with a corresponding error in the image of only $1.5\,degrees$ (about the same size of the target).

2.4 Incremental Sensory-Motor Maps[1]

The previous approach is best suited for static situations, i.e absolute positioning tasks, and not for real-time control. In this section we are going to present solutions that are able to relate desired sensory incremental changes to the corresponding control.

Image-based visual servoing methods provide very efficient and robust solutions to control robot motions from an initial position to a precise goal defined in image space [Hutchinson et al., 1996]. It provides high accuracy for the final pose and good robustness to noise in image processing, camera calibration and other setting parameters.

The redundancy formalism [Rosen, 1960, Liegeois, 1977, Marchand and Hager, 1998] extends the task-function approach [Samson et al., 1991] to compute a control law that realizes a main task, while simultaneously taking supplementary constraints into account. This approach is particularly relevant for redundant robots, since it can be used when the vision-based task does not constrain all the robot's degrees of freedom (DOF). A secondary task can then be added to meet another objective, without disturbing higher priority tasks. The control law for the second task is projected into the set of motions constituting the null space of the first task, thus leaving the first tasks unmodified. The computation of linear projection operator is based on the jacobian of the first task.

[1] Work done in collaboration with Nicolas Mansard and François Chaumette, INRIA, Rennes, France

2.4.1 Visual Servoing

The goal of visual servoing is to control robots with information acquired solely by a camera. Considering the model:

$$\mathcal{I} = \mathbf{P}\mathbf{T}(\theta)\dot{\theta}$$

where the image features $\mathcal{I}$ are task dependent, a transformation dependent on the control variables $\mathbf{T}(\theta)$ and the camera projection model $\mathbf{P}$. The question is now to determine what are the joint positions θ corresponding to some desired features $\mathcal{I}^*$. The problem has several solutions, which are not trivial to determine. The servoing controller will move the camera in a gradient like method to achieve the desired position. Let us consider the image derivative:

$$\dot{\mathcal{I}} = \mathbf{J}\dot{\theta}$$

where $\mathbf{J}$ is called the jacobian matrix or, in other words, the incremental Sensory-Motor Map. In the remaining of this chapter we will refer to the term Jacobian that is regularly used in the visual servoing literature, instead of incremental SMM.

$$\dot{\theta} = -\mathbf{J}^{+}\dot{\mathcal{I}}$$

Two main different architectures for servoing can be adopted: eye-in-hand and eye-to-hand. The situation when the robot is observed externally and the goal is to move the robot to a desired position, is called eye-to-hand. The justification is that the eye observes the hand and controls it, as when primates control their hands to grasp objects. Other situation is when the camera is itself moving and the desired features are related to an external object. This is the eye-in-hand configuration, because the eye is moving with the manipulator, it is similar when humans move the head to see an object in a different perspective. We then have,

Eye-in-hand:

$$\mathcal{I} = \mathbf{P}^C_R\mathbf{T}^R_W\mathbf{T}(\theta_R)^W\mathbf{F}$$

Eye-to-hand:

$$\mathcal{I} = \mathbf{P}^C_W\mathbf{T}^W_R\mathbf{T}(\theta_R)^R\mathbf{F}$$

In the previous equations, we considered three reference frames, the world, W; the robot R and camera C, with ${}^I_J\mathbf{T}$ being a coordinate transformation from frame J to frame I. Usually, a distinction is made between the interaction matrix $\mathbf{L}$ and the robot jacobian $\mathbf{J}$.

$$\dot{\mathcal{I}} = \mathbf{L}(\theta)\mathbf{J}(\theta)\dot{\theta}$$

with most works dealing just with the interaction matrix, doing the control in camera coordinates $\mathbf{V} = J(\theta)\dot{\theta}$.

$$\dot{\mathcal{I}} = \mathbf{L}(\theta)\mathbf{V}$$

2.4.2 Jacobian Estimation Methods

All visual control methods involve the computation of the task jacobian, linking the evolution of the visual features to the robot articular motion. It thus requires knowledge about the camera-world and world-actuator transformations that influence the interaction matrix (relating image and camera velocities) and the robot jacobian (relating end-effector and joint velocities). Such transformations are usually obtained during an offline calibration phase.

However, full system calibration (and even a coarse one) is not always possible and/or desirable. Some robots may lack proprioceptive sensors to do that and some parameters may vary over time, due to malfunction, changes in mechanical parts or modification in camera lenses. For instance, in [Marchand et al., 2001] a submarine robot is described that has no position sensors due to technological and cost issues. Nevertheless, a visual servoing task was accomplished without the use of proprioception. Even when calibration information is available, the analytic computation of the interaction matrix often requires an estimate of the depth of the tracked features. When a single camera is used, depth estimation requires knowledge about the object model. For all these reasons, robot calibration can be very difficult or even impossible in practice.

Several methods have already been proposed to estimate the interaction matrix, the robot jacobian or the task jacobian. One of the first works was [Hosoda and Asada, 1994], where robust learning rule was derived and a convergence proof given. This method is based on the Broyden update rule, well known from optimization theory [Fletcher, 1987] and has been widely used in real robotic applications with visual control. In [Jagersand and Nelson, 1996], it was used for visually-guided grasping. Adhoc task sequencing was used in [Dodds et al., 1999] to separate the reaching phase and the grasping phase. Another object-grasping task is presented in [Lopes et al., 2005], where the estimation algorithm is used to provide an approximation of a highly non-linear mapping, using several local linear models. For the eye-to-hand configuration, with a moving target, an error function for the jacobian approximation is defined. By minimizing this function with a Newton method [Piepmeier et al., 1999], a time-varying system is obtained. A later work by the same author applies the same scheme to an eye-in-hand system, also with moving targets [Piepmeier et al., 2002].

It was also suggested to learn the inverse jacobian directly, instead of the jacobian [Lapresté et al., 2004], although only an offline formulation was proposed in that work. Experimental results show faster convergence rates, when using the *Direct-inverse.* A very complete discussion about adaptive identification methods for slowly varying parameters is presented in [de Mathelin and Lozano, 1999]. The same work presents a new method to improve the robustness of parameter identification, by combining directions with new information with those where the information had been lost and had to be recovered.

A large amount of information is required to recompute the jacobian at each iteration. As presented in Section 2.4.1, the jacobians can be divided in two parts: the articular jacobian $\mathbf{J}_\theta$, and the interaction matrix $\mathbf{L}_\mathbf{s}$ (see (2.27)). The articular jacobian can only be computed if the full arm-eye calibration is available. Instead, the interaction matrices require the 3D parameters of the target object plane. This can be estimated using pose computation (if the object model is available) or using a homography [Faugeras, 1993] between the image and relevant scene planes. In this last case, the scale parameters cannot be estimated, and the object depth has to be fixed *a priori* or estimated using other solutions. All these parameters can lead to errors in the computation of the interaction matrix.

Different approaches can be used to estimate the tasks jacobians. Learning the task jacobian, $\mathbf{J_i}$, is quite difficult, due to numerous non-linear parameters involved in the true analytic equation. The experiments have proved the extreme difficulty in obtaining a good estimation of $\mathbf{J_i}$. In [Lopes et al., 2005], a mixture of several linear models was described to tackle the problem.

Instead of trying to estimate $\mathbf{J_i}$, we propose to use an approximation of the articular jacobian, $\widehat{\mathbf{J}_\theta}$, computed from gross robot calibration data. Then, since an approximation of $\mathbf{v}$ can be computed, only the matrices $\widehat{\mathbf{L}_{\mathbf{s_i}}}$ need to be estimated.

$$\begin{aligned}\hat{\mathbf{v}} &= \widehat{\mathbf{J}_\theta}\dot{\theta}\\ \dot{\mathbf{e}}_\mathbf{i} &= \widehat{\mathbf{L}_{\mathbf{s_i}}}\hat{\mathbf{v}}\end{aligned}$$

The major uncertainty lies in the robot model $\mathbf{J}_\theta$. Therefore, it might seem illogical to learn $\mathbf{L}_{\mathbf{s_i}}$, since the analytical computation would not be too difficult and it requires less information, when compared to $\mathbf{J_i}$. However, estimating $\mathbf{L}_{\mathbf{s_i}}$ will allow to "absorb" some of the errors in $\mathbf{J}_\theta$, caused by the coarse robot calibration. This solution is also able to take into account the uncertainties in the target model, yielding better results, even when using a properly calibrated industrial robot, as it will be shown in section 2.4.4. In the following sub-sections, we present the methods used in the experiments to learn the jacobian.

Broyden Update

The first work presented for jacobian estimation in visual servoing was [Hosoda and Asada, 1994]. This method is based on the Broyden update rule. The

Jacobian estimation update rule is given by:

$$\hat{\mathbf{J}}(t+1) = \hat{\mathbf{J}}(t) + \alpha \frac{\left(\Delta\mathbf{e} - \hat{\mathbf{J}}(t)\Delta\mathbf{x}\right)\Delta\mathbf{x}^\top}{\Delta\mathbf{x}^\top\Delta\mathbf{x}} \tag{2.15}$$

After observing some image motion, $\Delta\mathbf{e}$, caused by a motor command $\Delta\mathbf{x}$, the Jacobian is updated directly, with α defining the update speed. This method has several positive aspects: low memory usage because only the last observation is used; low computational cost and a single parameter to be tuned. When the motions are too small, this computation may become unstable. One solution consists in including a regularization term in the denominator to prevent singularities. Alternatively, the learning can simply be switched off, whenever the motion falls below a certain threshold.

Correlation

A different approach can be made with least squares estimation [de Mathelin and Lozano, 1999]. Considering the cost function l as:

$$l = \sum_{i=0}^{t} \gamma^{i-t}(\Delta\mathbf{e} - \mathbf{J}\Delta\mathbf{x})^\top(\Delta\mathbf{e} - \mathbf{J}\Delta\mathbf{x})$$

Representing the prediction error with a time decay term γ. The minimization will give the usual least squares solution:

$$\mathbf{J} = \mathbf{Q}\mathbf{R}^+$$

where $\mathbf{Q}$ and $\mathbf{R}$ are:

$$\begin{aligned}\mathbf{Q} &= \sum_{i=0}^{t} \gamma^{(t-i)} \mathbf{\Delta e}_i^\top \mathbf{\Delta x}_i \\ \mathbf{R} &= \sum_{i=0}^{t} \gamma^{(t-i)} \mathbf{\Delta x}_i^\top \mathbf{\Delta x}_i\end{aligned}$$

or in an online formulation;

$$\begin{aligned}\mathbf{Q} &= \gamma\mathbf{Q} + \mathbf{\Delta e}_t^\top \mathbf{\Delta x}_t \\ \mathbf{R} &= \gamma\mathbf{R} + \mathbf{\Delta x}_t^\top \mathbf{\Delta x}_t\end{aligned}$$

Direct-Inverse

Learning the jacobian boils down to minimizing the prediction error of the image velocities. Yet, for control purposes, we need the inverse map $\mathbf{J}^+$, that corresponds to the “reconstruction” of the robot joint velocities from image velocities. Thus, in

[Lapresté et al., 2004], it was suggested that one should learn the inverse Jacobian directly, instead of the Jacobian. The main motivation becomes from the servoing control law, where the goal consists in finding which motor command produces the desired motion in the image. The cost function becomes:

$$l = \sum_{i=0}^{t} \gamma^{i-t} (\Delta\mathbf{x} - \mathbf{H}\Delta\mathbf{e})^\top (\Delta\mathbf{x} - \mathbf{H}\Delta\mathbf{e})$$

where $\mathbf{H} = \mathbf{J}^+$. This cost function can be seen as a reconstruction error, as opposed to the prediction error used before. The main advantage is that we no longer need to invert the Jacobian for computing the control law. An additional benefit is that the least-squares fitting requires the inversion of a smaller, possibly better conditioned, information matrix, R. This is particularly relevant for the task sequencing approach whereby the (sub-)task dimension is much smaller than the control space.

2.4.3 Visual Control with Redundancy

It is well known that visual servoing and closed-loop control offer a high level of robustness to jacobian errors. As a consequence, learning the jacobian usually provides very good results, keeping a good convergence rate and overcoming the need for analytic computations.

If, as we have seen before, the available degrees of freedom are in excess with respect to the task to be performed, the inverse Jacobian cannot be determined. One way to overcome this problem, when computing the conptrol law, is to rely on the so-called redundant formalism, [Siciliano and Slotine, 1991, Baerlocher and Boulic, 2004]. The key idea consists in breaking a complex servoing task down to a sequence of simpler, redundant tasks, that are added to a stack when the robot approaches the goal. Each task in the sequence is projected in the null space of preceding ones. At each step, the robot moves to fulfill a redundant task, maintaining all the elementary tasks already completed. During this motion, the already completed tasks should remain unchanged.

In [Mansard and Chaumette, 2005] present a solution that generalizes the the redundancy formalism to several tasks. Redundant tasks are stacked on top of the others, until all degrees of freedom of the robot are constrained, and the desired position is reached.

Here, we follow a similar approach but, in addition, we assume that the system is either coarsely calibrated or totally uncalibrated. It is important to notice that errors in the Jacobian estimate can cause errors in control law, as well as in the computation of the projection operators, disturbing higher priority tasks.

To overcome this problem, we use the disturbance caused in a high priority task, when a new one starts, as an error signal to improve the jacobian online. We compare several estimation methods, in order to access their quality in terms of real-parameter estimation and online behavior of the system.

Our results show that it is possible to estimate the jacobian online, in the context of task sequencing for visual servoing. In addition, the learning stability is greatly improved by the task sequencing approach. Hence, task sequencing and learning are intertwined, mutually constrained processes that bring additional performance and flexibility to the control of complex robotic systems, when calibration information is unavailable or highly uncertain.

Redundancy formalism for two tasks

In this section, we recall how to sequence redundant tasks and to maintain the tasks already achieved [Mansard and Chaumette, 2004]. We start by presenting the redundancy formalism [Samson et al., 1991]. It has first been used for visual servoing in [Espiau et al., 1992], and in numerous applications since (e.g. avoiding visual occlusions [Marchand and Hager, 1998], or human-machine cooperation using vision control [Hager, 2002]).

The redundancy formalism we can compute a control law to keep the priority order defined by the stack of tasks, and ensuring the continuity of the robot velocity when the stack changes. This control law can be implemented for various kinds of closed-loop control, provided that the objective can be written as a task function [Samson et al., 1991]. In this thesis, we have used it for visual servoing. Section 2.4.3 gives the selected visual features.

Let θ be the articular vector of the robot. Let $\mathbf{e_1}$ and $\mathbf{e_2}$ be two tasks, $\mathbf{J_i} = \frac{\partial \mathbf{e_i}}{\partial \theta}$ $(i = 1, 2)$ their jacobian, defined by:

$$\dot{\mathbf{e_i}} = \frac{\partial \mathbf{e_i}}{\partial \theta}\dot{\theta} = \mathbf{J_i}\dot{\theta} \tag{2.16}$$

Since the robot is controlled using its articular velocity $\dot{\theta}$, (2.16) has to be inverted. The general solution (with $i = 1$) is:

$$\dot{\theta} = \mathbf{J_1^+}\dot{\mathbf{e_1}} + \mathbf{P_1}\mathbf{z} \tag{2.17}$$

where $\mathbf{P_1}$ is the orthogonal projection operator on the null space of $\mathbf{J_1}$ and $\mathbf{J_1^+}$ is the pseudo-inverse (or least-squares inverse) of $\mathbf{J_1}$. Vector $\mathbf{z}$ can be used to apply a secondary command, that will not disturb $\mathbf{e_1}$. Here, $\mathbf{z}$ is used to carry out at best the task $\mathbf{e_2}$. Introducing (2.17) in (2.16) (with $i = 2$) gives:

$$\dot{\mathbf{e_2}} = \mathbf{J_2}\mathbf{J_1^+}\dot{\mathbf{e_1}} + \mathbf{J_2}\mathbf{P_1}\mathbf{z} \tag{2.18}$$

By inversing this last equation, and introducing the computed $\mathbf{z}$ in (2.17), we finally get:

$$\dot{\theta} = \mathbf{J_1^+}\dot{\mathbf{e_1}} + \mathbf{P_1}(\mathbf{J_2}\mathbf{P_1})^+(\dot{\mathbf{e_2}} - \mathbf{J_2}\mathbf{J_1^+}\dot{\mathbf{e_1}}) \tag{2.19}$$

Since $\mathbf{P_1}$ is Hermitian and idempotent (it is a projection operator), (2.19) can be written:

$$\dot{\theta} = \mathbf{J_1^+ \dot{e_1}} + \widetilde{\mathbf{J_2}}^+ \widetilde{\dot{\mathbf{e_2}}} \tag{2.20}$$

where $\widetilde{\mathbf{J_2}} = \mathbf{J_2 P_1}$ is the limited jacobian of the task $\mathbf{e_2}$, giving the available range for the secondary task to be performed without affecting the first task, and $\widetilde{\dot{\mathbf{e_2}}} = \dot{\mathbf{e_2}} - \mathbf{J_2 J_1^+ \dot{e_1}}$ is the secondary task function, after subtracting the part $\mathbf{J_2 J_1^+ \dot{e_1}}$ already accomplished by the first task. A very good intuitive explanation of this equation is given in [Baerlocher and Boulic, 2004].

Extending redundancy formalism for several tasks

Let $(\mathbf{e_1}, \mathbf{J_1})$ ... $(\mathbf{e_n}, \mathbf{J_n})$ be n tasks so that Task $\mathbf{e_i}$ should not disturb task $\mathbf{e_j}$ if $i > j$. A recursive extension of (2.20) is proposed in [Siciliano and Slotine, 1991]:

$$\dot{\theta}_\mathbf{i} = \dot{\theta}_\mathbf{i-1} + (\mathbf{J_i P_{i-1}^A})^+ (\dot{\mathbf{e}}_\mathbf{i} - \mathbf{J_i} \dot{\theta}_\mathbf{i-1}) \tag{2.21}$$

where $\mathbf{P_i^A}$ is the projector onto the null-space of the augmented Jacobian $\mathbf{J_i^A} = (\mathbf{J_1}, \ldots \mathbf{J_i})$. The recursion is initialized by $\dot{\theta}_\mathbf{0} = 0$. The robot velocity is $\dot{\theta} = \dot{\theta}_\mathbf{n}$.

Using this recursive equation directly, a projector has to be computed on each step of the computation. A recursive formula for the computation of the projector is proposed in [Baerlocher and Boulic, 2004]. We recall this equation here

$$\mathbf{P_i^A} = \mathbf{P_{i-1}^A} - \widetilde{\mathbf{J_i}}^+ \widetilde{\mathbf{J_i}} \tag{2.22}$$

where $\widetilde{\mathbf{J_i}} = \mathbf{J_i P_{i-1}^A}$ is the limited jacobian of the task i. The recursion is initialized by $\mathbf{P_0^A} = \mathbf{I}$ (identity matrix).

Computing the control law

Let $(\mathbf{e_1}, \ldots, \mathbf{e_n})$ be a stack of n tasks. The convergence speed of each task can be chosen separately by using

$$\dot{\mathbf{e}} = \begin{bmatrix} \dot{\mathbf{e_1}} \\ \vdots \\ \dot{\mathbf{e_n}} \end{bmatrix} = - \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix} \begin{bmatrix} \mathbf{e_1} \\ \vdots \\ \mathbf{e_n} \end{bmatrix} = -\Lambda \mathbf{e} \tag{2.23}$$

The control law computed from equations (2.21) and (2.23) would ensure the priority order fixed by the stack along with an exponential decreasing of the errors of the tasks in the stack when respecting the priority. However, each time the stack changes (because a task is added or removed) such a law will lead to a break of continuity. To solve this problem, it has been proposed in [Soueres et al., 2002, Mansard and Chaumette, 2004] to use a non-homogeneous first order differential equation instead of (2.23):

$$\dot{\mathbf{e}} = -\lambda \mathbf{e} + e^{-\mu(t-\tau)} \cdot (\dot{\mathbf{e}}'(\tau) + \lambda \mathbf{e}(\tau)) \tag{2.24}$$

where τ is the time of the last change of the stack, $\mathbf{e}'$ is the stack error before the last change, and μ is a parameter used to set the length of the transient time. Using equations (2.21) and (2.24), the complete expression of the control law is thus:

$$\begin{cases} \dot{\theta}_{\mathbf{i}} = \dot{\theta}_{\mathbf{i-1}} + (\mathbf{J_i P^A_{i-1}})^{+}(-\lambda_i \mathbf{e_i} - \mathbf{J_i}\dot{\theta}_{\mathbf{i-1}}) \\ \dot{\theta} = \dot{\theta}_{\mathbf{n}} + e^{-\mu\cdot(t-\tau)} \cdot (\dot{\mathbf{e}}'(\tau) + \Lambda \mathbf{e}(\tau)) \end{cases} \tag{2.25}$$

In this final control law, two matrices have to be learn: $\mathbf{J_i}$ and $\mathbf{P^A_i}$. Since $\mathbf{P^A_i}$ can be computed from $\mathbf{J_i}$, we will learn only the jacobian and then compute the projection operator from it. In this sense, the two matrices are learned. The effect of the learning can be considered from two different points of view: convergence of a task while the jacobian is learned online and the disturbances caused upon higher priority tasks due to the effect of learning the projection operator.

Application to visual servoing

In this thesis, we propose an implementation of this control law using vision control. The task functions $\mathbf{e_i}$ used in the remainder of the text are computed from the visual features [Espiau et al., 1992]:

$$\mathbf{e_i} = \mathbf{s_i} - \mathbf{s_i^*} \tag{2.26}$$

where $\mathbf{s_i}$ is the current value of the visual features for task $\mathbf{e_i}$ and $\mathbf{s_i^*}$ their desired value.

The interaction matrix $\mathbf{L_{s_i}}$ related to $\mathbf{s_i}$ is defined so that $\dot{\mathbf{s_i}} = \mathbf{L_{s_i}v}$, where $\mathbf{v}$ is the camera kinematic screw. From (2.26), it is clear that the interaction matrix $\mathbf{L_{s_i}}$ and the task jacobian $\mathbf{J_i}$ are linked by the relation:

$$\mathbf{J_i} = \mathbf{L_{s_i}J_q} \tag{2.27}$$

where the matrix $\mathbf{J_q}$ denotes the robot jacobian ($\mathbf{v} = \mathbf{J_q}\dot{\theta}$). In the following section we will see several methods that can be used to learn the jacobian matrix.

2.4.4 Experiments and results

In this section we present results comparing the quality of task execution using several methods of jacobian estimation. Two robots with different kinematics and servoing architectures were used for the experiments. We first describe quickly the selected visual features used for the experiments. Four representative experiments are then presented in detailed.

Visual features for vision-based control

In order to have a better and easier control over the robot trajectory, approximately decoupled tasks were chosen. As explained before, the tasks do not need to be perfectly independent, thanks to the redundancy formalism. We have used visual features derived from the image moments. Let $P_i = (x_i, y_i)$ be the position of a set of points in the image. The moments $m_{i,j}$ are defined by

$$m_{i,j} = \sum_{k=1}^{N} x_k^i y_k^j \tag{2.28}$$

To simplify the image processing as we mainly focus on the control part, we have used a simple white-points-on-black-board target as shown Fig 2.10. The first task $\mathbf{e_g}$ - *centering* - is based on the position of the center of gravity of the for points

$$(x_g, y_g) = (\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}) \tag{2.29}$$

The second task $\mathbf{e_Z}$ - *zooming* - uses the area of the object in the image to control the range between the robot and the target. The third task $\mathbf{e}_\alpha$ - *rotation* - rotates the camera around the optical axis, so that the object will be correctly oriented in the image. It uses the orientation of the object in the image, which can be obtained from the second order moments [Chaumette, 2004]. The last task $\mathbf{e_R}$ - *perspective correction* - uses third order moments to decouple v_x from ω_y and v_y from ω_x. The reader is invited to refer to [Chaumette, 2004] for more details.

Results

The three first experiments were realized with the robot Baltazar. Baltazar is an anthropomorphic robotic torso [Lopes et al., 2004] equipped with a six DOF arm, an eleven DOF hand and a four DOF head. In the presented experiments, the target was attached to the robot hand. A eye-to-hand visual servo was used to position the hand parallel to the eye image plane, centered at a distance of $20cm$. This robot has a high payload/weight ratio causing some elasticities, its motors are equipped with position sensors but the lack of an home sensor causes some errors if a precise calibration is needed. The camera is coarsely calibrated and for the experiments a $4.5mm$ lens had to be used. Figure 2.10 presents the robot and the initial and final hand position. The total motion is about $30cm$ corresponding to a maximum joint translation of $90\,dg$.

The first experiment includes a comparison of the estimation methods presented in Section 2.4.2. The second experiment shows that it is possible to correct the robot (coarse) calibration by learning only the interaction matrix when realizing a full sequencing. The third experiment is a first step to show (experimentally) that the trajectory obtained when applying a sequencing control law provides a very good dataset for learning. The last experiment was realized on an accurately calibrated robot (see Section 2.4.4). It shows that a small uncertainty can result in a big perturbation, and that the online estimation is able to provide a nearly perfect behavior.

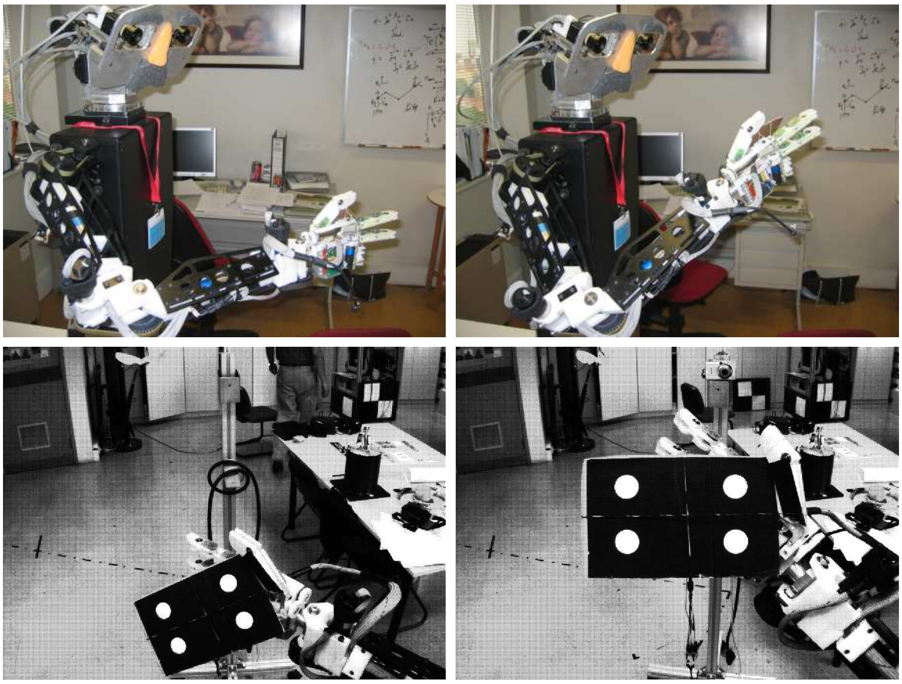

Figure 2.10: Initial and final position of the sequential visual servoing task. Top: outside view, Down: camera view.

Experiment 1 - online learning In this first experiment, the stack has two tasks: centering ($\mathbf{e_g}$) and Z-rotation ($\mathbf{e}_\alpha$). The goal consisted in testing how the jacobian estimation errors would influence the task sequencing approach, due to errors introduced in the projection operators. When the jacobian of the first task is mis-estimated, the centering is lost with the activation of the second task. When the error increases, the target moves further away from the image center, and can leave the image if the disturbance is too strong (which results of course in the visual servoing failure).

Figure 2.11 presents the evolution of the error for the first task using analytic/offline learning versus online estimation methods. Figure 2.12 shows the result for the second (rotation) task. Offline learning relies on simple motions of the arm, done during approximately 250 iterations. Online learning was carried out at every frame.

The first result shows that analytic or offline learning are worse, in terms of having a larger perturbation and longer convergence times. Instead, online estimation methods lead to much better results, outperforming the results with the analytical jacobian. Although a large disturbance appears when the second task is added, it is quickly reduced afterwards.

The amplitude of the perturbation ranged from 20 to 30 pixels. *Broyden* and *Correlation* methods were able to eliminate the error after 30 iterations. The maximal perturbation is equivalent to the one obtained with analytic computation,

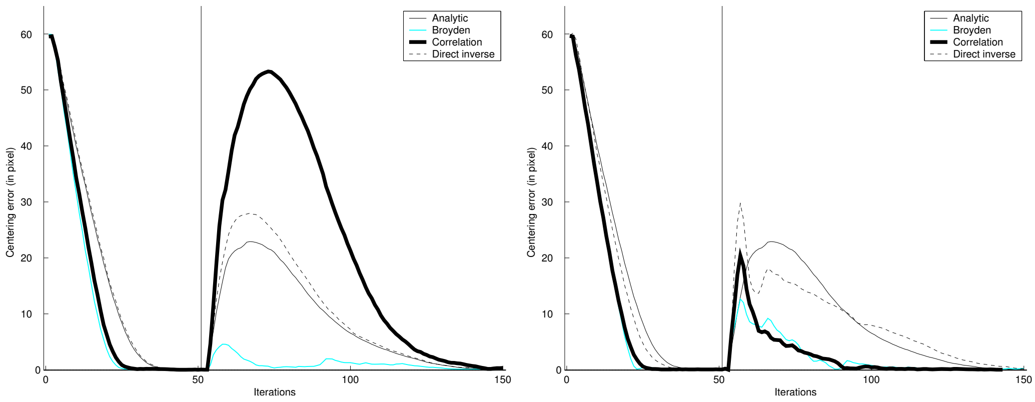

Figure 2.11: Temporal evolution of the image error during servoing using offline (left) or online (right) learning methods. The vertical line shows the time instant where the second task started.

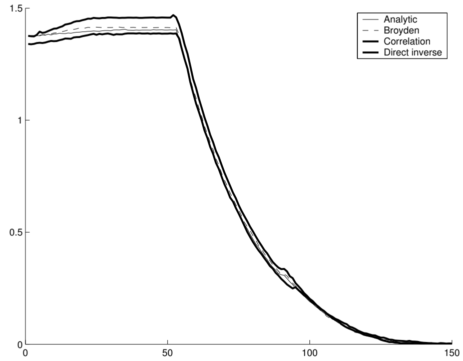

Figure 2.12: Convergence rates for the second task (rotation) while trying to keep the centering at zero (refer to Figure 2.11). Due to robustness to jacobian error, the convergences are the same for all the tasks.

but the duration is much shorter.

The *Direct-inverse* method was unable to reduce the error as fast as the two other methods or the analytic version, whatever the tuning realized. The advantage of directly minimizing the reconstruction errors instead of the prediction error does not appear significant in this setting. Indeed, to compute the projection it is necessary to have the direct map and the result, in the end, is worst.

It is also interesting to see that the task-error convergence is very similar for all methods (for Task $\mathbf{e_g}$, Fig. 2.11 before Iteration 50, and for Task $\mathbf{e}_\alpha$, Fig. 2.12). This emphazises that the reduction of the perturbation is not made at the cost of worse convergence. The convergence is very robust to jacobian error, since all the task convergences are the same. It is nevertheless not true for the projection operator estimation, which is very sensitive and requires an accurate estimation.

All online learning methods succeeded to solve the task. From several experiments, starting from different initial positions and using different tasks, the *Correlation* method produced better results in sense of perturbation amplitude, pertur-

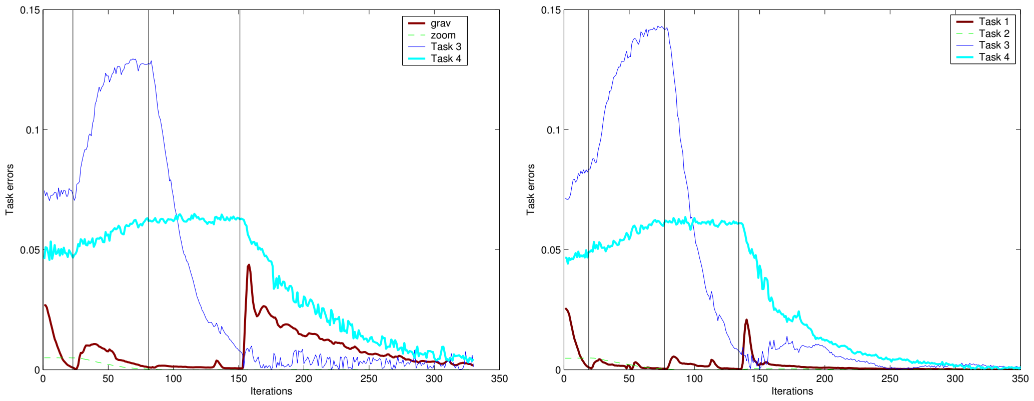

Figure 2.13: Results for a sequence of four tasks. Left: Analytic method. Right: *Correlation*. Due to calibration error, the analytical solution is not able to ensure the stack priorities. When a perturbation appears, it is not corrected until the active task has converged. The online estimation is able to quickly correct the perturbations. The perturbation average is thus much lower. The perturbation amplitudes are also lower. The vertical line shows the time instant where a new task is added in the stack.

bation average and perturbation-correction time, when properly tuned. However, it is not as robust to gain-tuning as the *Broyden* approach that could solve the task in all situations with the same parameters settings (note the *Broyden* performances for offline learning).

Experiment 2 - generalization to more tasks To verify that the method can learn all tasks, a complete sequence was done, consisting in: centering, zooming, rotation and perspective. All online learning methods could solve the task. The lowest perturbations were obtained using the *Correlation* method. Figure 2.13 cares the results of this method (on the bottom) with the analytical one (on the top). The vertical lines represent the time instant where a new task is added to the stack, after all the tasks already in the stack have converged to zero. It is interesting to see that in a badly calibrated system, the learning scheme yields better results than using the analytic solution, in terms of convergence speed, amplitude and average of the perturbations.

Experiment 3 - better learning through task sequencing A very important point was to note that learning improves the sequencing quality by reducing convergence times and the size of the perturbations. At the same time, the sequencing generates more efficient trajectories for learning. This experiment tests the hypothesis that learning four simpler tasks in sequence is easier than learning four tasks at the same time.

We compared the learning when running the robot under three different control laws. During the first run, task sequencing was used, in the same way as in previous experiments. In the second trial, all tasks are active at the same time. In other words, the same formalism is used but every task is active from the beginning, as opposed to starting a new task only after all the previous ones are completed. The last trial consisted on classical visual servoing, using only one single task of full rank. The conditioning number of the full-rank jacobian matrix was then estimated at each iteration. When a sequencing was used, the jacobians of all tasks were piled up and the overall conditioning number evaluated.

For *Correlation* and *Direct-inverse* methods, the condition number was the same for the three experiments. For these learning methods, the sequencing does not improve the learning. It is however very different when using *Broyden* algorithm. Figure 2.14 shows that for the *Broyden* method, the condition number of the matrices are much worse for the full task and convergence cannot be attained. Unfortunately, we have no theoretical explanation of this improvement, but we can note that an adhoc approach was done to bootstrap learning of bigger tasks with smaller tasks information in [Dodds et al., 1999] .

Experiment 4 - calibrated robot The last experiment was realized with an industrial robot. This robot is a six-DOF eye-in-hand robot with a very low payload/weight ratio. It has position and home sensors and its high repeatability allows to do a very precise calibration. Full, accurate calibration is available and the articular jacobian is no longer an approximation. However, the depth of the target is unknown and must be estimated to compute the correct jacobians. Mis-estimating depth will induce scale errors in the interaction matrices.

The experiments consist in a full sequencing until the camera desired position is reached. For each experiment, we vary the method used to compute the interaction matrix. The tasks were introduced in the same order as before, at fixed time for a better comparison ($\mathbf{e_g}$ at $t = 0$, $\mathbf{e}_\alpha$ at $t = 150$, $\mathbf{e_Z}$ at $t = 60$, and $\mathbf{e_R}$ at $t = 110$). Only the evolution of the perturbation norm is shown for each estimation scheme to shorten the results (Fig. 2.15). The complete results are detailled for the *Broyden* method in Fig. 2.16.

The first test ("current") shows that a perfect behavior is obtained when all the required knowledge is available. Depth was estimated with a pose computation algorithm, using the object geometric model and camera calibration. In the second trial ("misestimated"), depth was mis-estimated by a factor of 2. In the third experiment ("desired"), the interaction matrices were computed at the desired position using the desired depth. This is often done in vision-based control to enhance the control performance [Malis, 2004]. Unfortunately, as can be shown on the top of Fig. 2.15, inaccuracies in the projection operators introduce perturbations in the

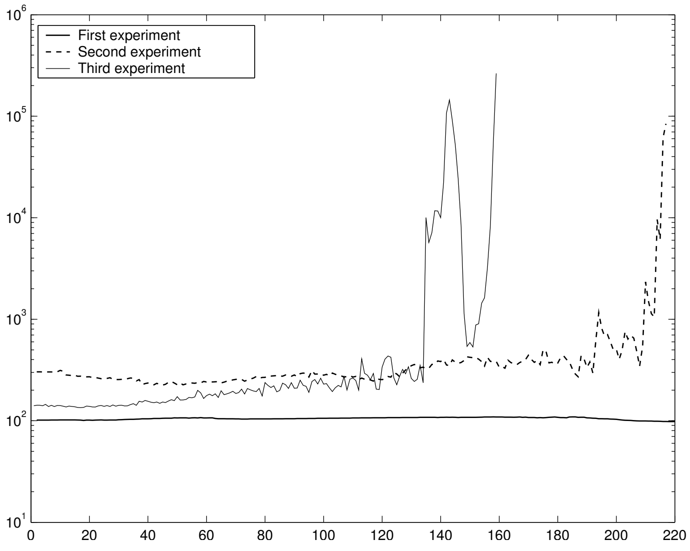

Figure 2.14: Condition-number evolution of the estimated interaction matrix during the servo. The matrix is learned from three different trajectories. The first one is a sequencing as done above (first experiment). The second used the sequencing formalism, but all the tasks were activated at the same time at the first iteration (second experiment). The last one is obtained from a classical visual servoing using a six-DOF task composed of all the visual features (third experiment). The matrix learned from a classical servo has a very large condition number. It increases until the servo becomes impossible. The learning realized from sequencing provides a properly conditionned matrix.

most prioritary tasks.

In a second set of experiments presented on the bottom of Fig. 2.15, we analyze the use of estimated jacobian matrices. The first two trials use both the online and offline versions of the *Correlation*. The third one starts with the analytical matrix, that is updated with the *Broyden* rules. These methods are compared with the same analytic version ("current") as before.

Figure 2.15 shows the results of the perturbation for each estimation scheme. As expected, the use of the "perfect" analytical solution leads to better results than the versions with estimated jacobians. However, online estimation always outperforms analytical approximations to the true jacobian. In most experiments, the best behavior was obtained with the *Broyden* algorithm, using the true interaction matrix as the initialization. It provides a very robust and fast online estimation without the need of a first offline learning. The performance of all methods can be improved if a small time period of learning is done in advance. Around 80 samples were enough for all methods.

Finally, Fig. 2.16 shows the results with the estimated jacobians using the *Broyden* method for all the four tasks. The tasks were introduced in the same order as before, at a fixed time for a better comparison ($\mathbf{e_g}$ at $t = 0$, $\mathbf{e}_\alpha$ at $t = 50$, $\mathbf{e_Z}$ at $t = 80$, and $\mathbf{e_R}$ at $t = 120$). We can note that the perturbation amplitudes are very small, for all tasks. They are also quickly reduced. The perturbation average is nearly zero. The robot behaviour is nearly perfect. The perturbations are not visible when runing the servo.

2.4.5 Conclusions

We have described the use of incremental SMMs for control purposes, equivalent to the Jacobian matrices used in Visual Servoing. Several learning methods have been tested for jacobian estimation in task sequencing. The *Broyden*, *Correlation* and *Direct-inverse* methods were tested and compared. An online version of the *Direct-inverse* method was proposed.

It is well know that visual servoing is robust to small errors in the jacobian matrices. For that reason it is not surprising that an estimated jacobian will be sufficient for control purposes. However, the jacobian matrices are also used to compute the projection matrices necessary for the task decomposition approach. In that case, even small errors in the jacobians may dramatically affect the projection operators and the overall performance.

Our results show that it is possible to learn the jacobians in the context of a task sequencing approach. All the learning methods were able to learn the jacobian matrix and accomplish the sequence of tasks, both in eye-in-hand and eye-to-hand configurations.

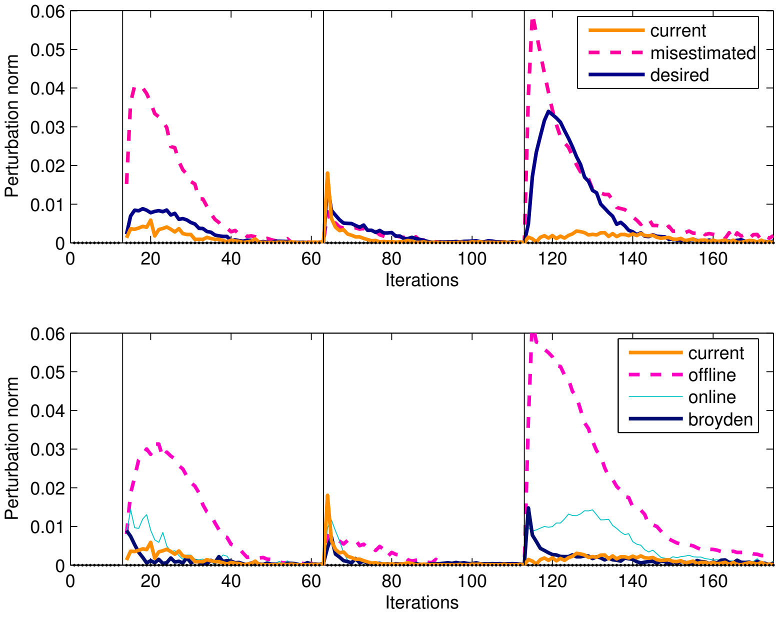

Figure 2.15: Perturbation norm (i.e. norm of the error of the tasks already completed) during the positioning. Top: results obtained with the true jacobian ("current") and two approximate versions where the depth is misestimated by a factor of 2 or set to the desired value in the goal position. Bottom: comparison of the true jacobian version ("current") with learned jacobians (Correlation offline and online) and Broyden. Small approximations in the analytic solution can generate disturbance that are reduced using a learning scheme.

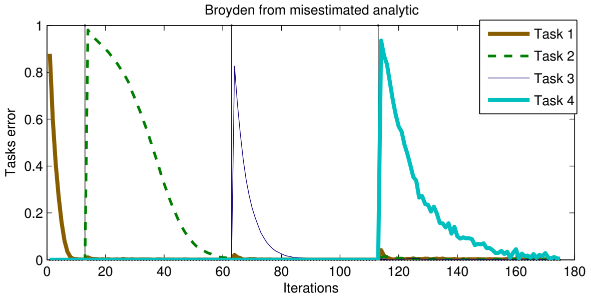

Figure 2.16: Correcting a depth mis-estimation by an online estimation of the interaction matrix. Results for a sequence of four tasks sequencing on the full-calibrated robot Afma6. The jacobian estimation method is *Broyden*, using the analytical solution as an initialization of the learning.

As expected, when the "perfect" analytic solution is available, it yields the best performance. However, in the presence of model uncertainty or estimation errors, online learning methods lead to the best results. If the calibration is not precise enough, learning methods are always better. For the learning methods, the online version is better than the offline learning (used as initialization), with respect to the time needed to reject transient perturbations, and also the amplitude of this transient.

Another important observation is that the use of task sequencing can indeed improve the quality of the learning, in the sense that the (partial) estimation problems are better conditioned. Conversely, learning improves the sequencing task by reducing the perturbations and convergence times.

In term of comparison of the various learning methods, *Correlation* yielded the best results. This method can be tuned in such a way that performances are nearly perfect, in terms of the time needed to reject the perturbation and the amplitude of the transient. The *Broyden* method performed also very well, being very robust. It does not require any tuning, and can perform all the tasks with the same parameters. Moreover, it does not require any offline learning phases.

We believe our results show that task-sequencing and online jacobian learning can be used together, with mutual benefits for the learning and for the task decomposition. This combination can then be used in a wide family of systems where accurate calibration information is not available or some parameters (e.g. depth) cannot be estimated accurately.

Using the proposed solution, all the redundancy formalism can also be used, in particular the sequencing approach. In future work, we plan to apply these techniques to real robot task such as grasping using vision-based control while taking into account all the robot constraints (joint limits, visibility ...). We also may explore the use of multiple jacobian models to better adapt the learning in some parts of the workspace. This might be particularly relevant if the system is operated open loop and not under closed loop control. If some open-loop motions are necessary maybe several models should be kept for better fitting in particular positions. These aspects will be the motivation of a grasping method presented in next chapter.

2.5 Final Remarks

In this chapter several mechanisms suited to be used in sensory-motor coordination were presented. These behaviors are the most fundamental for an effective development. All mechanisms were learned from auto-observation, following a developmental approach. The learning process is open-ended, in the sense that the system can start performing soon after "birth", while learning continues in an incremental manner. All excitations were produced by the robot and no external supervision was needed.

The following developmental stages will use all the knowledge acquired to accomplish more complex behaviors. In particular, once sensory-motor coordination is mastered, the system (human or robot) can start interacting and experimenting with (external) objects. It is important to notice that all the mechanisms already learned will keep on adjusting themselves over time to further improve their quality.

Chapter 3

World Understanding and Interaction

We have seen, in the previous chapter, how different types of sensory-motor maps (SMM) can be learned through self-observation in an initial developmental stage. As such SMMs are learned, the robot (or a human infant) progressively acquired control over its own body. This control ability facilitates new exploratory actions of the system, which in turn will lead to the acquisition of additional knowledge or capabilities to facilitate subsequent stages in development. Once the system has learned how to control the arm posture, it is now able to explore the world objects through reaching and grasping being able to: observe object properties and learning about their affordances.

A very importance and interesting theory about the world and our relation with it is the theory of *affordances*[Gibson, 1979]. As Gibson says: "The affordances of the environment are what it offers the animal, what it provides or furnishes, either good or ill. (..) I mean by it something that refers to both the environment and the animal in a way that no existing term does. It implies the complementarity of the animal and the environment." An affordance is the usability of something in the world to a particular animal. Water is something we can touch, drink, swim but not walk. For a Water strider (*Aquarius remigis*) water is for walk, affords support.

The presence of visually salient objects in the field of view motivates the robot to move the arm toward such objects and into attempting to grasp them. By grasping (or pushing) the object, one can learn about non-visual object attributes: weight, temperature, roughness, elasticity, or even how to grasp it in a stable manner (that depends on the mass distribution).

During this exploratory process, the system will then learn how to perform certain manipulative (grasp) gestures. The recent discovery of a class of visuo-motor neurons in the premotor cortex of some macaque monkeys suggests that the ability to perform certain goal-oriented gestures may facilitate the recognition of similar gestures by other individuals. These so-called "mirror neurons" [Fadiga et al., 2000] discharge both when the monkey or a demonstrator perform the same (specific) goal oriented gesture.

The ability to recognize other individual manipulative gestures has two important consequences. It represents an implicit form of communication (gesture understanding) and will allow the system to learn or update new affordances.

The first part of this chapter is dedicated to the problem of object grasping based on previously learned sensory-motor maps. The remainder of the chapter describes an approach to model how gesture recognition is facilitated by the ability to perform those same gestures.

3.1 Reaching and Grasping

It is known from developmental psychology and physiology that grasping an object involves two distinct movements: reaching and grasping. Infants start reaching for objects without any visual feedback. The movement is triggered by a visual stimulus (attention directed toward a visually salient scene object) but it is not guided by vision throughout the entire action. In case of failure, the movement restarts from the very beginning. Only at later developmental stages, the actual grasp is visually controlled (see Table 1.1 for details).

It is worth stressing that these two-phase movements are always used, even after the ability to perform visual control has been developed. A pre-requisite to visually control the hand toward an object is that both the hand and the object are within the field of view, which justifies the initial (open-loop) reaching phase.

[Gaskett and Cheng, 2003] presents a system with the open-loop part implemented with a self-organizing map, then a coarsely calibrated closed-loop is used to move the head and, finally, the arm moves in open-loop. A relaxation mechanisms allows the head to return to a comfortable position. The closed loop mechanism is implemented with several heuristics that substitute the jacobian matrix in a standard visual servoing formulation. Our approach, presented by [Lopes et al., 2005], improves that algorithm by having the goal of grasping, instead of just reaching, in a more precise way, some object interaction and imitation is also performed. It consists in three steps. First the head looks directly to the object. Then the arm moves in to the front of the eyes in open-loop. These two maps are learned by self-observation and using a statistical learning method. The final error correction is done with an uncalibrated visual servoing method. Other approaches for object grasping were either completely visual controlled [Kragic et al., 2002], with problems in guaranteeing the presence of the hand in the visual field, or completely open-loop [Natale, 2004] with no capability of error correction.

In this thesis, the artificial development of grasping will follow along similar phases. In the end, a two-phase movement will be used for the robot to grasp and explore objects. One can then distinguish the following steps:

1. Visual detection of the object of interest.
2. Reaching (or transport) phase in open-loop.
3. Final grasping phase under visual (closed loop) control.

3.1.1 Object detection and mapping

It is very useful to know where an object is and whether it can be grasped or not. After all the time spent interacting with its own hand, the system can already distinguish objects at different depths and search for the desired one.

There is neurological evidence of spatial aware neurons that are active when movement or objects are present near the skin [Rizzolatti et al., 1977]. It is also known in developmental psychology that infants became aware of the near and far space very early [Rochat et al., 1999].

The near-space contains the touchable objects and the own body, and it is very important to perceive what happens there. [Bernardino and Santos-Victor, 2002] suggest a method where the disparity between images is used, together with some neuronal-based filters, to segment objects at different depths. The head can be moved to look toward the hand using disparity as a feedback signal to control it.

By having an exploratory behavior, we create a map of the localization of objects around - the peripersonal map - through various steps:

1. Find an object in the visual space
2. Foveate on this object
3. Memorize the object position in body centered (proprioceptive) coordinates.

The robot thus creates a mental image of the surrounding space. The position of the objects are memorized in terms of proprioceptive coordinates. In a case of a moving robot, this map should be updated with the ego-motion.

The whole grasp process is triggered by the detection of an interesting object within the field of view. This visual saliency can be defined in terms of motion, color, texture that clearly distinguishes an object from the surrounding world.

Visual attention can be drawn toward a certain object as a consequence of a visual search process driven by some contextual information (e.g. searching a specific object) or purely driven from the image data.

Once the object of interest is detected, the system verges toward it. As a result, the object position can be represented in ego-centric coordinates.

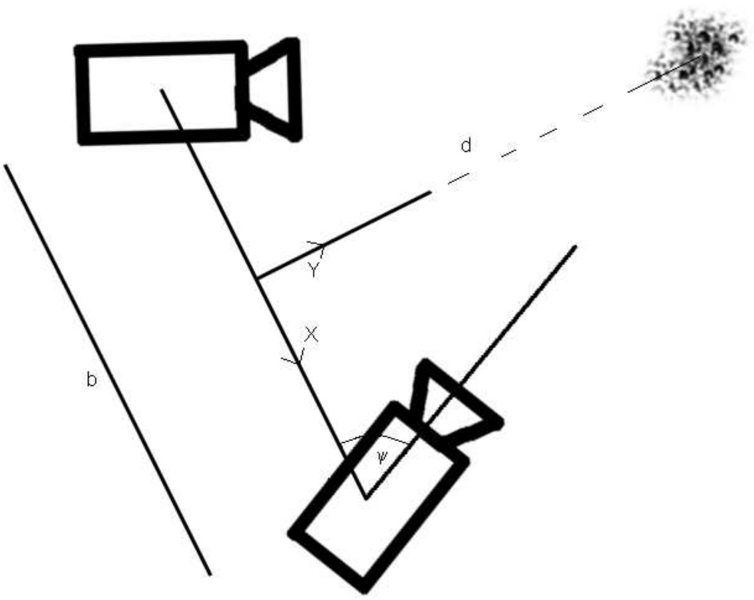

Figure 3.1: Object localization and mapping from camera coordinates to a robot-centric frame of reference.

We assume that the vergence angles of the two cameras (eyes) is symmetric. Then, the measurement of the vergence angles (ψ_1, ψ_2) determine the distance to the object in the 3D space:

$$d = \frac{b}{2} \tan \psi_1$$

where d denotes the distance to the object and b is the interocular distance, refer to figure 3.1 to details. The object ego-coordinates can be determined as:

$$x = d \cos \psi_1 \ , \ y = d \sin \psi_1$$

where x and y denote the object coordinates in a robot-centric frame of reference. If the robot is moving in the environment it is useful to update the position of know objects and agents in the world without having to localize them. If the robot moves with linear velocity v_x and v_y and with angular velocity w_z, the object position can be updated according to:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -y \\ 0 & 1 & x \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ w_z \end{bmatrix}$$

where all quantities are expressed in head coordinates.

The coordinates of the detected target can now become a reference signal for the reaching process as we will see in the next section.

3.1.2 Reaching (open-loop)

As we have seen in the previous chapter, during the first developmental stage, the robot estimates a *Arm-Head Sensory-motor Map.* This map allows the system to move the hand toward an object. As the map may not be extremely accurate, this positioning capability will not be very precise but still achieving the accuracy sufficient to place the hand within the visual field of the robot.

Hence, if a simple (crude) trajectory is followed, the hand may well succeed in touching desired objects. The problem with this (open-loop) approach is the absence of a mechanism for error correction. This is the reason why babies in this phase restart the grasp quite often, instead of (incrementally) correcting it [Payne and Isaacs, 1999].

Hence, the use of the sensory-motor map explained in the previous chapter is sufficient to have the robot hand and the target of interest within the field of view. This is the necessary pre-requisite for the final grasp phase that can now make use of visual closed loop control.

3.1.3 Final grasping phase (closed-loop)

The second stage of object grasping relies on visual feedback, coping with the problem of error correction. After using the *Head-Arm map* to move the hand to the objects vicinity, accurate positioning is achieved by closed-loop visual guidance. With this phase, it is possible to grasp objects in a reflex type manner, the hand closing after touch.

The visual controlled phase of grasping makes use of the dynamic sensory-motor maps (jacobians) learned during the first developmental stage and explained in Chapter 2. This incremental sensory-motor map allows for incrementally correct the distance and orientation between the robot hand and the object to grasp.

Figure 3.3 shows the resulting behavior of the system while grasping objects. The hand is closed after sensing the contact with the object. The capability of pre-shaping the hand will only develop at a later stage. For small grasping velocities, this type of movement can be sufficient, but bigger velocities will require learning some form of pre-shaping and predicting the time of contact with the object.

When the working volume is very large, the Jacobian can no longer be accurately estimated with a single linear model. To overcome this problem, we propose a new method to estimate approximations of the Jacobian in different regions of the workspace.

With a single linear model, the update mechanism must be fast enough to yield an accurate model for each region. This is not the case when (fast) open-loop movements are used, thus preventing the system from updating the model of a previously estimated Jacobian for a certain region.

Instead, we partition the workspace in several regions, $R_i,\ i = 1 \ldots N$. At each instant, the measured distance c between the current position and all the regions is used to select the Jacobian $\mathbf{J}$ corresponding to the nearest area, R_i. We use the Mahalanobis distance with covariance $\mathbf{D}$. The covariance can be updated online to reduce the number of regions and to better adjust the linear model to the non-linear (global) jacobian. Trying, to update the regions center creates problems by overlapping regions and with region transitions. The regions are initialized when the system is far from existing regions, reducing the distance between regions gives a better approximation to the true jacobian but, as the number of models increase, it takes more samples to have a reliable model. When adding a new region the jacobian is initialized with the same value as the nearest one.

The Jacobian update rate (α) should be larger when the model is inaccurate and then reduced to improve convergence. One measure to access the model quality (mq) can be:

$$mq(t) = mq(t-1) + \gamma < \mathbf{\Delta y}, \mathbf{J_k \Delta} \theta >$$

γ is a decaying factor and $< . >$ represents internal product. mq is positive when the observed movements has a direction error less than 90 *degrees*.

The regions centers, x_i, may correspond to motor features $x = \theta$, visual features $x = y$ or a combination of them. With visual features there is the possibility of doing planning in visual space but there are different motor positions that give the same visual features and should have different linearizations. Table 3.1 presents the complete algorithm for doing the visual controlled grasp.

It is important to note that the computation of $\mathbf{J}^+$ must be done quite carefully. As some directions are not observed, the Jacobian inversion will be very unstable. To solve this problem, the pseudo inverse is implemented with a SVD method and all singular values smaller than 10% of the largest one, are treated as zero.

Some common problems with Visual servoing methods are described in [Chaumette, 1998]. Our method solves the problem of the Jacobian derivation and the calibration of both the robot and cameras. In general, these methods are sensitive to the initial position and prone to fall in local minima. In our work, the system always starts near the final position due to the approach phase with the *Head-Arm Sensorimotor Map*, thus making convergence easier.

We made several experiments to access the quality of the resulting algorithm. Our system measures a specific dot in the hand with two cameras giving an image position of the hand (u_l, v_l) for the left eye and (u_r, v_r) for the right eye. The features are calculated as follows:

$$\mathbf{y} = \begin{bmatrix} \frac{u_l + u_r}{2} \\ \frac{v_l + v_r}{2} \\ u_l - u_r \end{bmatrix}$$

To move the system to the desired image position y^*

1. Choose the region R_i corresponding to the actual state x:

$$c_i = (\mathbf{x} - \mathbf{x}_i)^T \mathbf{D}_i (\mathbf{x} - \mathbf{x}_i)$$

$$R_i \ : \ \min_i \ c_i$$

   if $\max c_i < C$, create a new area l with $x_l = x$, $D_l = D$ and $J_l = J_i$. Choose $R_i = R_l$.

2. Apply the control law:

$$\boldsymbol{\Delta}\theta = K_i \frac{J_i^+ \left(\mathbf{y}^* - \mathbf{y}\right)}{\left\| J_i^+ \left(\mathbf{y}^* - \mathbf{y}\right) \right\|}$$

3. Observe image changes Δy

4. Make the update to the model i corresponding to position x with:

$$\hat{J}_i = \hat{J}_i + \alpha_i \frac{\left(\Delta\mathbf{y} - \hat{J}_i \Delta\theta\right) \Delta\theta^T}{\Delta\theta^T \Delta\theta}$$

5. if $|y^* - y| > E$ goto 1

Table 3.1: Algorithm used for the final (closed-loop) grasp phase using uncalibrated visual servoing.

This gives position and distance information estimation of the hand related to the head. The head was maintained fixed and four arm joints were used. The distance between the central point of each zone was 10 *degrees*. The Jacobian update rate was equal in all regions and choosen as $\alpha = 0.1$ while $mq < 0$ and $\alpha = 0.01$ while $mq > 0$.

Figure 3.2 shows quantitative results of the grasp sequence shown in Figure 3.3 using our proposed algorithm. The hand was positioned near the object using the *Head-Arm Sensory-motor Map*. The resulting positioning error corresponds to about 8 *cm*. The associated image error is corrected in the final phase (visually controlled) with a linear convergence rate.

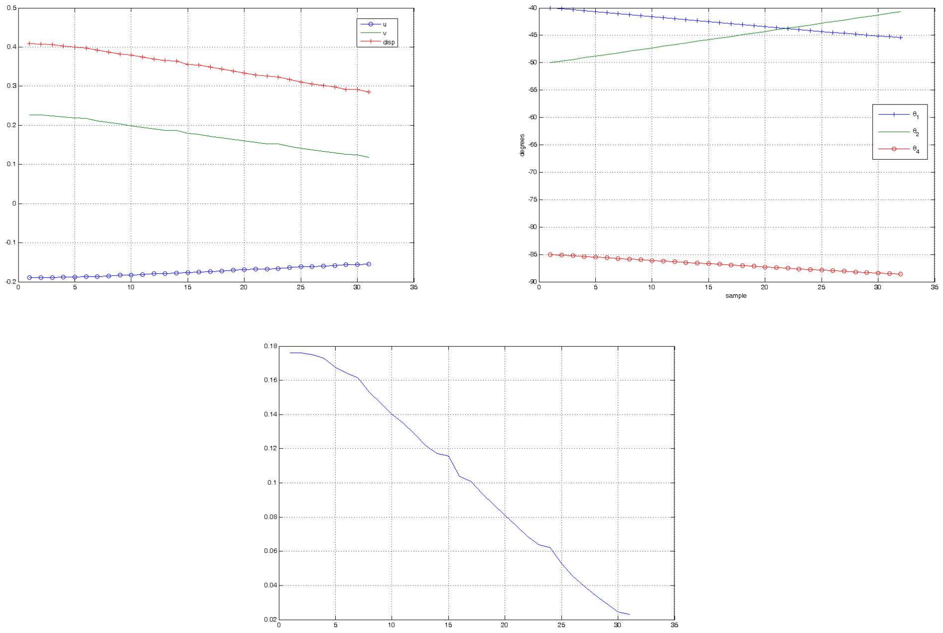

Figure 3.2: Servoing results for object grasping. First graph: evolution of the visual features to the desired values. Second graph: evolution of the joint angles. Third graph: convergence rate.

To have a successful grasp, defining the final position is not enough. The hand must be pre-shaped and oriented to arrive to the target with the palm, avoiding touching the object with the fingers. The sequencing technique can give a different solution for this problem. We can define two tasks to defining a good trajectory to grasp an object: orienting and approaching.

For orienting, first we define a line from the index finger tip to the thumb tip. This line should be perpendicular to the line defined by the hand palm center and the goal position.

In the end we have a task with 5 dof: 3 for approaching and 2 for the orientation. There is still a degree of freedom corresponding to the rotation along the axes defined

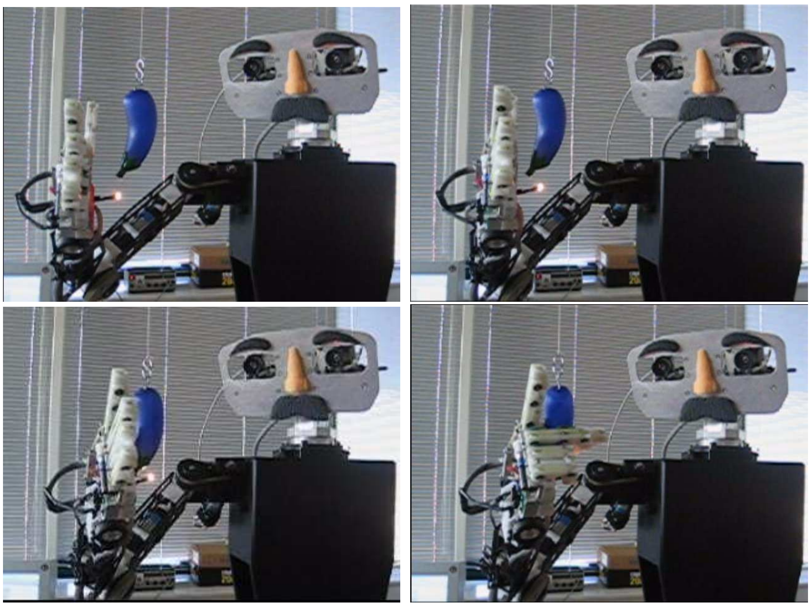

Figure 3.3: Several frames in the sequence from the initial position resulting from the *Head-Arm Map*, then the visual guided part and finally the object grasping.

by the fingers.

3.1.4 Beyond reaching and grasping

The methods described in the previous sections endow the system with the ability to approach and grasp objects. Such skill can now be used for exploring the objects in the world, sensing their properties and learning their affordances.

For example, the system must learn how to perform stable grasps, which implies assessing the mass distribution of an object. This problem is out of the scope of the thesis but the interested reader can be directed to [Mishra and Silver, 1989]. Exploring physical properties of objects is a good example of what the robot (or an infant) might do in this phase. In [Fitzpatrick et al., 2003] an artificial system learns about rolling objects and their preferred directions of movement.

In the following section we assume that some object properties, affordances and stable grasps have been learned. We then see how the evidence of mirror neurons suggests that action recognition is facilitated by the ability to perform a certain action.

3.2 The Mirror neurons and grasping action recognition

In this section we present a model for grasping recognition based on recent neurological findings. We assume that the system is already capable of interacting with objects in the world, discover properties of those objects or learn how they can be grasped. All these abilities are built upon previously learned knowledge or skills.

Here, we extend the concept a step further by showing how the ability to perform such grasping actions allows the system to recognize similar actions performed by other individuals. This process will be based on the results and hypotheses drawn by neuroscientists after the discovery of a class of visuomotor neurons in the area F5 of the pre-motor cortex of macaque monkeys. Such neurons may correspond to a motor representation for objects (and their affordances) and gestures, as we will discuss in the following sections.

3.2.1 Canonical Neurons and object affordances

Many objects are grasped in very precise ways, allowing them to be used for some specific purpose. A pen is usually grasped in a way that affords writing and a glass is hold in such a way that we can use it to drink. Hence, if we recognize an object that is being manipulated, it immediately tells us some information about the most likely grasping possibilities (expectations) and hand appearance, simplifying the task of gesture recognition.

The link between objects and their affordances [Gibson, 1979] is possibly played by the *canonical neurons* found in the area F5 of the macacque's brain [Murata et al., 1997]. Surprisingly, Canonical Neurons respond when objects, that afford a *specific* type of grasp, are present in the scene, even if the grasp action is not performed or observed. If two objects can be grasped in the same way, the same neurons will fire when either object is presented.

The affordances of an object have thus an attention property because the number of possible (or likely) events are reduced, thus overcoming possible ambiguities. This will be the first module of our overall system architecture, where we will interpret grasp types as a particular type of *affordances* associated to a certain object.

Affordances provide important prior information for recognizing gestures, by indicating which grasps are more likely, when acting upon a certain object class. Hence, Canonical Neurons could be a representation of an object in terms of the grasp actions (or affordances) that could be used upon that object.

[Fitzpatrick et al., 2003] developed a system with some internal stimulus to "understand" the environment. Several experiments were design to allow the robot to learn the effects and consequences of self-generated actions. The actions consisted in poking objects along principal and non-principal axis and the observed features where object texture and resulting motion. The robot could learn how to segment the human motion and then imitate this movements, in a generalized way, of humans on objects. As a consequence of the learned object motion, some object properties were estimated such as the existence, or not, of a principal axis of motion. [Natale et al., 2004] developed a system able to incorporate tactile information with weight in order to classify objects.

3.2.2 Mirror Neurons and motor representations

The mechanism for human recognition of other's actions may be related to a population of visuomotor neurons, the *Mirror* Neurons [Fadiga et al., 2000], also found in the F5 area of the macaque's brain. These neurons discharge during the execution of hand/mouth movements. In spite of their localization in a pre-motor area of the brain, *mirror* neurons fire not only when the animal performs a specific goal-oriented grasping task, but also when observing that same action performed by another individual. By establishing a direct connection between gestures performed by a subject and similar gestures performed by others, mirror neurons may be intimately connected to the ability to imitate found in some species [Ramachandran, 2000], establishing an implicit level of communication between individuals.

The discovery of mirror neurons raises the fundamental question of understanding the role of motor information for "visual" gesture recognition, and how can it be facilitated by the fact that we know how to perform those gestures. It has also been shown that lesions in the motor part of the brain do affect recognition capabilities.

This observation suggests that the motor system responsible for triggering an action is also involved when recognizing that same action. In addition, it seems to indicate that recognition is done in motor terms and not in a visual space. Since only visual information is available during recognition, this hypothesis requires the existence of a transformation, mapping visual to motor spaces.

We have seen in Chapter 2 that such Visuomotor maps could be learned through auto-observation. Those maps could allow the positioning of the arm in a certain configuration in the visual field or the appearance of the hand for a given motor command. In addition, the ability to reach for and grasp objects in the scene enables the system to explore and learn about stable grasps, object properties and affordances.

Here, we extend the concept of learning-by-observation to the observation of other individuals. By observing *other* people manipulating objects, we can learn the most likely (successful) grasps or functions for a given class of objects. We can also learn what effect certain gestures produce on world objects subject to manipulation.

We have discussed already that grasp actions can be partitioned into the *transport* and *grasp* phases [Fogassi et al., 2001]. It has been shown that the transport phase can change significantly, according to the particular grasp type that is performed in the end of the movement. However, it seems that this information is not used by humans for gesture recognition and only the final grasp phase seems to be relevant.

Figure 3.4 illustrates the hand appearance during the approach phase, together with the final phase of two broad classes of grasps that will be used in this work: precision grip and power grasp.

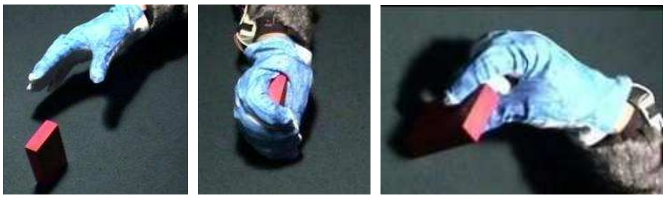

Figure 3.4: Hand appearance during the approach phase (left), power grasp (center) and precision grip (right).

Gesture recognition has been addressed in the computer vision community in many different ways [Black and Jepson, 1996, Gavrila, 1999, Rehg and Kanade, 1995, Wu and Huang, 2000]. The difficulty of hand tracking and recognition arises from the fact that the hand is a deformable, articulated object, that may display many different appearances depending on its configuration, viewpoint or illumination. In addition, there are frequent occlusions between hand parts (e.g. fingers). Traditional approaches imply performing full $3D$ reconstruction of the hand, followed by a pose classifier. To make the 3D reconstruction, it is necessary to track the fingertips, while handling the multiple occlusions generated by the complex hand motion. State-of-the-art algorithms rely on good initial estimates and require sophisticated kinematic models of the hand. This is, in general, quite difficult, depending on the taken viewpoints and image acquisition conditions. To overcome this difficulty, we exploit more iconic representations for the hand shape, that are commonly believed to be used by humans when recognizing (known) gestures.

This follows the common approach to recognition that involves comparing acquired visual features to data from a training set. Instead, the observations of mirror neurons suggest that recognition might be carried out in terms of motor variables.

The advantage of doing this inference in the motor space is two-fold. Firstly, while visual features can be ambiguous, we show that converting these features to the motor space may reduce ambiguity. Secondly, as the motor information is directly exploited during this process, imitation can be done immediately, as all the information/signals are readily available.

The question that remains is that of choosing what type of visuomotor map (and therefore, visual stimuli) should be used. As we will focus on the classification and imitation of coarse gestures (power grasp and precision grip), we will rely on global appearance-based image methods. Together with the prior information provided by the canonical neurons, appearance based methods offer an easier, fast and more robust representation than point tracking methods.

This is clearly distinct from most approaches for gesture recognition, where only visual information is involved. We show that performing the recognition step in the motor space really simplifies the problem by affording a larger degree of invariance to viewpoint modifications.

The work described in [Oztop, 2002] is closely related to ours and proposes a model for *mirror neurons*. However, the visual features used in that work are very difficult to extract from a video sequence, which makes the approach unreliable, also there is no use of the expected way to grasp an object to reduce ambiguities.

As a final comment, we would like to remark that, if we want to look at gestures performed by the entire arm, it requires performing some sort of visual transformation to deal with the problem of viewpoint shape variance [Bruner, 1972]. For hand movements, our approach is invariant to large variety of view points. Also, during self-observation, the system can generate a large variety of hand visual stimuli that will be used for the construction of visuo-motor maps. The viewpoint transformation for arm gestures is specifically addressed in Chapter 4.

Next section presents a Bayesian approach for gesture recognition that includes models of the *canonical* and *mirror* neurons, using visual appearance methods. The approach leads to excellent classification rates by classifying in motor space.

3.2.3 A neuronal inspired bayesian model for gesture recognition

Gesture recognition can be modeled in a Bayesian framework, which allows to naturally combine *a priori* information and knowledge derived from observations (likelihood). The role played by canonical and mirror neurons will be interpreted within this setting.

The approach we propose here differs from other works in several ways: (i) use of object affordances in the recognition process (canonical neurons); (ii) recognition is performed in the motor space (mirror neurons) and (iii) global of descriptors for hand appearance.

Let us assume that we want to recognize (or imitate) a set of gestures, G_i, using a set of *observed* features, F. For the time being, these features can either be represented in the motor space (as mirror neurons seem to do) or in the visual space (directly extracted from images). Let us also define a set of objects, O_k, present in the scene, that represents the goal of a certain grasp action.

Figure 3.5 present a logical diagram describing the approach. The observed object gives a prior expectation about the grasping type. When combining this information with motor features (or any kind of features extracted from the observed data) an a posteriori probability is computed that classifies the gesture. The final result can be used to update the model.

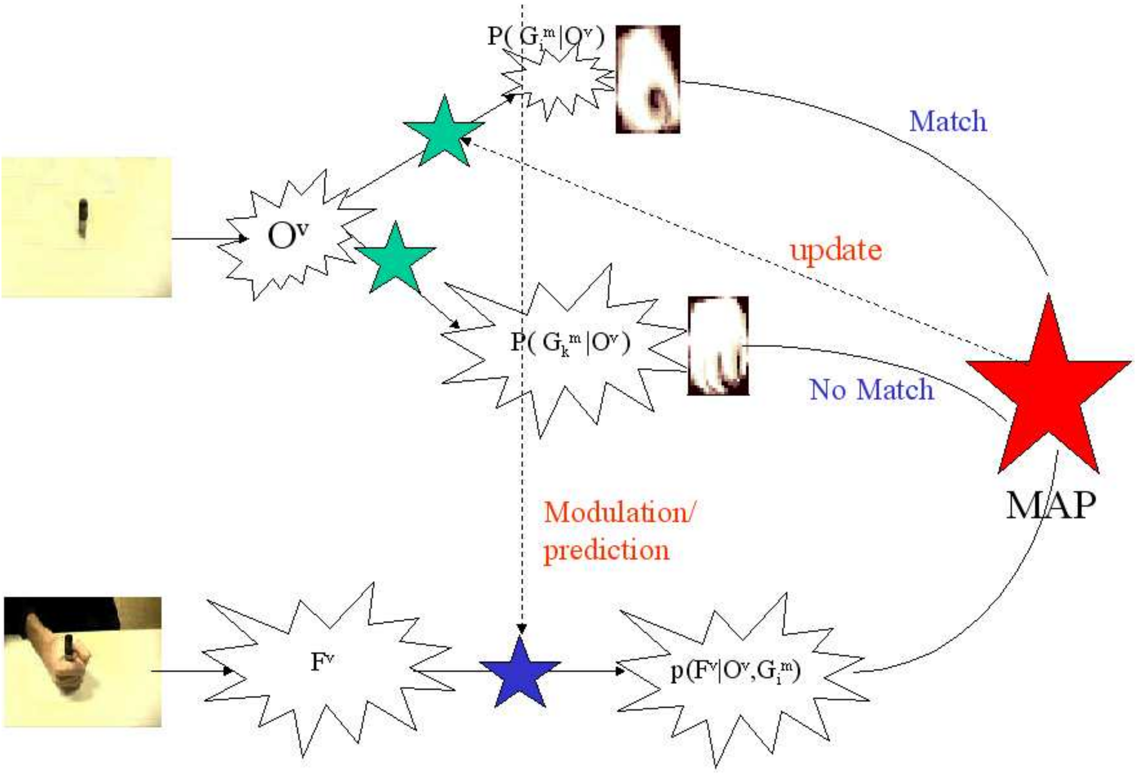

Figure 3.5: Bayesian grasping-type recognition method. Mixing information about object affordances with runtime visual appearance gives improved recognition rates.

The prior information (grasp affordances) is modeled as a probability density function, $p(G_i|O_k)$, describing the probability of each gesture given a certain object. The observation model is captured in the *likelihood function*, $p(F|G_i, O_k)$, describing the probability of observing a set of (motor or visual) features, conditioned to an instance of the pair gesture and object. The *posterior* density can be directly obtained through Bayesian inference:

$$\begin{aligned} p(G_i|F, O_k) &= p(F|G_i, O_k)p(G_i|O_k)/p(F|O_k), \\ \hat{G}_{MAP} &= arg \max_{G_i} p(G_i|F, O_k) \end{aligned} \tag{3.1}$$

where $p(F|O_k)$ is just a scaling factor not influencing the classification.

The MAP estimate, G_{MAP}, is the gesture that maximizes the posterior density in Equation (3.1). In order to introduce some temporal filtering, features of several images can be considered:

$$p(G_i|F, O_k) = p(G_i|F_t, F_{t-1}, ..., F_{t-N}, O_k),$$

where F_j are the features corresponding to the image at time instant j. The posterior probability distribution can be estimated using a naive approach, assuming independence between the observations at different time instants. The justification for this assumption is that, recognition does not necessarily require the accurate modeling of the density functions. We then have:

$$p(G_i|F_t, F_{t-1}, ..., F_{t-N}, O_k) = \prod_{j=0}^{N} \frac{p(F_{t-j}|G_i, O_k)p(G_i|O_k)}{p(F_{t-j}|O_k)}$$

The role of canonical neurons

The role of canonical neurons in the overall classification system lies essentially in providing the affordances, modeled as the *prior* density function, $p(G_i|O_k)$ that, together with evidence from the observations, will shape the final decision. This density can be estimated by the relative frequency of gestures in the training set.

Canonical neurons are also somewhat involved in the computation of the likelihood function, since it depends both on the *gesture* and *object*, thus implicitly defining another level of association between these. Computing the likelihood function, $p(F|G_i, O_k)$, is more elaborated and is described in detail next.

Estimating the likelihood function

As the likelihood function may correspond to a complex distribution, it will be modeled by a Gaussian mixture, which is fitted to data points. In what follows we will describe the process of fitting a mixture model to a density, $p(x)$:

$$p(x) = \sum_{j=1}^{K} \pi_j \, p(x|j),$$

where $p(x|j) \sim N(\mu_j, \sigma_j)$, is a Gaussian distribution. For a proper probability density function, we need to ensure that $\sum_{i=1}^{K} \pi_i = 1, \; \pi_i \geq 0$.

The Expectation-Maximization (EM) algorithm can be used to estimate the parameters μ_i, π_i and the covariance matrix C_i, for the multidimensional case, that best fit the data. The main problem with this solution is the necessity of knowing in advance the number of kernels, K. In [Baggenstoss, 2002, Vlassis and Likas, 1999] there is the option of modifying the number of Gaussian kernels used to best fit the data. The number of kernels can be increased during the learning process, based on a measure designated as the total *kurtosis*, $\mathcal{K}$:

$$\mathcal{K} \triangleq \int_{-\infty}^{\infty} \left(\frac{x - \mu_j}{\sigma_j} \right)^4 \frac{p(j|x)}{\pi_j} p(x)dx - 3$$

The *kurtosis* measures how far a distribution is from a Gaussian and it is zero for a Gaussian function. If the *kurtosis* is not close to zero for a given kernel, it means that the data are not Gaussian and this kernel is split. On the other hand, the number of kernels can sometimes be reduced (merged) in order to reduce the model complexity. A "closeness" metric between two kernels, can be defined as follows:

$$d(p_1, p_2) = \frac{\prod_{x_i \in X_1} p_2(x_i) \prod_{x_i \in X_2} p_1(x_i)}{\prod_{x_i \in X_1} p_1(x_i) \prod_{x_i \in X_2} p_2(x_i)}$$

where X_i stands for the data points used for the estimation of $p_i(x)$.

Two different kernels can be merged if the distance between them is sufficiently small. At the end of this process, we have an estimate of the likelihood function directly from the data, without imposing a particular structure for the underlying distribution. An important point worth mentioning is that this method can cope with clusters with very irregular shapes and it automatically adapts to the shape of such clusters.

Modeling the mirror neurons

The classification done by our system has several properties similar to the mirror neurons. In this section we will see how to account to some of the observations regarding this neurons into our Bayesian framework. We must first consider a Visuo-Motor Map that transforms observed visual data, to the motor representations that will eventually drive the recognition process.

Visual versus motor features An image contains a large amount of highly redundant information. This allows for the use of methods whereby the image information is compacted in lower dimensional spaces, thus boosting computational performance. Our visual features consist of projections of the original image onto linear subspaces, using Principal Components Analysis (PCA). As a result, our images can be compressed to a 15 dimension coefficient vector.

Rather than representing the hand as a kinematic model built from tracked fingers and finger tips, we code directly the image as templates projected in the low-dimensional subspace. This method has the advantage of being robust and fast.

In a real (robotic or living) system, motor features would correspond to proprioceptive information about the hand/arm pose/motion. In our experiments [Schenatti et al., 2003], this is obtained through the use of a data-glove that records joint angles of someone's hand performing gestures. The structure of the VMM and the way it develops was addressed in section 2.2.

3.2.4 Experimental results

To gather data we asked several subjects to perform three grasps on different objects [Schenatti et al., 2003]. The experiment begins with the subject sitting in a chair with his hand on the table. Finally the subject is told to grasp the object that is in front of him.

The experiments include two types of grasp: power grasp and precision grip. Power grasp is defined when all the hand fingers and palm are in contact with the object. Instead, in precision grip, only the fingertips touch the object.

We considered three different objects: a small sphere, a large sphere and a box. The small sphere is sufficiently small so that only precision grip is allowed. The big sphere allows only power grasps. The box is ambiguous because it allows all possible grasps with different orientations.

Every experiment was repeated several times under varying conditions. The subject and the camera go around the table to cover a large variation of viewpoints. To record the sequences we use a stereo-pair. In total, we record the experiments from 6 different azimuths (12 if we consider the stereo-pair). The motor information is acquired with a data-glove [CyberGlove,] capable of recording 23 values of the hand configuration. We used the first 15 values that correspond to all the joint angles (3 for each finger). Finger's abduction and palm and wrist flexion were also available but they were not used in the recognition. Altogether the data-set contains sixty grasp sequences with three objects, two grasps with six different azimuths.

Figure 3.6 shows sample images of the data set acquired according to process just described. Notice the multiplicity of grasps and view points.

Every video sequence is automatically processed in order to segment the hand. First, a color-based clustering method, in the Y-Cr-Cb space, was applied to extract hand-colored pixels. The bounding box is determined based on the vertical/horizontal projections of the detected skin region. Finally, the hand is resized for a constant scale before applying the PCA. This approach yields uniformly scaled hand image regions. Figure 3.7 presents some segmentation results.

Table 3.2 shows the obtained classification rates. It allows us to compare the benefits of using motor representations for recognition as opposed to visual information only. The results shown correspond to the use of the ambiguous objects only, when the recognition is more challenging. We varied the number of viewpoints included in both the training and test sets, so as to assess the degree of view invariance attained by the different methods.

In the first experiment, both the training and test sets correspond to one single view point. Training was based on 16 grasp sequences, while test was done in 8 (different) sequences. The achieved classification rate was 100%. The number of visual features (number of *PCA* components) was also tuned and the value of

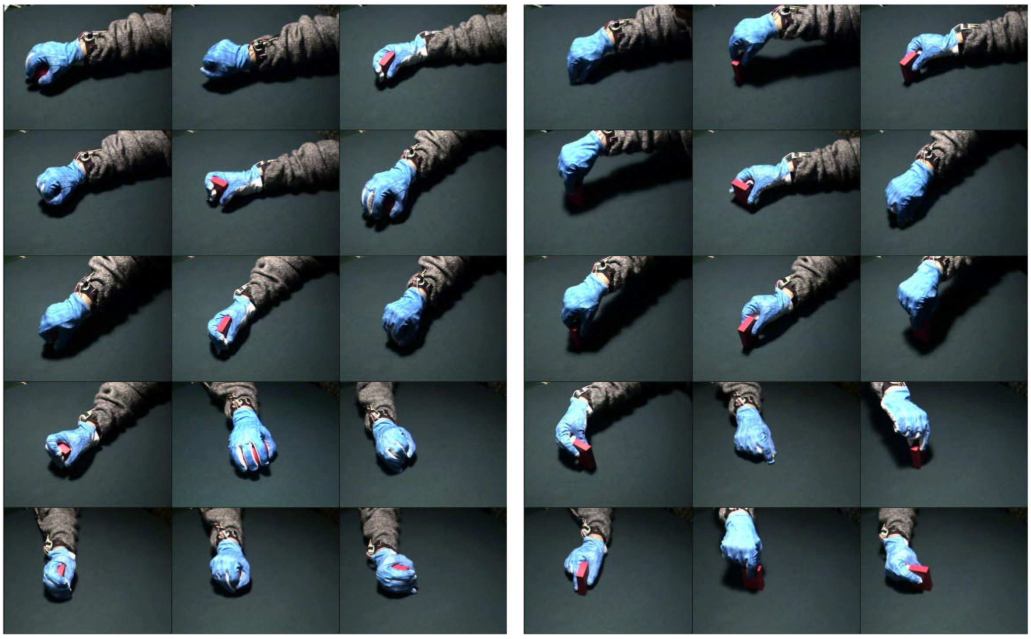

Figure 3.6: Data set illustrating some of the used grasp types: power (left) and precision (right). Altogether the tests were conducted using 60 sequences, from which a total of about 900 images were processed.

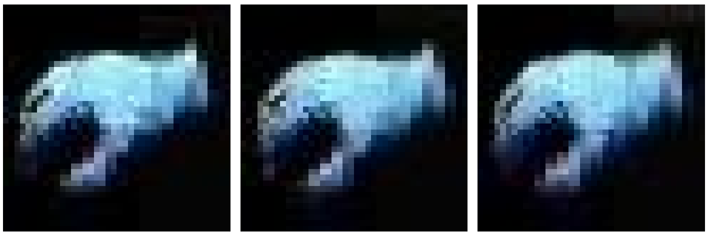

Figure 3.7: Segmentation results of scale-normalized hand regions automatically detected from colour clustering.

5 provided good results. The number of modes (gaussians in the mixture) were typically from 5 to 7.

The second experiment shows that this classifier is not able to generalizes to other view points / camera positions. We used the same training-set as in *Exp.I*, but the test-set is formed with image sequences acquired with 4 different camera positions. In this case, the classification rate is worse than random (30%).

In the third experiment, we added view point variability in the training set. When sequences from all camera positions are included in the training-set, the classification rate in the test-set drops to 80%. While this is a more acceptable value, it is nevertheless a significant drop from the desired 100%. This result shows that the view point variation introduces such challenging modifications in the hand appearance that classification errors occur.

The final experiment corresponds to the main approach proposed in this paper. The system learns a visuo-motor map during an initial period of self-observation. Then, the VMM is used to transform the (segmented) hand images to motor information, where classification is conducted. A very high degree of classification was achieved (97 %). Interestingly, the number of modes need for the learning is between 1-2 in this case as opposed to 5-7, when recognition takes place in the visual domain. This also shows that mapping visual data to motor representations, helps clustering the data, as it is now view-point invariant.

Notice that view-point invariance is achieved when the training set only contains sequences from one single view point.

	Exp. I (visual)	Exp. II (visual)	Exp. III (visual)	Exp. IV (motor)
	Training			
# Sequences	16	24	64	24
View Points	1	1	4	1
Classif. Rate	100%	100%	97%	98%
# Features	5	5	5	15
# Modes	5-7	5-7	5-7	1-2
	Test			
# Sequences	8	96	32	96
View Points	1	4	4	4
Classif. Rate	100%	30%	80%	97%

Table 3.2: Grasp Recognition results. Notice the gain obtained in the classification rate and viewpoint invariance due to the use if motor features.

These experiments show that motor representations describe the hand better. As only visual information is available during recognition, the process greatly depends on the *VMM.* The results also validate that our approach to estimate the VMM allows recognition to be performed. For the case of only one camera position the quality obtained was very good, if the number of visual features used were 15.

3.3 Conclusions

This chapter presented the second major phase in the development of the robotic system Baltazar. This developmental phase was facilitated by the ability to control one's own body. The control ability resulted from previous developmental stages and through the learning of various sensory-motor maps through self-observation.

The motivation in this stage is the ability to interact with objects in the world through reaching and grasping. The reaching action results naturally from the direct use of previously learned sensory-motor maps and does not require any visual feedback. In turn, the final grasping phase requires more accurate and dynamic sensory-motor maps (jacobians) to be estimated and used in a visual servoing loop.

The ability to reach for objects and grasp them allows a system (or an infant) to learn about properties of those objects (weight, temperature, friction, surface roughness, mass distribution) or affordances of such objects, like ways of grasping and what they can be used for.

In the final part of the chapter we show how the ability to perform certain motor acts (like grasping) facilitates the recognition of similar actions performed by others. This extends the concept of learning through observation a bit further. Attention is now directed toward actions performed upon some objects instead of our own movements.

This ability to recognize other people's gestures is inspired by the discovery by neurophysiologists of the Mirror Neurons in the monkeys pre-motor cortex. Mirror neurons respond both during the execution and observation of a goal-oriented gesture. Furthermore, this observation suggests that recognition is performed in the motor space and not in the visual space.

Another population of neurons - the Canonical Neurons - discharges with the execution of a grasp action and the observation of an object that can be grasped in that particular way, representing the affordances of those objects. Hence both types of visuomotor neurons represent a congruence between the execution of motor actions and the observation of those actions/objects.

We propose a Bayesian formalism that models the observations of mirror and canonical neurons and shows the advantage of performing recognition in motor terms when compared to visual terms, making use of the visuomotor maps learned in previous developmental stages.

By recognizing other people's action upon objects, the system can learn about new object properties and new affordances of known objects, e.g. stable grasps. In the next chapter we will further extend this learning-from-observation metaphor to the ability to imitate other people's gestures, as the system gets more socially involved and ready to start looking at humans or other robots and the tasks they perform.

Chapter 4

Imitation

In this chapter, we present several algorithms enabling a robot to learn by imitation. These algorithms are built on top of skills learned in previous stages of the proposed development roadmap.

As before, we stress the fact that an artificial system can retrieve a tremendous amount of knowledge, simply by looking at other individuals, humans or robots, working in the same area. In the previous stages of development, observation allowed the robot to acquire body control and explore the properties of reachable objects. Now, we will see how a robot can learn how to perform tasks by imitating what others are doing, a procedure routinely adopted by humans.

We assume that, at this point of development, the system has already acquired a vast set of skills, as described in the previous chapters. By self-exploration, it learned how to coordinate its own body. It became capable of achieving desired positions and/or motions of its body. This capability is fundamental for imitation because the robot knows how to move its body to cause the same effect as an observed action.

The second developmental stage equipped the system with important capabilities to interact with the world: localize and manipulate objects, learn their properties and recognize actions performed by others. This capability of recognizing other people's actions is enhanced by knowledge about its own body but also about its own goals. It is very important to infer what the others are observing, their goals and believes. This "perspective tacking" gives a priori information to recognize what others are doing and create and implicit level of communication between agents.

Now, in the third developmental stage, the interaction and imitation of other people in the world will be increased. Figure 4.1 shows our proposed architecture for imitation learning. When observing a task demonstration, the first component to be used is a visuo-motor process called the view-point transformation (VPT). It converts the observed motion to an ego-centric frame of reference, through a "mental rotation" of the demonstrator's body.

A second issue to address is the "body correspondence" problem that analyzes

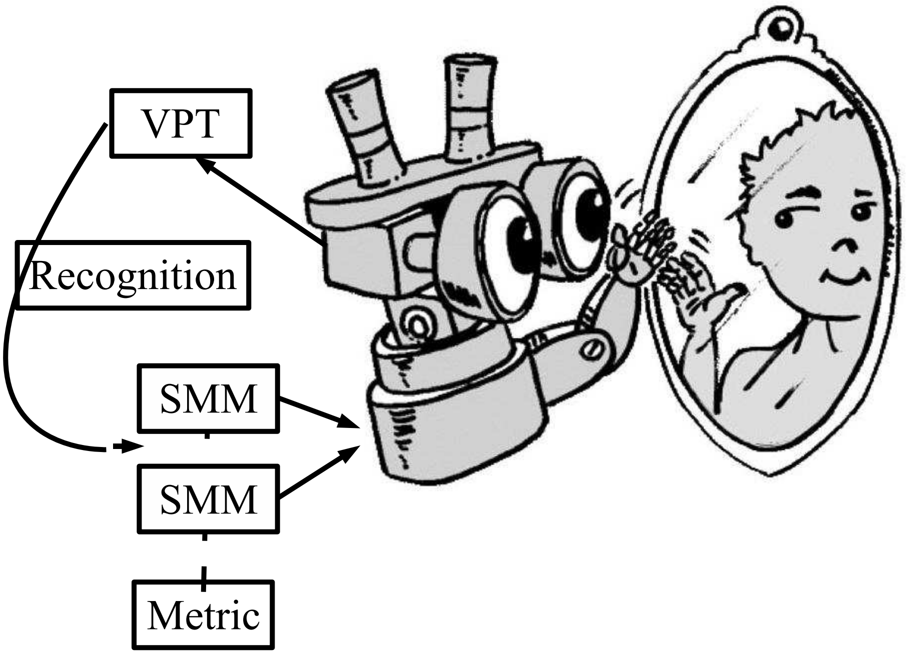

Figure 4.1: The imitation process consists of the following steps: (i) observation of the demonstrator's actions; (ii) a view-point transformation (VPT) is used to transform these image coordinates to the *ego-image*, as proposed in Section 4.1; (iii) task recognition and/or abstraction (iv) with the chosen metric the SMM generates the adequate joint angle references to execute the same task.

how to transform the motion seen in one specific body to a different body. This point is particularly relevant when the demonstrator and learner have very different kinematics. In this thesis, we solve the correspondence problem implicitly, using the skills learned in previous developmental stages.

Another aspect worth mentioning is that the ability to recognize the actions and goals of others, enables different imitation behaviors. In other words, the word imitation can be used to describe different phenomena. We will distinguish two main different imitative behaviors: action-level and program-level imitation. In action-level imitation, the goal consists in replicating the body trajectories of the demonstrator. Instead, program-level imitation consists in replicating the actions or goals of the demonstrator but not necessarily the exact motion.

This chapter presents solutions for both mechanisms. With the learned sensory-motor coordination it is possible to do action-level imitation. Program-level imitation involves the interpretation or recognition of actions in order to infer the action's goals. This kind of imitation requires abstracting high-level information from the observation of a task, as opposed to raw kinematic information. When observing an object manipulation task, the learner's attention is drawn toward the task objective and not to the underlying movements. Hence, program-level imitation is the natural choice in such a context.

Usually, choosing between among imitation behaviors and the evaluation of the resulting quality of imitation is done by selecting a proper “imitation metric”. As the metric is intimately related with the body correspondence solution, this choice will restrict the possible sensory-motor maps to be used to do the final imitation of the task. The goal of imitation is chosen by the imitator itself by selecting among the available metrics.

Figure 4.2 presents a demonstrator and two different imitation behaviors. The hand and feet position are correctly matched but there is a difference in the elbow and knee position. This difference appears due to the use of different body correspondence mechanism. The whole body is considered in one case but not in the other. In a sense, both responses are correct, the difference arising from the use of different imitation metrics.

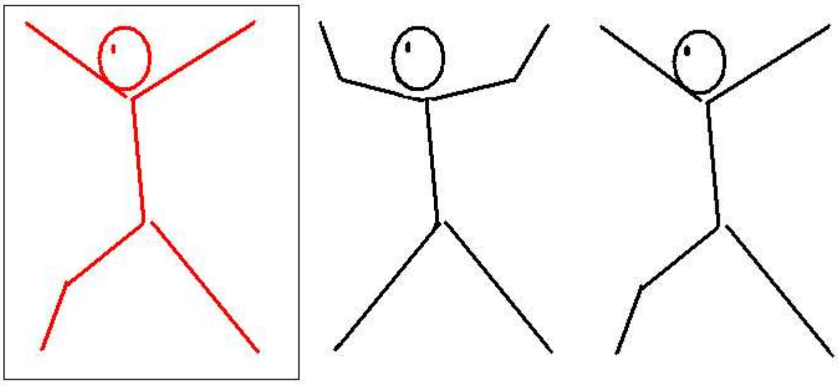

Figure 4.2: Different imitation behaviors can be obtained if different sensory-motor maps are used to do the body correspondence. The leftmost image is the demonstrator. The imitation behavior can achieve a full body mapping (rightmost) or just hands and feet position mapping, the position of elbows and knees being irrelevant.

Figure 4.3 presents different behaviors in imitation. On the top it is possible to see that a difference in the VPT can lead to ipsi vs contra-lateral imitation. On the bottom we can see that imitation can go along more abstract term as “kick the ball”, where the focus is on the task objective as opposed to individual movements.

4.1 View-Point Transformation

A certain arm gesture can be seen from very different perspectives depending on whether the gesture is performed by the robot (self-observation) or by the demonstrator.

One can thus consider two distinct images: the *ego-centric* image, I_e, during self-observation and the *allo-centric* image, I_a, when looking at other robots/people. The role of the *View-Point Transformation* (VPT) is to align the allo-centric image of

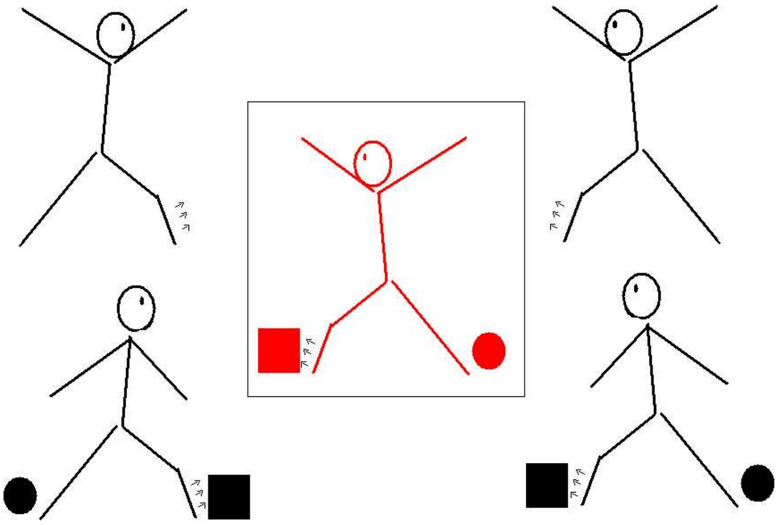

Figure 4.3: Different imitation behaviors obtained by appropriate selection of different view-point transformations and abstracting or not the observed behavior. The center image represents the demonstration. The top images represent and action-level imitation where only the motion is imitated with the difference being an ipsi or contra-lateral imitation. In the bottom figures, the imitation is abstracted to “kick the square” where the arm positions and left/right is not important.

the demonstrator’s arm, with the ego-centric image, as if the system were observing its own arm [Lopes and Santos-Victor, 2003].

This mechanism allows the system to do a “perspective taking” by considering itself in the demonstrators place. Other imitation approaches like [Kuniyoshi et al., 2003, Demiris et al., 1997] do not do this perspective taking because imitation is achieved by a direct coupling between perception and action. In spite of the fundamental importance of the VPT in visual perception and in the psychology of imitation [Bruner, 1972], it has received relatively little attention by researchers in robotics.

The precise structure of the VPT is related to the ultimate meaning of imitation. Experiments in psychology show that imitation tasks can be ambiguous. In some cases, humans imitate only partially the gestures of a demonstrator (e.g. replicating the hand pose but having a different arm configuration, as in sign language), use a different arm or execute gestures with distinct absolute orientations [Rochat, 2002, Meltzoff, 1988]. In some other cases, the goal consists in mimicking someone else’s gestures as precisely as possible, as when performing dancing or dismounting a complex mechanical part. This is closely related with the body correspondence and metric to be used. Conversely, the choice of the VPT will restrict the possible metrics and/or correspondences.

According to the structure of the chosen VPT, a class of imitation behaviors can be generated. We consider two different cases. In the first case - 3D VPT - a complete three-dimensional imitation is intended. In the second case - 2D VPT - the goal consists in achieving coherence only in the image, even if the (3D) arm pose might be different. Depending on the desired level of coherence (2D/3D) the corresponding (2D/3D) VPT allows the robot to transform the image of an observed gesture to an equivalent image as if the gesture was executed by the robot itself.

4.1.1 $3D$ View-Point Transformation

In this approach we reconstruct the posture of the observed arm in 3D and use fixed points (shoulders and hip) to determine the rigid transformation that aligns the allo-centric and ego-centric image features: We then have:

$$\mathcal{I}_e = \mathbf{P}\,\mathbf{T}\,Rec(\mathcal{I}_a) = VPT(\mathcal{I}_a)$$

where $\mathbf{P}$ is the camera projection matrix, $\mathbf{T}$ is a 3D rigid transformation and $Rec(\mathcal{I}_a)$ stands for the 3D reconstruction of the arm posture from allo-centric image features. Posture reconstruction and the computation of $\mathbf{T}$ are presented in the following sections.

Posture reconstruction

To reconstruct the 3D posture of the observed arm, we will follow the approach suggested in [Taylor, 2000], based on the orthographic camera and articulated arm models presented in Annex A. In an orthographic camera the image is formed by assuming parallel projection of the scene plus a scaling.

Let $\mathbf{M} = [x\ y\ z]^T$ denote a 3D point expressed in camera coordinates. Then, with an orthographic camera model, $\mathbf{M}$ is projected onto $\mathbf{m} = [u\ v]^T$, according to:

$$\begin{aligned} \mathbf{m} &= \mathbf{PM} \\ \begin{bmatrix} u \\ v \end{bmatrix} &= s \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \end{aligned} \tag{4.1}$$

where s is a scale factor that can be estimated placing a segment with size L fronto-parallel to the camera and measuring the image size l ($s = l/L$).

Let $\mathbf{M}_1$ and $\mathbf{M}_2$ be the 3D endpoints of an arm-link whose image projections are denoted by $\mathbf{m_1}$ and $\mathbf{m_2}$. Under orthography, the x, y coordinates are readily computed from image coordinates (simple scale). The depth variation, $dz = z_1 - z_2$, can be determined as:

$$dz = \pm\sqrt{L^2 - \frac{l^2}{s^2}}$$

where $L = \|\mathbf{M}_1 - \mathbf{M}_2\|$ and $l = \|\mathbf{m}_1 - \mathbf{m}_2\|$.

If the camera scale factor s is not known beforehand, one can use a different value provided that the constraint, involving the relative sizes of the arm links, is met:

$$s \geq \max_i \frac{l_i}{L_i} \quad i = 1..4 \tag{4.2}$$

where L_1 and L_2 denote the arm and forearm length, L_3 designates the shoulder width and L_4 stands for the torso height (refer to Figure 2.3 for more details).

Figure 4.4 illustrates results of the reconstruction procedure. It shows an image of an arm gesture and the corresponding 3D reconstruction, achieved with a single view and considering that s and the arm links proportions were known in advance. This is interesting because the mapping between bodies of different sizes is made directly.

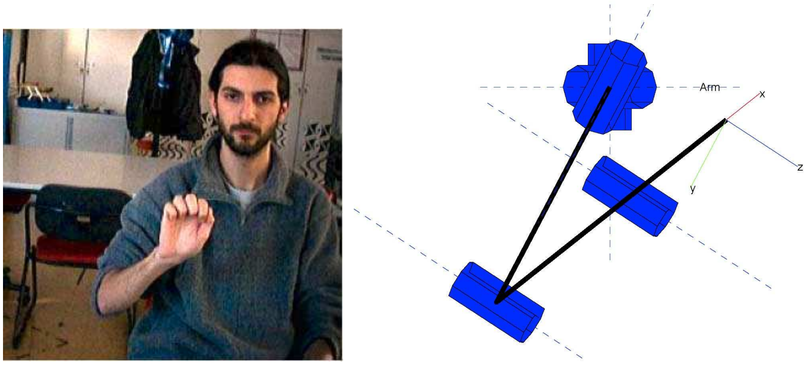

Figure 4.4: Final result using the proposed VPT and the learned sensori-motor map Left: Original view. Right: Reconstructed arm posture.

With this method there is an ambiguity in the sign of dz. We overcome this problem by restricting the working volume of the arm.

Rigid Transformation (T)

A 3D rigid transformation is defined by three angles for the rotation and a translation vector. Since the arm joints are moving, they cannot be used as reference points. Instead, we consider three points: left and right shoulders, $(\mathbf{M}_{ls}, \mathbf{M}_{rs})$ and hip, $\mathbf{M}_{hip}$, with image projections denoted by $(\mathbf{m}_{ls}, \mathbf{m}_{rs}, \mathbf{m}_{hip})$. The transformation T is determined to translate and rotate these points until they coincide with those of the system's own body.

The translational component must place the demonstrators right shoulder at the image origin (which coincide's with the system's right shoulder) and can be defined directly in image coordinates:

$$\mathbf{t} = -{}^{\mathbf{a}}\mathbf{m_{rs}}$$

After translating the image features directly, the remaining steps consist in determining the rotation angles to align the shoulder line and the shoulder-hip contour.

The angles of rotation along the z, y and x axes (refer to Figure A.4), denoted by ϕ, θ and ψ are given by:

$$
\begin{aligned}
\phi &= \arctan\left(v_{ls}/u_{ls}\right) \\
\theta &= \arccos\left(u_{hip}/L_4\right) \\
\psi &= \arccos\left(v_{hip}/L_3\right)
\end{aligned}
$$

A 3D rotation can be estimated from the proportions between shoulder width and head-shoulder height. Hence, by performing the image translation first and the 3D rotation described in this section, we complete the process of aligning the image projections of the shoulders and hip to the ego-centric image coordinates.

4.1.2 $2D$ View-Point Transformation

The $2D$ VPT is used when one is not interested in imitating the depth variations of a certain movement, alleviating the need for a full 3D transformation. It can also be seen as a simplification of the 3D VPT, assuming that the observed arm describes a fronto-parallel movement with respect to the camera.

The $2D$ VPT performs an image translation to align the shoulder of the demonstrator (${}^{\mathbf{a}}\mathbf{m}_{\mathrm{s}}$) and that of the system (at the image origin, by definition). The VPT can be written as:

$$
VPT({}^{\mathbf{a}}\mathbf{m}) = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \left[{}^{\mathbf{a}}\mathbf{m} - {}^{\mathbf{a}}\mathbf{m}_{\mathrm{s}}\right] \tag{4.3}
$$

and is applied to the image projection of the demonstrator's hand or elbow, ${}^{\mathbf{a}}\mathbf{m}_{\mathbf{h}}$ or ${}^{\mathbf{a}}\mathbf{m}_{\mathbf{e}}$.

Notice that when the arm used to imitate is the same as the demonstrator, the imitated movement is a mirror image of the original. If we use a identity matrix in Equation (4.3) then the movement will be correct. At the image level both the 2D and 3D VPTs have the same result but the 3D posture of the arm is different in the two cases.

From the biological standpoint, the 2D VPT is more plausible than the 3D version. In [Rochat, 2002] several imitation behaviors are presented which are not always faithful to the demonstrated gesture: sometimes, people do imitate with the contra-lateral hand, depth is irrelevant in some other cases, movements can be reflections of the original ones, etc. The 3D VPT might be more useful in industrial facilities where gestures should be reproduced as faithfully as possible.

4.2 Imitation Metrics and Body Correspondence

In this section, we present the metrics used in this thesis to evaluate and guide imitation. Two different sets of metrics are presented for the cases of action-level and program-level imitation (in the context of an object manipulation task).

In action-level imitation, the problem of body correspondence is implicitly solved via an adequate choice of the Sensory-Motor Map, instead of transforming the observed trajectories to the new embodiment. In fact, the SMM captures the knowledge the system has about gestures that produce the same (visual) effects as those observed.

For the case of program-level imitation, the situation is slightly different because the task description is formulated in a more symbolic manner. Hence, the correspondence is solved by the system in two steps. First, the observed grasp action is recognized. Then, a compatible grasp action is executed based on the grasping algorithm learned in earlier developmental stages. Chapter 3 describes in detail both the grasping method and the action recognition algorithm.

Action-level imitation

Gestures are a very important mean of communication. They are used to wave someone goodbye or hello, or to make some warnings like: you're out of time, everything is fine. Although the gesture itself can be produced in a variety of different ways, the meaning is almost always unambiguous and recognition and understanding will be relatively easy. When waving goodbye, the speed or the exact distance between the hand and the head are not critical. Instead, the situation is quite different for sign languages. As the set of symbols is very large, small variations can correspond to a large change in meaning.

We have discussed that the choice of the metric and the viewpoint transformation are extremely intertwined. If a metric is defined in 3D terms, it is not possible to use a VPT that expresses a partial transformation (e.g two dimensional). Therefore, in the general imitation architecture the metric is the first thing that needs to be defined, then, all the rest follows. The following equations are examples of 2D/3D metrics used for action-level imitation:

$$\text{3D imitation:} \qquad im_{3D} = \int \left(VPT_{3D}(\mathcal{I}_a) - \mathcal{I}_e^{self} \right) dt$$

$$\text{2D imitation:} \qquad im_{2D} = \int \left(VPT_{2D}(\mathcal{I}_a) - \mathcal{I}_e^{self} \right) dt$$

In these equations, $\mathcal{I}_a$ denotes the image of the demonstrator seen by the imitator (allo-image) and $\mathcal{I}_e^{self}$ represents the image of the imitator's body as seen by itself (ego-image). With these metrics the imitator is asked to move its body in order to match as closely as possible the position of the demonstrator. The great advantage of the VPT becomes now very clear. Because of the ego-representation of the gestures, all the sensory-motor coordination mechanisms learned in the first development stage can now be used. To imitate according to a given metric, the body correspondence problem is solved through the selection of the appropriate SMM.

Program-level imitation

A different type of task involves acting on objects. Considering tasks like placing dishes on a table or storing books in a shelf, it is clear that the key issue here does not reside on pure gesture imitation. For these examples the most important part is the final state (or goal) of the task, the way in which the task is solved, i.e. the posture, the speed, is not relevant. This calls for different metrics than the ones we have seen before. The actions and movements of the demonstrator must be segmented and coded in a way meaningful for imitating the task goals and sub-goals.

We developed a method consisting in a multiple object tracking and a action detector. In a manipulation task, our hand will occlude objects very frequently. Grasping and releasing can be very difficult to detect. The fact that the hand is the only active element in the scene provides some implicit information that will help dealing with occlusions. We assume that every object can have two movement models: "rest" and "moving". When an object is being moved, it has the same velocity as the hand. Object grasping is detected in two situations: i) when it starts to move, ii) when it is occluded by the hand. Detecting object releasing is done by detecting a previously grasped object becoming static while the hand moves away. Using these hypotheses, our algorithm will mark every grasping/releasing point in the trajectories of the objects. Figure 4.5 gives a finite-state machine that controls the detection of object state. The process of task segmentation is illustrated in Figure 4.6. If the grasping type is important, the grasping classification method presented in section 3.2 could be used.

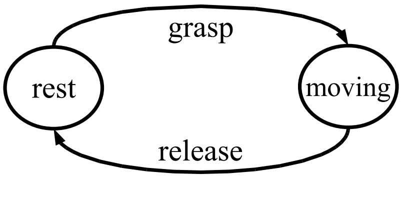

Figure 4.5: Object state transtions.

The task is then codified in a sequence of world states, the transitions between states are done by grasp or release a given object. Each state describes the objects spatial relations (A between B and C;A right of B or A left of B) and metric positions.

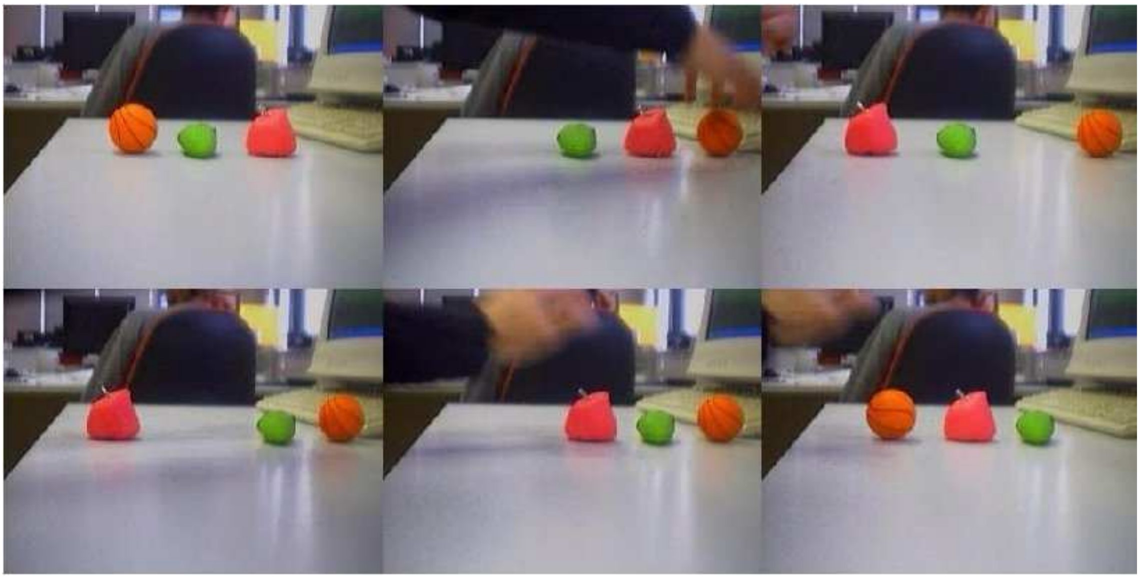

Figure 4.6: Task Segmentation. Notice that from the third to the fourth image there is no difference in the ordering of the object, just their absolute distances. These relevant points were extracted online from a video sequence with 234 frames.

Although this approach cannot be seen as a general framework for goal-directed, program-level imitation, it is noteworthy to mention that the goals and sub-goals of certain tasks can be abstracted in this way. As a consequence, a rich imitation behavior is achieved, following the proposed developmental roadmap. The summary of the algorithm is summarized below.

1. Detect and localize objects around the demonstrator and apply the VPT to map those objects in the observer's coordinate frame.

2. Observe the sequence of task execution.

3. Segment the sequence by detecting interesting points (changes in the tracker state), in time and space.

4. Make a description of the task, as a chain of meaningful events such as grasp and release objects.

5. Perform the same task.

4.3 Experiments

We have implemented the modules discussed in the previous chapters to build a system able to learn by imitation. We start by describing the approach used for hand-tracking before presenting the overall results on imitation. In the following sections we present results both for action and program-level imitation.

4.3.1 Vision

Vision system

In order to model the arm position of the demonstrator, we have three steps of segmentation for the background, person and hand.

During initialization, the background is estimated by modeling the intensity of each pixel as a Gaussian random variable. We need about 100 frames to obtain a good model. After this process, we can estimate the probability of each pixel belonging to the background. In order to increase the robustness of segmentation to illumination variations, we use RGB color representation normalized by the blue channel.

Having a model of the background, we can determine the areas in the image where motion has been observed. After detection, the position of the person is estimated by template matching and correlation. The template consists of a rectangle for the body, on top of which, a second rectangle represents the head. The body-head proportions used were those corresponding to a fronto-parallel person at a nominal distance from the cameras. By scaling the template, we can estimate the size of the person and the scale parameter, s, of the camera model. In addition, if we need to detect if the person is rotated with respect to the camera, we can scale the template independently in each direction, and estimate this rotation by the ratio between the head height and shoulder width.

To detect the hand, we used a skin-color segmentation process. Figure 4.7 shows the result of this process.

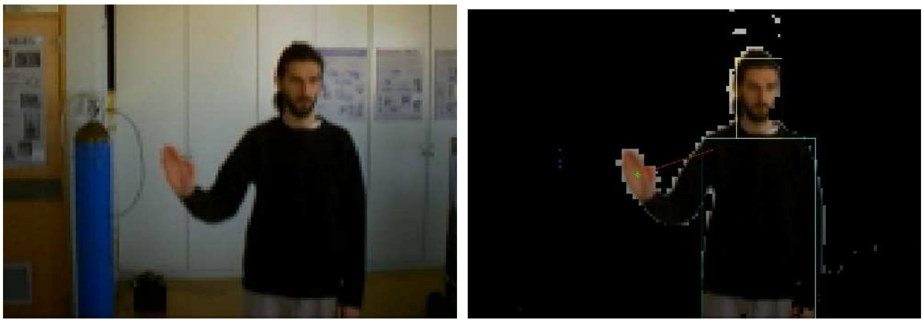

Figure 4.7: Vision system. Left: original image. Right: background segmentation with human (the rectangular frame corresponds to the template matching) and hand detection.

Skin Color Segmentation

To find the hand in the image we use a color segmentation scheme, implemented by a feed-forward neural network with three neurons in the hidden layer. As inputs we use the hue and saturation channels of HSV color representation. The training data are obtained by selecting the hand and the background in a sample image. After color classification a *majority* morphological operator is used. The hand is identified as the largest blob found and its position is estimated over time with a Kalman filter. Figure 4.8 shows a typical result of this approach.

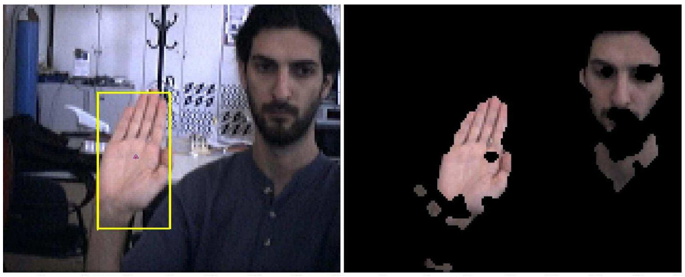

Figure 4.8: Skin color segmentation results.

4.3.2 Action-level imitation

The first imitation experiments deal with action-level imitation. Here gestures made by a person should be repeated by the robot. Using the generic architecture of Figure 4.1 the robot observes the scene using the person/hand tracking system presented earlier. After choosing the metric the robot applies the correct VPT and then the previously learned sensory-motor map than give directly the necessary motor commands.

Figure 4.9 shows experimental results obtained with the $3D$-VPT with the learned SMM (full-arm). To assess the quality of the results, we overlaid the images of the executed arm gestures (wire frame) on those of the demonstrator. The figure shows that the quality of imitation is very good, in the sense that the motion of the demonstrator is reproduced faithfully.

Figure 4.10 shows results obtained in real-time when using the $2D$ VPT and the *free-elbow* SMM. Here, the system succeeds in imitating the hand gesture but, as expected, there are differences in the configuration of the elbow, particularly at more extreme positions.

If we assume that the movement of the hand is constrained to a plane or that the depth changes are small, we can use the proposed view-point transformation to estimate the position of the person. The system is able to imitate the tutor in real-time. Some results are shown in Figure 4.11.

4.3.3 Program-level imitation

The goal of the imitation task illustrated here consists on moving a set of objects, as shown by a demonstrator. It follows the imitation system presented in Sec. 4.2.

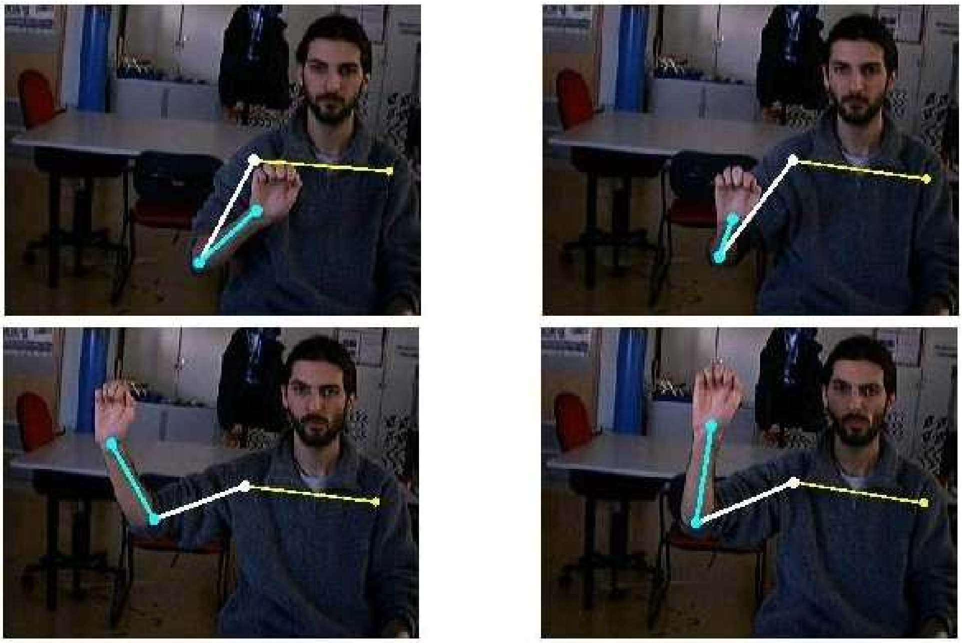

Figure 4.9: The quality of the results can be assessed by the resemblance of the demonstrator gestures and the result of imitation.

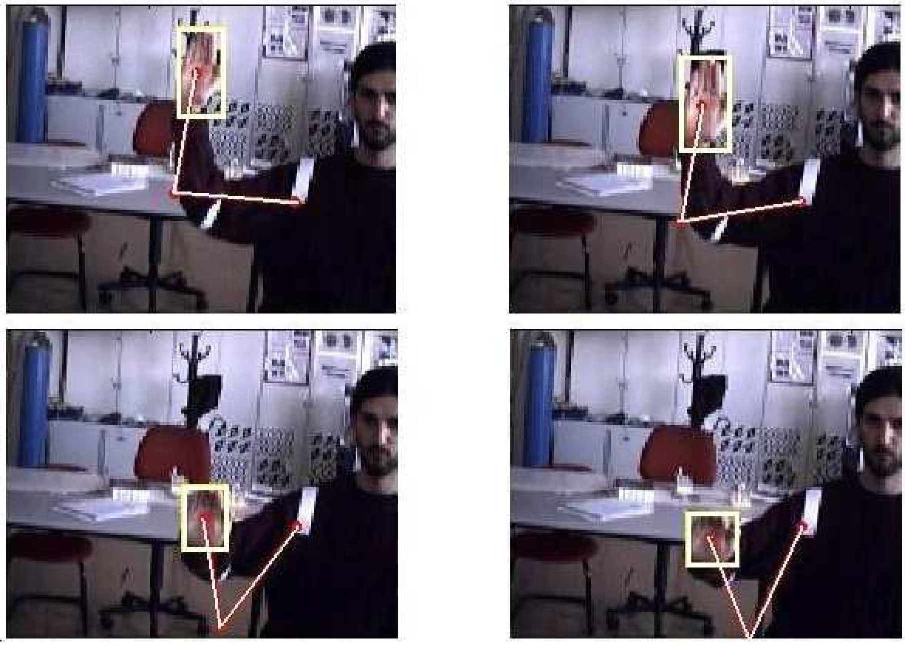

Figure 4.10: Family of solutions with different elbow angles, while the hand position is faithfully imitated.

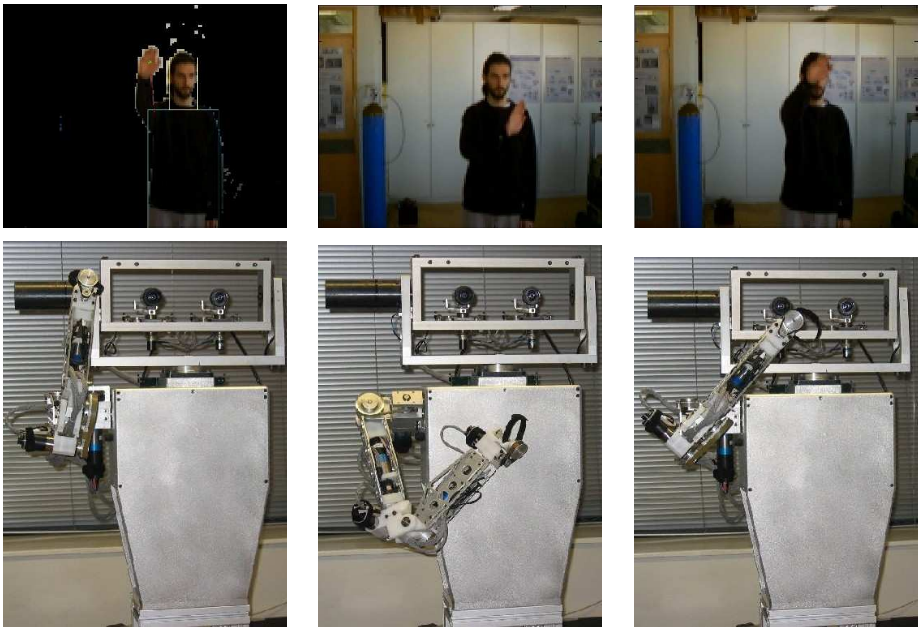

Figure 4.11: Robot imitating the hand movements made by a demonstrator. A partial SMM is used in such a way that the elbow position is left unconstrained.

We can see that all the modules developed until this point are essential to replicate the task at hand. Although the way we describe this particular set of tasks could be replaced by possibly more sophisticated processes, the modules would still remain as valid building blocks to perform such a new set of tasks.

Figure 4.12 shows an example of the execution of a task, consisting of grasping a set objects and moving them around. To imitate this task, the robot first needs to understand the spatial relations of objects in the vicinity the demonstrator (understand the far space). Then, understanding the near space becomes fundamental to establish correspondence between the demonstrator perspective and its own (self) viewpoint (i.e. the blue object located on the left hand side of the demonstrator is in front of me). After the observation of the demonstrator's movements, the important task moments must be extracted and temporally segmented.

Finally, the learned task is repeated by the robot (Figure 4.13), using the imitation architecture and the developmental pathway proposed in the thesis. The robot places the objects in the same order as the demonstration. In the final step, the robot assumes that the task sub-goal consists in changing the absolute position of one object, since the demonstrator did not affect the objects relative spatial relations.

The task interpretation and execution is the following:

1. By moving the head, detect objects A and B (on the right and on the left of the demonstrator)

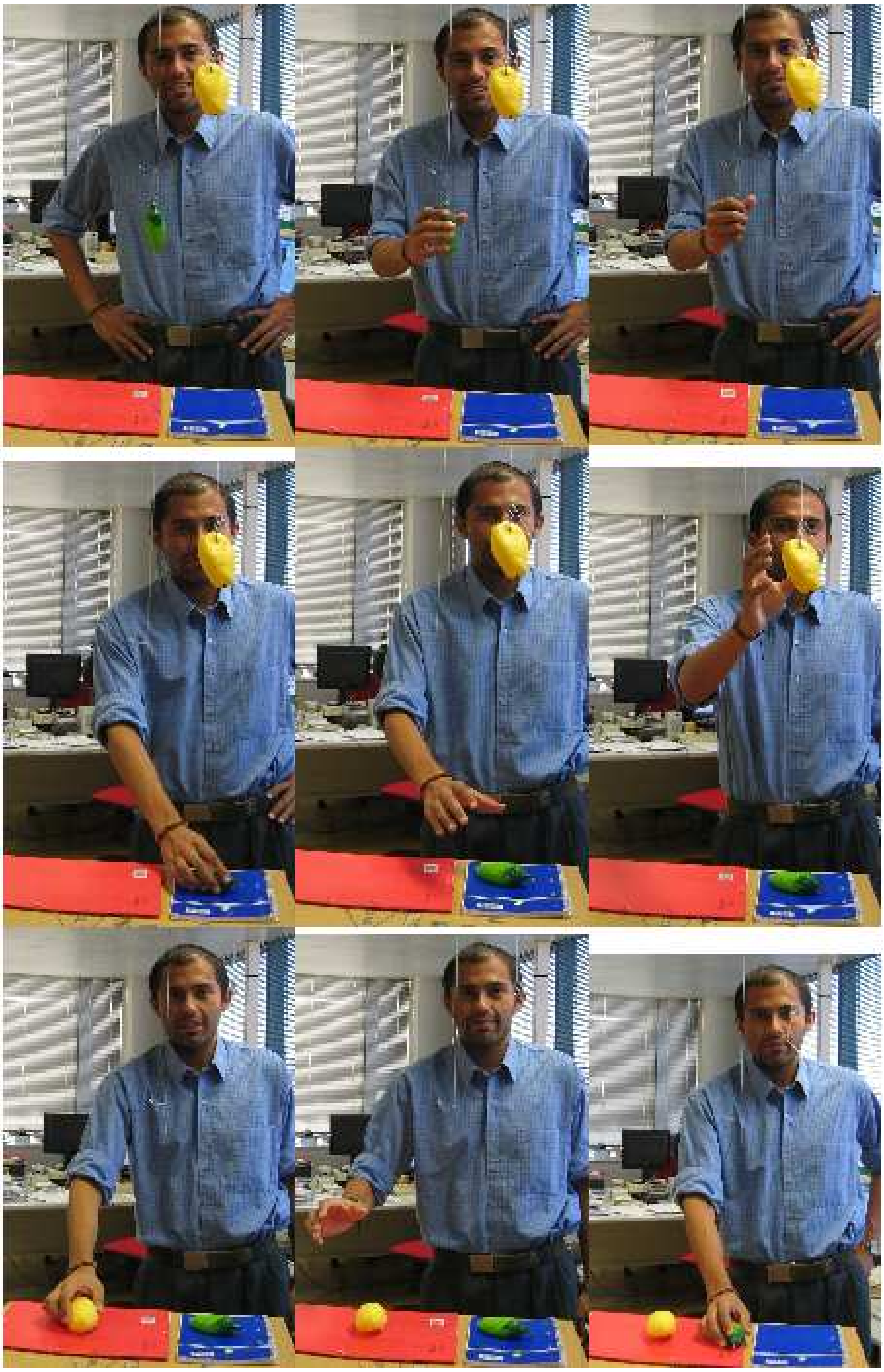

Figure 4.12: Several frames of the task demonstration. The person is moving objects from position to position.

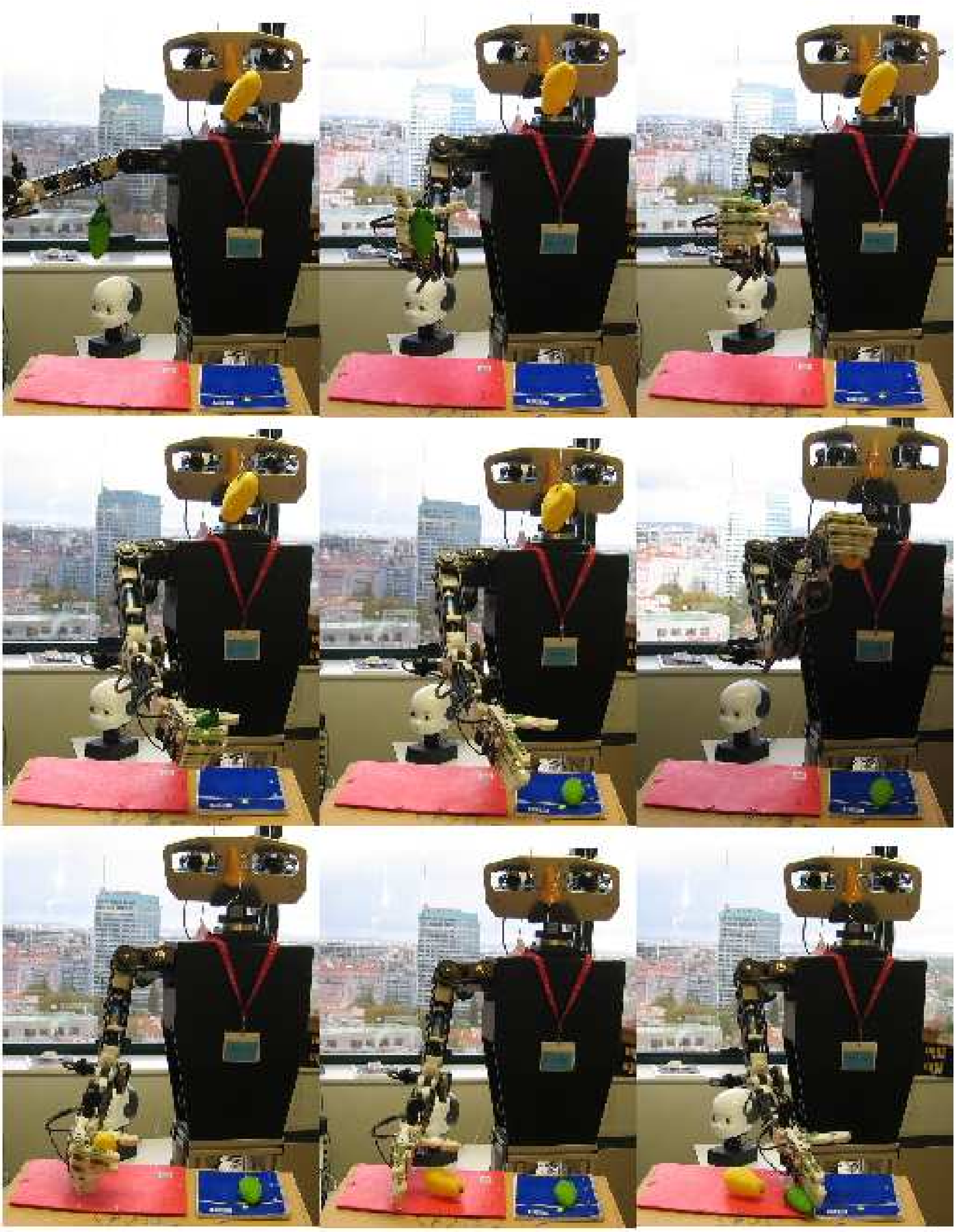

Figure 4.13: Repetition of the task by the robot. Two different metrics are being used here: object ordering and metric position. The robot localizes and grasps object using the behaviors learned in previous developmental stages.

2. Foveate on object A, Grasp Object A
3. Foveate on position 0, Release Object A
4. Foveate on object B, Grasp Object B
5. Foveate on position 1, Release Object B
6. Foveate on object A, Grasp Object A
7. Foveate on position 2, Release Object A

To note that all positions are restricted to a vertical plane. To grasp and foveate the head moves directly to the position where the objects were detected. This step facilitates the control because the target is in the center of the image, but is also a necessity due to the limited field-of-view of the robotic head. To grasp (or release) a static head-arm map is used to move the arm to the image and near the target position. Then a visual servoing loop starts to move the hand closer to the target. Upon contact the hand closes.

4.4 Conclusions

We proposed an approach for learning by imitation that relies exclusively on visual information. The tests done show that good results are obtained with the proposed framework under realistic conditions.

One of the main contributions is the *View-Point Transformation* that performs a “mental rotation” of the image of the demonstrator's arm to the *ego-image*, as if the system were observing its own arm. We described two different VPTs needed for 3D or 2D imitation. The *View-Point Transformation* can have an additional interest in the study of the *Mirror Neurons*, by providing a canonical frame of reference that greatly simplifies the recognition of arm gestures.

The observed actions are mapped into muscles torques by the *Sensory-Motor Map*, that associates image features to motor acts. Again two different types of SMM were involved, depending on whether the task consists of imitating the entire arm configuration or the hand position only. The SMM was learned in a previous developmental stage, and is used to directly solve the correspondence problem. For program-level imitation there is the additional step of recognizing/detecting the grasp action during the observation of the task.

Several metrics were presented that can evaluate imitation at several levels. We have presented results with our experimental platform both for action and program-level imitation. For program level imitation, we have proposed a method to temporally segment a task in its key frames, that will then be reproduced by the robot.

In the majority of the examples presented in this chapter, the imitation goal is related to the use of the hand for manipulation and not to place the arm in a specific (global) posture. For this reason, there are redundant degrees of freedom not involved in the imitation process. As we proposed in Chapter 2, those extra degrees of freedom can be incorporated in an auxiliary metric to meet additional performance criteria (e.g. energy minimization, additional tasks, obstacle avoidance, posture matching, etc).

Chapter 5

Conclusions/Future Work

In this thesis, we have presented a developmental process for a humanoid robot to acquire increasingly more complex skills, over time, through the interaction with the environment, objects and other individuals. The long-term goal consists of endowing a robot with the skills necessary to work in environments designed for humans (as opposed to industrial) and to interact with them in a friendly and yet powerful manner. Our approach is inspired in two main paradigms: *imitation learning* and *artificial development.*

Imitation learning offers a very intuitive way for people to “program” robots to perform complex tasks, as they can simply show the robot what to do. With this capability, robots can accumulate information just by observing what is happening around them. This process of knowledge transfer through imitation is used intensively by human infants (and possibly other primates) in the first years of life. Although the benefits of learning through imitation are clear, the problem is by no means an easy one, specially when considering the use of sophisticated (in terms of motor and perceptual capabilities) humanoid-type robots.

The *artificial development* approach aims to handle the complexity of both the task and the system, inspired after the development of living beings and experiments in developmental psychology. The main motivation is that the solutions for simpler (sub-)problems can be combined together to solve more complex ones. Also, a partial solution for some problems can be improved, as the system capacities increase. Finally, development requires interaction with the world at all moments, where the significance of “world” changes according to the level of cognitive development of the robot (or human infant).

Our development agenda is based on three main phases: *sensory-motor coordination*, *world interaction* and *imitation.* In the first level, the robot learns how to control its own body through self-exploration. A variety of multimodal sensory-motor maps are learned that allow to control and coordinate the high complex motor and perceptual system of the robot. In the world interaction phase, the robot is attracted to salient objects in the visible world. It learns how to grasp objects in a (visually) controlled way and creates a map of the interesting objects in the surrounding space. Additionally, the system learns how to recognize the (grasp) actions of others which, in turn, provides the system with information regarding actions that can be exerted upon objects or how to use such objects. In the final developmental phase, people acting in the environment are the major source of information and draw most of the system's attention. The observed tasks are temporally segmented in special points to create a sort of abstract plan. This plan can then be used when the robot is asked to execute that task.

It is worth stressing that each module in the developmental route is built upon skills already learned in precedent levels. As time goes by, the system learns how to address more complex tasks in an incremental fashion. Also, at each level, the system is driven by some motivations (control its own body, understand object properties or the actions of others) and learns new skills (how to grasp, how to recognize actions, how to imitate). In the following paragraphs we will review the main aspects addressed at each developmental level.

Sensory-motor coordination

Sensory-motor maps describe the relationship between the system motor degrees of freedom and its perceptual channels. These maps are bidirectional, and allow the system to predict the results of a motor action or, conversely, the action needed to produce a certain perceptual output. These maps can be used in open-loop (static maps) or closed-loop (incremental maps) situations, as appropriate.

We have addressed the key problem of estimating these maps in redundant systems, which is almost always the case for humanoid robots. Due to redundancy, the inverse map cannot be estimated since the forward model is not bijective. The methods we propose are simple, computationally efficient and well suited for online learning.

We define a partial sensory-motor map that only involves the non-redundant degrees of freedom. The redundant degrees of freedom remain available to meet some additional tasks or performance criteria. An interesting observation is that this partitioning improves both the learning and the control.

In the future, we plan to investigate automatic methods for the division between redundant and non-redundant degrees of freedom. One possible direction of research can go along a similar path as the one presented in [Hosoda and Asada, 2000].

World interaction

We present a method for learning how to grasp objects that does not require calibration, and that relies on the sensory-motor maps learned beforehand. It consist in a first open-loop phase, followed by a final more accurate phase, under visual closed-loop control.

We proposed a framework for gesture recognition based on a model for *canonical and mirror neurons*, that seem to play a fundamental role for grasp recognition and imitation in primates. We propose a bayesian formulation, where all these observations are taken into account. We describe how to estimate the prior density and likelihood functions directly from data. The results show that it is possible to achieve high recognition rates based on this approach. This behavior was learned using knowledge about the system own capabilities of grasping and visuo-motor maps between hand appearance and motor control.

These results illustrate the fact that neurophysiology can be a source of inspiration for engineers to build better (cognitive) artificial systems. On the other hand, designing artificial systems, grounded on such biological principles, is a valuable means of validating hypotheses or theories in biology.

Imitation

We presented a general approach for action and program level imitation. In the former, the goal is to reproduce the demonstrator's gestures as faithfully as possible, while in the latter the objective consists in achieving the task goals, independently of the way the gestures are executed.

One of the main contributions here is the *View-Point Transformation* (VPT) that performs a "mental rotation" to align the image of the demonstrator's arm with the system's body coordinate frame. In spite of the fundamental importance this mechanism has in visual perception and in the psychology of imitation [Bruner, 1972], it has received little attention by researchers in robotics. We described several VPTs to be selected according to the metrics underlying the task imitation problem. Different metrics were discussed in the context of action and program level imitation.

An important component in imitation is body correspondence. In our approach, this problem is solved in an indirect manner. After learning several sensory-motor maps, the robot is able to produce similar effects (defined at several levels), without the need to explicitly make a correspondence between the trajectories and/or actions of the demonstrator and the learner.

For program-level (goal-directed) imitation, an additional module was necessary to abstract the description of a task from observation. In the context of a task of object manipulation, we proposed a method to identify key frames and states during the observation of the task execution. This description is used later on for the robot to reproduce that same task.

Task description and selection of the relevant information is, with the present knowledge, made by the designer but it would be of great benefit to automatically extract it from observation. This task descriptions should not be pre-programmed, because, although with some work a very complex description can be obtained, it is more important to be able to extract a description from a completely new world setting. All the imitated tasks should be learned in a way allowing to apply the same solutions to other similar problems.

Future Work

Several improvements in other areas will also help our endeavor. Probably most contributions will come from the domain of computer vision. A better understanding of human activity recognition can help endowing robots with more sophisticated imitative capabilities. In turn, some of our ideas in imitation studies can help research on human activity recognition from video.

Also some behaviors learned in this work can be further improved. As for grasp control, research on bi-manual manipulation and coordination is starting just now and it will give new exciting opportunities and applications in the years to come.

Imitation Imitation learning should be improved in many ways before we can deploy systems able to work robustly in the real world. Although we, and several other authors, have discussed and suggested several different imitation metrics, few works have dealt with the problem of metric selection and/or inference. Until now the imitation goal is selected by design. Biological systems have several imitation like behaviors that have several evolutionary advantages. To obtain the full advantages of imitation, the artificial systems should use imitation as another mechanism in its repertoire of action selection and always use it in relation with another objective for the agent: e.g. survival, initiating some interaction or speeding up learning. So an agent, operating in a complex environment and subject to longer interactions, will require the metric to be selected according to longer-term objectives of the robot. This problem of *when and what to imitate* will only be solved when integrated in a real autonomous system.

Without being the objective the creation of an *artificial human*, the long term goal of a robotic system can be prolonged user interaction and a fast learning rate. The robot would use imitation to interact with its user, confirming that the information it learned was the correct. Also, all the time spent observing others must be used in task learning.

Development With several learning mechanisms running at the same time, our approach may exhibit some oscillations. This is already known in neurophysiology and control theory but, usually, these mechanisms have very different time constants so that the problem can be minimized.

One axis of development that could be better exploited in the future is *maturation.* The initial coupling and/or low resolution of sensors and actuators can give advantages by “starting small”. This concept has been started by [Elman, 1993] with the seminal paper “The Importance of Starting Small”.

The appearance and disappearance of reflex in babies seem to suggest that a *developmental plan* is in action. Some reflex facilitate behavior acquisition, while others substitute unknown behaviors that perform a given survival task. In our developmental architecture the transition from level to level was pre-defined and not autonomously supervised and selected. There is a strong need of mechanism that guide this development, e.g. performance ratios, user feedback, time, data availability, and others. The system would proceed to the next stage only when a given measure of performance is attained.

Appendix A

Baltazar Kinematics

A.1 Introduction

Research in imitation, skill transfer and visuomotor coordination have become increasingly important in the past few years, partly motivated by the advances in computing power and knowledge about vision or motor control in biological systems, and pushed by applications like service robotics or robot companions. However, such research efforts are often undermined by the unavailability of adequate, hand-eye-head robotic research platforms.

The goal of this section is to describe the design procedure (and final result) of a humanoid robotic torso combining an anthropomorphic arm, hand and binocular head (Figure A.1).

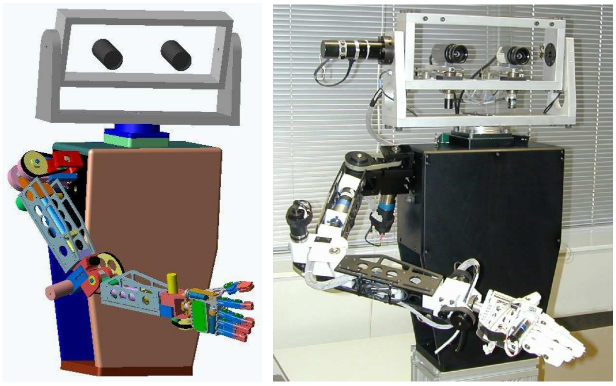

Figure A.1: Cad Model and real robot

Ever since the 60's, when the first robotic arms appeared, the development of new systems has never stopped. The *PUMA* is perhaps the most widely known robot arm, with 6 degrees of freedom and it is extremely robust. However, the control architecture is closed, the cost is relatively high for many research labs, the controllers (and power amplification) are quite large and only a simple gripper is available. There are other systems in the market, with lower cost and size. For example, the *KATANA* [Neuronics,] is a small manipulator, with an open architecture but it only has 5 degrees of freedom and a simple gripper.

With the increasing interest in humanoid robots, many different arm and hand designs have been suggested. The design varies quite significantly according to the desired use for these robots. Humanoid torsos, comprising arms and head, have been used to study head/arm coordination [Asfour et al., 1999, Brooks et al., 1998, Metta et al., 2001]. For legged robots there are several topics for research. These robots must generate the necessary torques to carry their own weight [Konno et al., 2000, Nagakubo et al., 2000]. Some of these robots may be able to fall down and stand-up by themselves. Other robots are designed to study human locomotion [Nishiwaki et al., 2000, Moeslund and Granum, 2003].

It is known that for a stable grasp we need at least 3 fingers, which is the reason why most robotic hands contain 3 fingers. The kinematics of *Shadow* [Rich Walker, 2003] resembles that of the human hand, with 21 degrees of freedom (DOFs). Other projects tried to achieve the same number of degrees of freedom, but not all of them can be controlled independently. The Karlsruhe[Fukaya et al., 2000] hand has 20 DOFs, actuated by a single motor. The rationale of the design is that the exact position of each finger is not important, when grasping objects in different ways. If the hand can automatically (passively) adapt to the object shape, then stable grasp can be achieved with minimum computation. Another robot hand with 13 degrees of freedom is described in [Butterfab et al., 2001].

Our design of a human-like robotic torso, consisting of a head, arm and hand, was driven by the following constraints:

- The robot kinematics should resemble that of the human torso. It should be able to perform human-like movements and gestures, as well as to allow a natural interaction with objects (e.g. while grasping).
- Payload of at least $500grs$ (with the hand).
- Force detection should be possible.
- Ease of maintenance and low-cost.

The constraint on the kinematics basically precludes the usage of commercially available systems. Most standard robot arms, like the PUMA, have the first 3 degrees of freedom (base azimuth, elevation and elbow elevation) to position the end effector in space, while the last 3 DOFs (in the wrist) allow the control of orientation. One important movement in the human arm is the rotation around the upper arm, that one cannot find in commercial systems. This movement enables the arm to better deal with obstacles and comfort, when manipulating (e.g. writing with the arm standing on a table compared to writing on a black board), and leads to more natural movements. Our robot arm is able to perform rotations around the upper arm. Although we have only two degrees of freedom in the wrist, combined with the DOFs available in the hand we get sufficient dexterity.

The Karlsruhe hand design has the attractive feature of allowing full adaptation to object's shape, while using just a single motor. We followed a similar concept to reduce the number of actuators, by keeping some degrees of freedom in the hand coupled. One important limitation with the Karlsruhe hand is that it cannot perform hand gestures, since the different fingers cannot be controlled independently. For closing some fingers independently, some DOFs must be decoupled.

Our hand has eleven degrees of freedom, that are controlled by four motors included in the hand. With this choice, our robot hand can adapt passively to the shape of objects but it can also perform a number of hand gestures that a simpler design would not allow. Thus, our design represents a trade off between simplicity and multi-purpose use. We show that several objects can be grasped: sphere, box, cylinders and general geometric shapes. With its four independent controllers it is also possible to perform the most significant hand gestures.

Since our anthropomorphic arm-hand will be in contact with objects in the world, it is necessary to be able to sense forces acting upon the system. We chose motor controllers with this capability. In every movement, the arm motor is limited to a maximum torque (current). For the hand, we also installed force sensors in the fingers pulp and palm to better control the exact contact forces.

The usage of standard components, whenever possible, has an important impact on the overall cost of actually producing the system, an important constraint for many research groups. We used regular *DC* motors with reduced backlash and off-the-shelf mechanical parts. The robot parts were machined in a workshop at our institute.

The arm can be assembled or disassembled easily. The overall design involved several iterations between design, analysis of specifications and prototyping. We are now obtaining the first results and we consider that it can be very attractive for other research labs as well.

A.2 Medusa Head

The robot was equipped with a binocular robotic head [Santos-Victor et al., 1994], previously developed in our lab. It has four degrees of freedom: neck rotation, head elevation and independent eye vergence (see the kinematics in Figure A.4). Manual adjustments can be made to align the vergence and elevation axes of rotation with the cameras optic centers. The inter-ocular distance can also be modified manually.

Table A.1 shows the Modified Denavitt-Hartenberg parameters for the Medusa head.

Let ${}^2P$ denote the $3D$ coordinates of point P, expressed in the eyes coordinates. If we denote by ${}^AP$, the coordinates of this point expressed in the arm base coordinate system, the following relation holds:

$$ {}^AP = {}^A_HT\,{}^0_1T_h\,{}^1_2T_h\,{}^2P $$

where the head-arm transformation, ${}^A_HT$, is given by:

$$ {}^A_HT = \begin{bmatrix} 0 & 0 & 1 & -27 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 29.6 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

This transformation allows to coordinate the head with the arm. Suppose that the arm is pointing toward the point ${}^0P = [x\,y\,z\,1]^T$. This point can be represented in eye coordinates as: ${}^2P = [ed\,0\,0\,1]^T$, where ed denotes the distance from the eye to the point. This equation can be used to determine the *pan* and *tilt* angles to allow the head to look toward the same point.

Figure A.2 shows several images where the head looks directly to the wrist, illustrating the head-arm coordination.

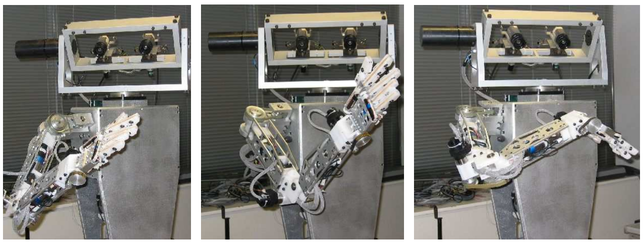

Figure A.2: Head/Hand Coordination example.

Difference to the right eye: $d_2 = -b/2$

b is the baseline of the eyes.

For the case of symmetric vergence $v_r = -\left|v\right|$ and $v_l = -\left|v\right|$

i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	0	pan
2	15	-90	0	$tilt$
3	0	90	$b/2$	0
4	0	90	0	v_r

Table A.1: Modified Denavitt-Hartenberg Parameters for the left eye.

A.3 Arm

A.3.1 The human arm anatomy

For the sake of completeness of the paper, we summarize here the main facts related to the human arm anatomy. It consists of a synthesis of what can be found in [Maurel, 1998].

The human arm is a very complex system. The upper limb is composed of three chained mechanisms, the shoulder girdle, the elbow and the wrist. Considering bones in pairs, seven joints may be distinguished. Except for the scapulo-thoracic joint, one can neglect the translational movement with respect to rotations, and assume that these joints behave as ball and socket joints, allowing 3 degrees of freedom (DOF) in rotation.

The seven joints can be identified as follows: the sterno-clavicular joint(3 DOF), which articulates the clavicle by its proximal end onto the sternum, the acromio-clavicular joint, which articulates the scapula by its acromion onto the distal end of the clavicle; the scapulo-thoracic joint(5 DOF), which allows the scapula to glide on the thorax; the gleno-humeral joint(3 DOF), which allows the humeral head to rotate in the glenoid fossa of the scapula; the ulno-humeral and the humero-radial joints; which articulate both ulna and radius on the distal end of the humerus, and finally the ulno-radial joint where both distal ends of ulna and radius join together(2 DOF).

To perform these movements, the upper limb is equipped with not less than 21 muscles actuators. Some muscles have very broad attachments, while some others divide in several bundles attached on different bones. These muscles can be classified in several groups according to the bone they move and the DOF they control. As muscles never work in isolation, natural movements always involve the motions of all bones. For a complete analysis, it is necessary to consider the motion of the mechanism as a whole.

A.3.2 Robot arm design

From the previous description, it is clear that it would be extremely difficult (if not impossible) to design a robot arm with the exact amazing capabilities of motion of the human arm. For constructing our robot arm, we were forced to introduce some simplifications, yet with the care of maintaining the main functional requirements. The shoulder was modeled with 3 DOF: external/internal rotation, abduction/adduction and extension/flexion. The elbow is endowed with 2 DOF: extension/flexion and supination/pronation. Finally, the wrist possesses 1 DOF: extension/flexion. Wrist abduction/adduction is not included but we will see that it is compensated by the hand.

The following sections describe the arm kinematics.

B.1. Direct Kinematics

In this section we present the computational model for the arm kinematics. The arm is described with the modified Denavitt-Hartenberg parameters [Craig, 1989], shown in Table A.2. The corresponding axes are illustrated in Figure A.4.

The first three joints model the shoulder. The first joint is responsible for abduction/adduction, the second for extension/flexion and the third for external/internal rotation. The fourth joint makes the extension/flexion of the elbow, the fifth the pronation/supination of the fore-arm and the sixth the extension/flexion of the wrist.

B.2. Inverse Kinematics

The inverse kinematics will be done in two parts: position of the wrist and orientation of the hand. Due the particular structure of the arm, an iterative process may be necessary. In most robot arms (e.g. the PUMA), the first 3 links determine the end-effector's position and the last three the orientation. In our arm the third joint contributes both to position and to orientation of the hand.

Let P denote the desired position of the wrist, and Z be the null vector. Writing these vectors in homogeneous coordinates, we have:

$$P = [x \;\; y \;\; z \;\; 1]^T \qquad Z = [0 \;\; 0 \;\; 0 \;\; 1]^T$$

The wrist position, P, can be related to the various joint angles by cascading the different homogeneous coordinate transformation matrices:

$$P = \prod_{i=0}^{5} {}^{i}_{i+1}T \; Z \tag{A.1}$$

where ${}^{i}_{i+1}T$ denotes the homogeneous coordinate transformation between frames $\{i+1\}$ and $\{i\}$. As, in general, most of the terms of this equation are transcendental, we will use the fact that the equation:

$$a\cos(\theta) + b\sin(\theta) = c, \quad \text{has solutions}$$

$$\theta = 2\arctan\left(\frac{b \pm \sqrt{a^2 + b^2 - c^2}}{a + c}\right) \tag{A.2}$$

This equation will be useful when determining the joint angles in the inverse kinematics. Notice that the equation provides two solutions. The desired joint position must be chosen according to the physical limits of the joint and/or using additional criteria (e.g. comfort, least change).

i) Positioning the wrist

To move the arm wrist to a given position, P, in space, we need to determine the corresponding values of θ_1, θ_2, θ_3 and θ_4. Given the kinematics of our anthropomorphic arm, the distance, ρ, from the base to the wrist depends only on θ_4. Using Equation (A.1), the following constraint holds:

$$a\cos(\theta_4) + b\sin(\theta_4) = \rho^2 - (a_2^2 + l_2^2 + l_1^2 + a_1^2)$$

where we have used:

$$\begin{cases} a &= 2(-a_2a_1 + l_2l_1) \\ b &= -2(l_2a_1 + a_2l_1) \end{cases}$$

The value of θ_4 is readily obtained applying Equation (A.2). To determine θ_2, we will use the expression related to the z component, in Equation (A.1). This equation provides a solution for θ_2, as well as a constraint on θ_3 to ensure that a solution to Equation (A.2) exists. Hence, we first define an initial value for θ_3 and solve for θ_2. Then, we can change the value of θ_3, within the prescribed limits, and re-calculate θ_2.

Having calculated θ_4, θ_2 and θ_3, we now have to determine θ_1. We first move all the terms in Equation (A.1) that depend on θ_1 and θ_2 to the left hand side, yielding:

$${}^{2}_{1}T\,{}^{1}_{0}T\,P = {}^{2}_{3}T\,{}^{3}_{4}T\,{}^{4}_{5}T\,{}^{5}_{6}T\,Z \tag{A.3}$$

From Equation (A.3), we obtain two transcendental equations on θ_1, each providing two possible solutions for θ_1. The final value for θ_1 must be a solution to both equations.

In the beginning, θ_3 can be chosen freely, within the restriction for evaluating θ_2. However, for some choices for θ_3, it may be impossible to solve for θ_1 or, more commonly, the solution obtained may be outside the physical limits of the robot. When this occurs, a new value must be chosen for θ_3 and the whole process for solving for θ_2 and θ_1 repeated, until a solution is found. This problem occurs when we want to reach positions just in in front of the body. In such a case, we can use $\theta_3 = 90\frac{z}{l_2}$, as an initial condition.

Given the kinematic structure of of the anthropomorphic robot arm, we need to use four different joints to reach the desired position of the wrist. In some sense, this gives us some redundancy to overcome obstacles or to find comfortable positions for the arm. However, we are left with only two degrees of freedom to orient the hand.

ii) Reaching an orientation

In conventional manipulators, the problems of positioning and orienting the end-effector are decoupled: the first 3 DOFs allow to position the end-effector and the remaining 3 are used to establish the orientation. The use of an anthropomorphic arm, in combination with a hand, allows us to proceed differently. When grasping a tool, attaching objects, or in many other tasks, the human hand may be constrained to work on a given plane. It is seldom the case that a specific orientation is needed. The reason is that the hand itself provides the extra mobility that might be necessary.

In our work, we propose to use a process of inverse kinematics that will fix the orientation of the hand parallel to a given working plane, allowing the hand to rotate freely around an axis perpendicular to this plane. For this approach, we only need two DOFs, the angles θ_5 and θ_6.

Let $V_\pi = [v_1 v_2 v_3]^T$ be normal to the working plane, Π. Our problem is then to determine the angles θ_5 and θ_6, such that the hand becomes parallel to Π.

Let ${}^0_6R$ represent the orientation of the coordinate frame $\{6\}$ with respect to the arm basis, $\{0\}$. The columns of ${}^0_6R$ are the axes of the frame $\{6\}$, expressed in the arm basis frame. From Figure A.4, we can observe that the $x-$axis of the frame $\{6\}$ is perpendicular to the hand palm. Hence, the problem of keeping the hand palm parallel to the plane Π can be re-stated as determining θ_5 and θ_6, to make the first column of ${}^0_6R$ equal to V_π.

$${}^0_6R \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

Since ${}^0_4R$ is already known, θ_6 and θ_5 can be determined from the equation above, completing the process of determining the inverse kinematics.

If, in addition to aligning the hand palm with the working plane, one wants to reach a specific orientation around the normal to this plane, the value for θ_3 must be chosen accordingly. As this implies searching in one parameter only, which has a small amplitude, this computation is very fast. In this case, θ_3 can no longer be used as a redundant DOF when driving the wrist to some specified position.

Figure A.3 shows a solution for a vertical working plane (parallel to the robot torso). The different orientations for the hand are obtained by choosing different values for internal/external shoulder rotation (θ_3).

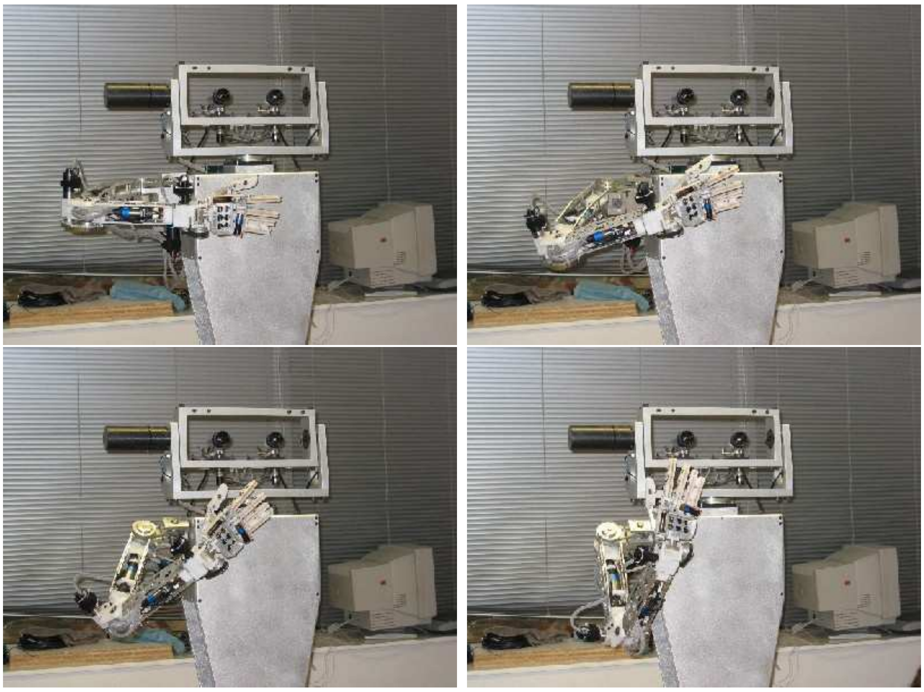

Figure A.3: Sequence of movements with the hand moving over a working plane, and changing the orientations, due to the change of θ_3

Joint	a_{i-1}	α_{i-1}	d_i	$\theta_i(deg)$	$limits(deg)$
1	0	0	0	θ_1	[-45 135]
2	0	90	0	$\theta_2 + 90$	[-110 10]
3	0	90	l_1	$\theta_3 + 90$	[-90 0]
4	a_1	90	0	θ_4	[-90 0]]
5	$-a_2$	-90	l_2	$\theta_5 + 90$	[-90 90]]
6	0	90	0	θ_6	[-45 45]

Table A.2: Modified Denavitt-Hartenberg parameters of Baltazar's arm. All angular variables are expressed in degrees.

A.3.3 Design of an anthropomorphic robot arm

As we mentioned before, our goal is to design a robot arm for conducting research in human-based imitation, learning and visuomotor coordination. For this reason, the arm kinematics must resemble that of humans. In this section, we present an overview of the kinematics of the human arm, followed by the description of our design options and results. Finally we describe the direct and inverse kinematics of

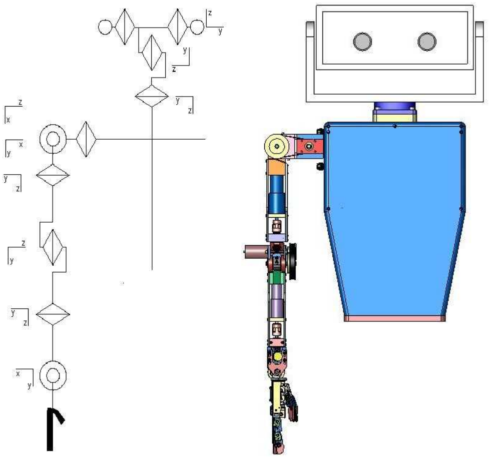

Figure A.4: Kinematic structure for the arm and head.

our anthropomorphic robot arm. When presenting the inverse kinematics, we will demonstrate some interesting properties of our design.

A.4 Hand

A.4.1 Design of an anthropomorphic robot hand

Similarly to what we did with the arm, we start by providing a concise description of the human hand's anatomy, synthesized from [Valero-Cuevas and Hentz, 1997], followed by the description of our robot hand.

The human hand anatomy

The hand is the organ of the human body that is most well adapted to prehensile function. The hand is composed of the palm and digits and is articulated to the forearm by the wrist (carpus). The palm is a flat surface that serves as the central support surface to the hand. The digits are composed of long bones called phalanges arranged in series continuing each metacarpal ray. The first digit, called the thumb, is composed of two phalanges; it is the most mobile of the digits and can oppose to the palm and the tips of the other fingers when these are flexed. The remaining four digits each contain 3 phalanges. Axial rotation (pronation-supination) of the hand occurs as the more mobile the two long bones of the forearm (the radius) rotates about the other relatively fixed bone (the ulna). The length, distribution and mobility of the digits with respect to the palm give the hand the ability to perform a wide variety of prehensile tasks.

The articulation connecting the digits to the metacarpals (MCP joint) allows for motion that is mostly independent of each other in flexion-extension and abduction-adduction (side-to-side motion). The thumb is different from the fingers in that it contains only 2 phalanges and its metacarpal bone has a wide range of motion where its base articulates with the carpus (i.e., thumb carpo-metacarpal, or CMC, joint). This makes the thumb the most independent and mobile of the digits. Its architectural and kinematic complexity separates the hand of man from other primates. The thumb occupies a special place in the digital pantheon.

The kinematics of the fingers have been approximated by rigid segments connected by ideal pin joints permitting ad-abduction and flexion-extension at the MCP joint, and flexion-extension at the proximal inter-phalangeal (PIP) and distal inter-phalangeal (DIP) joints.

The kinematics of the thumb are still not well understood. The large range of motion and mobility of the thumb has led to at least six different models in the literature.

The human hand nominally has 40 muscles classified as those located in the hand distal to the wrist (intrinsic muscles), and those located in the forearm (extrinsic muscles). Some tendons of the hand are atypical as they bifurcate or combine before inserting into bone to form the extensor mechanism (or extensor hood) of the fingers. The lumbrical muscle is atypical as it both originates from and inserts onto tendon (the flexor profundus tendon and the extensor hood, respectively) and has no direct bony attachment. Mammalian muscle tissue is considered to produce a maximal stress of $35N/cm^2$, which is a remarkable force/weight ratio difficult to match artificially.

Robot hand design

Our goal is to design an artificial hand capable of grasping objects and making gestures. Given the complexity of the human hand, we were forced to simplify the kinematics. In our robot hand, the index finger has three DOFs. The thumb has two DOFs plus one rotation and the other fingers have two DOFs. The pinkie and anelar fingers are mechanically coupled. The abductions were not implemented. The hand is shown in Figure A.5.

One design constraint was to include all the motors on the hand. For this reason, we could not afford the space to use one independent motor for each joint in the fingers. The index finger is controlled by a single motor that pulls a tendon to close the finger. The thumb has two motors: one for rotation and another for closing the thumb by means of a tendon. The other three fingers have one motor/tendon mechanism that closes them all together.

At a first glance, the fact that not all the DOFs of each finger are controlled independently may seem very limiting. However, the finger joints are also strongly coupled in the human hand. In fact, our design gives some compliance to the robot

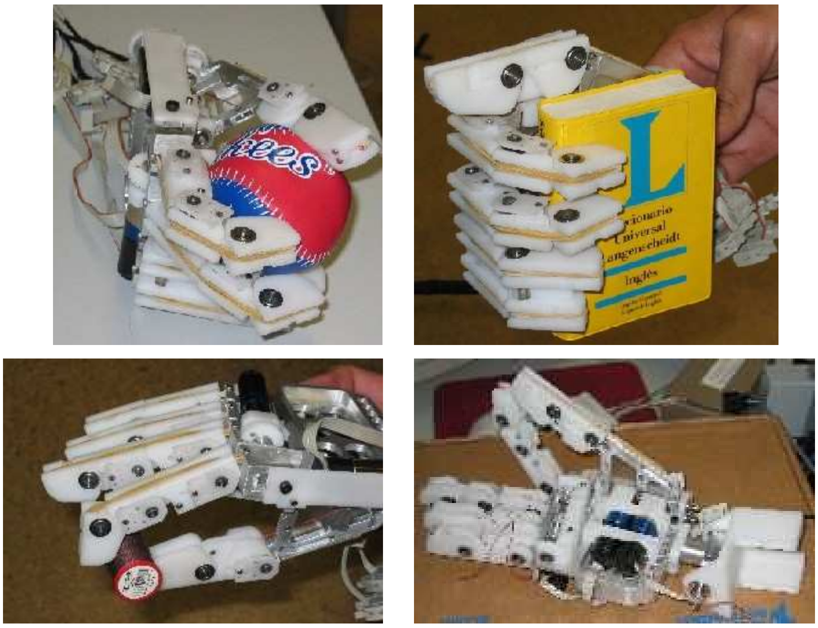

Figure A.5: Hand detail. Examples of grasping different objects.

hand, and makes the grasp control much easier, as the fingers adapt automatically to the shape objects. Without obstacles, the fingers start closing with the MCP joint. Figure A.5 shows the hand grasping different types of objects.

The use of a single motor and a tendon to close a finger provides the hand with the ability to passively adapt to the objects shape. However, we need more proprioceptive information (in addition to motor shaft position) to know the state of the hand. The thumb, little and anelar fingers have one potentiometer as a position sensor, the middle and index have two potentiometers. In order to interact with objects, we installed pressure sensors in several places of the hand to measure contact forces.

Bibliography

[Alissandrakis et al., 2005] Alissandrakis, A., Nehaniv, C. L., Dautenhahn, K., and Saunders, J. (2005). Achieving corresponding effects on multiple robotic platforms: Imitating in context using different effect metrics. In *Third International Symposium on Imitation in Animals & Artifacts*, pages 10–19. SSAISB.

[Asada et al., 2001] Asada, M., MacDorman, K., Ishiguro, H., and Kuniyoshi, Y. (2001). Cognitive developmental robotics as a new paradigm for the design of humanoid robots. *Robotics and Automation*, 37:185–193.

[Asada et al., 2000] Asada, M., Yoshikawa, Y., and Hosoda, K. (2000). Learning by observation without three-dimensional reconstruction. In *Intelligent Autonomous Systems (IAS-6)*.

[Asfour et al., 1999] Asfour, T., Berns, K., Schelling, J., and Dillman, R. (1999). Programming of manipulation tasks of the humanoid robot armar. In *International Conference on Advanced Robotics*, pages 25–27, Japan.

[Atkeson et al., 1997] Atkeson, C. G., Moore, A. W., and Schaal, S. (1997). Locally weighted learning. *Artificial Intelligence Review*, 11(1-5):11–73.

[Baerlocher and Boulic, 2004] Baerlocher, P. and Boulic, R. (2004). An inverse kinematic architecture enforcing an arbitrary number of strict priority levels. *The Visual Computer*, 6(20).

[Baggenstoss, 2002] Baggenstoss, P. M. (2002). Statistical modeling using gaussian mixtures and hmms with matlab. http://www.npt.nuwc.navy.mil/Csf/htmldoc/pdf/.

[Banks, 1980] Banks, M. S. (1980). The development of visual accomodation during early infancy. *Child Development*, 51:646–666.

[Bekkering et al., 2000] Bekkering, H., Wohlschläger, A., and Gattis, M. (2000). Imitation of gestures in children is goal-directed. *Quarterly Journal of Experimental Psychology*, 53A:153–164.

[Bernardino and Santos-Victor, 2002] Bernardino, A. and Santos-Victor, J. (2002). A binocular stereo algorithm for log-polar foveated systems. In *Biological Motivated Computer Vision*, Tuebingen, Germany.

[Bertero et al., 1988] Bertero, M., Poggio, T., and Torre, V. (1988). Ill-posed problems in early vision. *Proceedings of the IEEE*, 76(8):869–889.

[Billard, 2003] Billard, A. (2003). Robota: Clever toy and educational tool. *Robotics and Autonomous Systems*, 42:259–269.

[Billard et al., 2004] Billard, A., Epars, Y., Calinon, S., Cheng, G., and Schaal, S. (2004). Discovering optimal imitation strategies. *Robotics and Autonomous Systems*, 47:2-3.

[Billard and Hayes, 1997] Billard, A. and Hayes, G. (1997). Transmitting communication skills through imitation in autonomous robots. In *Sixth European Workshop on Learning Robots*, Brighton.

[Billard and Hayes, 1999] Billard, A. and Hayes, G. (1999). Drama, a connectionist architecture for control and learning in autonomous robots. *Adaptive Behaviour*, 7(1).

[Birch et al., 1998] Birch, E., Hoffman, D., Uauy, R., Birch, D., and Prestidge, C. (1998). Visual acuity and the essentiality of docosahexaenoic acid and arachidonic acid in the diet of term infants. *Pediatric Research*, 44(2).

[Birch et al., 2005] Birch, E., Morale, S., Jeffrey, B., OŠConnor, A., and Fawcett, S. (2005). Measurement of stereoacuity outcomes at ages 1 to 24 months: Randotő stereocards. *Journal of American Association for Pediatric Ophthalmology and Strabismus*, 9(1).

[Black and Jepson, 1996] Black, M. J. and Jepson, A. D. (1996). Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In *ECCV (1)*, pages 329–342.

[Blackburn and Nguyen, 1994] Blackburn, M. and Nguyen, H. (1994). Learning in robot vision directed reaching: A comparison of methods. In *ARPA Image Understanding Workshop*, Moterey, CA.

[Bobick and Davis, 1996] Bobick, A. and Davis, J. (1996). Real-time recognition of activity using temporal templates. In *3rd IEEE Workshop on Applications of Computer Vision*, Sarasoto, USA.

[Bower, 1977] Bower, T. G. R. (1977). A primer of infant development. In *Freeman*, San Franscisco.

[Breazeal, 1998] Breazeal, C. (1998). A motivation system for regulating human-robot interaction. In *Proceedings of the fifteenth National Conference on Artificial Intellilgence (AAAI 98)*, pages 54–61, Madison, USA.

[Breazeal, 1999] Breazeal, C. (1999). Imitation as social exchange between humans and robots. In *AISB Symposium on Imitation in Animals and Artifacts*, pages 96–104, Edinburgh.

[Breazeal and Scassellati, 2000] Breazeal, C. and Scassellati, B. (2000). Challenges in building robots that imitate people. In Dautenhahn, K. and Nehaniv, C., editors, *Imitation in Animals and Artifacts.* MIT Press.

[Brooks et al., 1998] Brooks, R., Breazeal, C., Marjanovic, M., Scassellati, B., and Williamson, M. (1998). The cog project: Building a humanoid robot. In Nehaniv, C., editor, *Computation for Metaphors, Analogy and Agents*, volume 1562, pages 53–88. Springer-Verlag.

[Bruner, 1972] Bruner, J. (1972). Nature and use of immaturity. *American Psychologist*, 27:687–708.

[Buss, 2004] Buss, S. R. (2004). Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. Technical report, University of California, San Diego, USA.

[Butterfab et al., 2001] Butterfab, J., Grebenstein, M., Liu, H., and Hirzinger, G. (2001). Dlr-hand ii: Next generation of a dextrous robot hand. In *International Conference on Robotics and Automation*, pages 109–114, Seoul, Korea.

[Byrne, 1995] Byrne, R. (1995). *The Thinking Ape Evolutionary Origins of Intelligence.* Oxford University Press.

[Byrne, 1999] Byrne, R. W. (1999). Imitation without intentionality. using string parsing to copy the organization of behaviour. *Animal Cognition*, 2:63–72.

[Byrne, 2002] Byrne, R. W. (2002). Imitation of novel complex actions: What does the evidence from animals mean? *Advances in the Study of Bahaviour*, 31:77–105.

[Chaumette, 1998] Chaumette, F. (1998). Potential problems of stability and convergence in image-based and position-based visual servoing. In Kriegman, D., Hager, G. ., and Morse, A., editors, *The Confluence of Vision and Control*, pages 66–78. LNCIS Series, No 237, Springer-Verlag.

[Chaumette, 2004] Chaumette, F. (2004). Image moments: a general and useful set of features for visual servoing. *IEEE Trans. on Robotics*, 20(4):713–723.

[Craig, 1989] Craig, J. J. (1989). *Introduction to Robotics.* Addison-Wesley Pub Co.

[CyberGlove,] CyberGlove. http://www.immersion.com.

[de Mathelin and Lozano, 1999] de Mathelin, M. and Lozano, R. (1999). Robust adaptive identification of slowly time-varying parameters with bounded disturbances. *Automatica*, 35:1291–1305.

[Demiris and Hayes, 1996] Demiris, J. and Hayes, G. (1996). Imitative learning mechanisms in robots and humans. In *European Workshop on Learning Robots*, pages 9–16, Bari, Italy.

[Demiris et al., 1997] Demiris, J., Rougeaux, S., Hayes, G. M., Berthouze, L., and Kuniyoshi, Y. (1997). Deferred imitation of human head movements by an active stereo vision head. In *6th IEEE International Workshop on Robot Human Communication*, pages 88–93, Sendai, Japan.

[Demiris and Khadhouri, 2006] Demiris, Y. and Khadhouri, B. (2006). Hierarchical attentive multiple models for execution and recognition of actions. *Robotics and Autonomous Systems*, to appear.

[Dodds et al., 1999] Dodds, Z., Jägersand, M., and amd K. Toyama, G. H. (1999). A hierarchical vision architecture for robotic manipulation tasks. In *Proc. of Int. conf. on Computer Vision Systems.*

[D'Souza et al., 2001] D'Souza, A., Vijayakumar, S., and Schaal, S. (2001). Learning inverse kinematics. In *International Conference on Intelligent Robots and Systems*, Hawaii, USA.

[Elman, 1993] Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–9.

[Espiau et al., 1992] Espiau, B., Chaumette, F., and Rives, P. (1992). A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326.

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

[Faugeras, 1993] Faugeras, O. (1993). *Three-dimensional computer vision: a geometric viewpoint.* MIT Press, Cambridge, Massachusetts.

[Fitzpatrick et al., 2003] Fitzpatrick, P., Metta, G., Natale, L., Rao, S., and Sandini., G. (2003). Learning about objects through action: Initial steps towards artificial cognition. In *IEEE International Conference on Robotics and Automation*, Taipei, Taiwan.

[Fletcher, 1987] Fletcher, R. (1987). *Practical Methods of Optimization.* Chichester, 2nd edition.

[Fogassi et al., 2001] Fogassi, L., Gallese, V., Buccino, G., Craighero, L., Fadiga, L., and Rizzolatti, G. (2001). Cortical mechanism for the visual guidance of hand grasping movements in the monkey: A reversible inactivation study. *Brain*, 124(3):571–586.

[Fukaya et al., 2000] Fukaya, N., Toyama, S., Asfour, T., and Dillman, R. (2000). Design of the tuat/karlsruhe humanoid hand. In *International Robots and Systems*, Japan.

[Furse, 2001a] Furse, E. (2001a). Imitation: a solution to end-user programming. In *First International Conference on End User Programming*, Sienna, Italy.

[Furse, 2001b] Furse, E. (2001b). A model of imitation learning of algorithms from worked examples. *Cybernetics and Systems*, 32:121–154.

[Gardner, 2002] Gardner, M. (2002). Imitation and egocentric perspective transformation. In *Virtual Poster associated with Perspectives on Imitation Conference*, Royaumont Abbey, France.

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

[Gavrila, 1999] Gavrila, D. M. (1999). The visual analysis of human movement: A survey. *CVIU*, 73(1):82–98.

[Gergely et al., 2002] Gergely, G., Bekkering, H., and Király, I. (2002). Rational imitation in preverbal infants. *Nature*, 415:755.

[Gibson, 1979] Gibson, J. J. (1979). *The Ecological Approach to Visual Perception.* Houghton Mifflin, Boston.

[Hager, 2002] Hager, G. (2002). Human-machine cooperative manipulation with vision-based motion constraints. *Workshop on visual servoing, iros02.*

[Hastie and Loader, 1993] Hastie, T. and Loader, C. (1993). Local regression: Automatic kernel carpentry. *Statistical Science*, 8:120–129.

[Hastie et al., 2001] Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning.* Springer.

[Hayes and Demiris, 1994] Hayes, G. and Demiris, J. (1994). A robot controller using learning by imitation. In *International Symposium on Intelligent Robotic Systems*, pages 198–204, Grenoble, France.

[Hosoda and Asada, 1994] Hosoda, K. and Asada, M. (1994). Versatile visual servoing without knowledge of true jacobian. In *International Conference on Intelligent Robots and Systems*, pages 186–193, Munchen, Germany.

[Hosoda and Asada, 2000] Hosoda, K. and Asada, M. (2000). Advances in robot learining. In Wyatt, J. and Demiris, J., editors, *How Does a Robot Find Redundancy by Itself? – A Control Architecuture for Adaptive Multi DOF Robots.* Springer.

[Hotz et al., 2003] Hotz, P. E., Gómez, G., and Pfeifer, R. (2003). Evolving the morphology of a neural network for controlling a foveating retina and its test on a real robot. In *Artificial Life VIII - 8th International Conference on the Simulation and Synthesis of Living Systems*, volume 2003.

[Hovland et al., 1996] Hovland, G., Sikka, P., and McCarragher, B. (1996). Skill acquisition from human demonstration using a hidden markov model. In *IEEE International Conference on Robotics and Automation*, pages 2706–2711, Minneapolis, MN.

[Hutchinson et al., 1996] Hutchinson, S., Hager, G., and Corke, P. (1996). A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670.

[Jagersand and Nelson, 1996] Jagersand, M. and Nelson, R. (1996). On-line estimation of visual-motor models using active vision. In *ARPA Image Understanding Workshop*, pages 677–682.

[Khatib et al., 2004] Khatib, O., Brock, O., Chang, K.-S., Ruspini, D., Sentis, L., and Viji, S. (2004). Human-centered robotics and interactive haptic simulation. *International Journal of Robotics Research*, 23(2).

[Konno et al., 2000] Konno, A., Kato, N., Shirata, S., Furuta, T., and Uchiyama, M. (2000). Development of a light-weight biped humanoid robot. In *International Robots and Systems*, Japan.

[Kozima, 2000] Kozima, H. (2000). Infanoid: An experimental tool for developmental psycho-robotics. In *International Workshop on Developmental Study*, Tokyo, Japan.

[Kozima et al., 2002] Kozima, H., Nakagawa, C., and Yano, H. (2002). Emergence of imitation mediated by objects. In *Second International Workshop on Epigenetic Robotics*, Edinburgh, Scotland.

[Kozima et al., 2003] Kozima, H., Nakagawa, C., and Yano, H. (2003). Attention coupling as a prerequisite for social interaction. In *IEEE International Workshop on Robot and Human Interactive Communication*, Millbrae, USA.

[Kozima and Yano, 2001] Kozima, H. and Yano, H. (2001). Designing a robot for contingency-detection game. In *International Workshop on Robotic and Virtual Agents in Autism Therapy*, Hertfordshire, England.

[Kragic et al., 2002] Kragic, D., Petersson, L., and Christensen, H. I. (2002). Visually guided manipulation tasks. *Robotics and Autonomous Systems*, 40(2-3):193–203.

[Kuniyoshi et al., 1994] Kuniyoshi, Y., Inaba, M., and Inoue, H. (1994). Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *Trans. on Robotics and Automation*, 10(6):799–822.

[Kuniyoshi et al., 2003] Kuniyoshi, Y., Yorozu, Y., Inaba, M., and Inoue, H. (2003). From visuo-motor self learning to early imitation-a neural architecture for humanoid learning. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 3132–3139.

[Lapresté et al., 2004] Lapresté, J. T., Jurie, F., Dhome, M., and Chaumette, F. (2004). An efficient method to compute the inverse jacobian matrix in visual servoing. In *International Conference in Robotics and Automation.*

[Lieberman, 1993] Lieberman, H. (1993). Tinker: A programming by demonstration system for beginning programmers. In Cypher, A., editor, *Watch What I Do: Programming by Demonstration.* MIT Press.

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

[Lopes et al., 2004] Lopes, M., Beira, R., Praça, M., and Santos-Victor, J. (2004). An anthropomorphic robot torso for imitation: design and experiments. In *International Conference on Intelligent Robots and Systems*, Sendai, Japan.

[Lopes et al., 2005] Lopes, M., Bernardino, A., and Santos-Victor, J. (2005). A developmental roadmap for task learning by imitation in humanoid robots. In Demiris, Y., editor, *AISB - Third International Symposium on Imitation in Animals and Artifacts*, Hatfield, Uk.

[Lopes and Santos-Victor, 2003] Lopes, M. and Santos-Victor, J. (2003). Visual transformations in gesture imitation: What you see is what you do. In *IEEE - International Conference on Robotics and Automation*, Taipei, Taiwan.

[Lopes and Santos-Victor, 2005] Lopes, M. and Santos-Victor, J. (2005). Visual learning by imitation with motor representations. *IEEE - Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 35(3).

[Lungarella et al., 2003] Lungarella, M., Metta, G., Pfeifer, R., and Sandini, G. (2003). Developmental robotics: a survey. *Connection Science*, 15(40):151–190.

[Maistros and Hayes, 2001] Maistros, G. and Hayes, G. (2001). An imitation mechanism for goal-directed actions. In *Towards Intelligent Mobile Robots*, Manchester, England.

[Malis, 2004] Malis, E. (2004). Improving vision-based control using efficient second-order minimization techniques. In *IEEE Int. Conf. on Robotics and Automation (ICRA'04)*, New Orleans, USA.

[Mansard and Chaumette, 2004] Mansard, N. and Chaumette, F. (2004). Tasks sequencing for visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'04)*, Sendai, Japan.

[Mansard and Chaumette, 2005] Mansard, N. and Chaumette, F. (2005). Visual servoing sequencing able to avoid obstacles. In *IEEE Int. Conf. on Robotics and Automation, ICRA'05*, Barcelona, Spain.

[Marchand et al., 2001] Marchand, E., Chaumette, F., Spindler, F., and Perrier, M. (2001). Controlling an uninstrumented rov manipulator by visual servoing. In *MTS/IEEE OCEANS 2001 Conference*, volume 2, pages 1047–1053, Honolulu, Hawaii.

[Marchand and Hager, 1998] Marchand, E. and Hager, G. (1998). Dynamic sensor planning in visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'98)*, volume 3, pages 1988–1993, Leuven, Belgium.

[Marjanović et al., 1996] Marjanović, M., Scassellati, B., and Williamson, M. (1996). Self-taught visually guided pointing for a humanoid robot. In *Fourth International Conference on Simulation of Adaptive Behavior*, MA.

[Matarict́, 2002] Matarict́, M. J. (2002). Sensory-motor primitives as a basis for learning by imitation:linking perception to action and biology to robotics. In Dautenhahn, K. and Nehaniv, C., editors, *Imitation in Animals and Artifacts*. MIT Press.

[Maurel, 1998] Maurel, W. (1998). *3D Modeling of the Human Upper Limb including the Biomechanics of Joints, Muscles and Soft Tissues*. PhD thesis, Laboratoire d'Infographie - Ecole Polytechnique Federale de Lausanne.

[Meltzoff, 1988] Meltzoff, A. N. (1988). Infant imitation after a 1-week delay: Long-term memory for novel acts and multiple stimuli. *Developmental Psychology*, 24(4):470–476.

[Meltzoff and Prinz, 2002] Meltzoff, A. N. and Prinz, W., editors (2002). *The Imitative Mind.* Cambridge University Press.

[Merriam-Webster, 2005] Merriam-Webster (2005). Merriam-Webster online dictionary. www.merriam-webster.com.

[Metta, 1999] Metta, G. (1999). *Babybot: A Study on Sensori-motor Development.* PhD thesis, University of Genova.

[Metta et al., 2000a] Metta, G., Manzotti, R., Panerai, F., and Sandini, G. (2000a). Babybot: an artificial developing robotic agent. In *From Animals to Animats: Sixth International Conference on the Simulation of Adaptive Behavior*, Paris, France.

[Metta et al., 2000b] Metta, G., Manzotti, R., Panerai, F., and Sandini, G. (2000b). Development: Is it the right way towards humanoid robotics? In *IAS*, Venice, Italy.

[Metta et al., 2001] Metta, G., Manzotti, R., Panerai, F., and Sandini, G. (2001). Babybot: an artificial developing robotic agent. In *SAB*, Paris, France.

[Metta et al., 1999] Metta, G., Sandini, G., and Konczak, J. (1999). A developmental approach to visually-guided reaching in artificial systems. *Neural Networks*, 2:1413–1427.

[Mishra and Silver, 1989] Mishra, B. and Silver, N. (1989). Some discussion of static gripping and its stability. *IEEE Trans. Syst. Man. Cybern.*, 19:783–796.

[Moeslund and Granum, 2003] Moeslund, T. B. and Granum, E. (2003). Modelling and estimating the pose of a human arm. *Machine Vision and Applications*, 14:237–247.

[Murata et al., 1997] Murata, A., Fadiga, L., Fogassi, L., Gallese, V., Raos, V., and Rizzolatti, G. (1997). Object representation in the ventral premotor cortex (area f5) of the monkey. *Journal of Neurophysiology*, 78(4):2226–2230.

[Nagai et al., 2002] Nagai, Y., Asada, M., and Hosoda, K. (2002). A developmental approach accelerates learning of joint attention. In *International Conference on Development and Learning.*

[Nagakubo et al., 2000] Nagakubo, A., Kuniyoshi, Y., and Cheng, G. (2000). Development of a high-performance upper-body humanoid system. In *International Robots and Systems*, Japan.

[Nakaoka et al., 2003] Nakaoka, S., Nakazawa, A., Yokoi, K., Hirukawa, H., and Ikeuchi, K. (2003). Generating whole body motions for a biped humanoid robot from captured human dances,. In *ICRA*, Taipei, Taiwan.

[Natale, 2004] Natale, L. (2004). *Linking Action to Perception in a Humanoid robot: a Developmental Approach to Grasping.* PhD thesis, University of Genova.

[Natale et al., 2004] Natale, L., Metta, G., and Sandini, G. (2004). Learning haptic representation of objects. In *International Conference on Intelligent Manipulation and Grasping*, Genoa, Italy.

[Nehaniv and Dautenhahn, 1998] Nehaniv, C. and Dautenhahn, K. (1998). Mapping between dissimilar bodies: Affordances and the algebraic foundations of imitation. In *European Workshop on Learning Robots*, Edinburgh, Scotland.

[Nehaniv and Dautenhahn, 2001] Nehaniv, C. L. and Dautenhahn, K. (2001). Like me? - measures of correspondence and imitation. *Cybernetics and Systems*, 32:11–51.

[Nehaniv and Dautenhahn, 2002] Nehaniv, C. L. and Dautenhahn, K. (2002). *Imitation in Animals and Artifacts*, chapter 2 - The Correspondence Problem. MIT Press.

[Neuronics,] Neuronics. Katana. Technical report, http://www.neuronics.ch/.

[Nishiwaki et al., 2000] Nishiwaki, K., Sugihara, T., Kagami, S., Kanehiro, F., Inaba, M., and H.Inoue (2000). Design and development of research platform for perception-action integration in humanoid robot: H6. In *International Robots and Systems*, Japan.

[Oztop, 2002] Oztop, E. (2002). *Modeling the Mirror: Grasp Learning and Action Recognition.* PhD thesis, University of Southern California.

[Payne and Isaacs, 1999] Payne, V. G. and Isaacs, L. D. (1999). *Human Motor Development: a Lifespan Approach.* Mayfield Publishing Company, California, USA, 4th edition.

[Perret et al., 1989] Perret, D. I., Harries, M. H., Bevan, R., Thomas, S., Benson, P. J., A. J. Mistlin, A. J. C., Hietanen, J. K., and Ortega, J. E. (1989). Frameworks of analysis for the neural representations of animate objects and actions. *Journal of Experimental Biology*, 146:87–113.

[Piepmeier et al., 2002] Piepmeier, J. A., Gumpert, B. A., and Lipkin, H. (2002). Uncalibrated eye-in-hand visual servoing. In *ICRA*.

[Piepmeier et al., 1999] Piepmeier, J. A., McMurray, G. V., and Lipkin, H. (1999). A dynamic quasi-newton method for uncalibrated visual servoing. In *ICRA*.

[Poggio et al., 1985] Poggio, T., Torre, V., and Kock, C. (1985). Computational vision and regularization theory. *Nature*, 317:314–319.

[Pomplun and Matarict́, 2000] Pomplun, M. and Matarict́, M. J. (2000). Evaluation metrics and results of human arm movement imitation. In *IEEE-RAS International Conference on Humanoid Robotics*, Cambridge, MA, USA.

[Price, 2003] Price, B. (2003). *Accelerating Reinforcement Learning with Imitation.* PhD thesis, University of British Columbia.

[Ramachandran, 2000] Ramachandran, V. (2000). Mirror neurons and imitation learning as the driving force behind the great leap forward in human evolution. *Edge*, 69.

[Rehg and Kanade, 1995] Rehg, J. M. and Kanade, T. (1995). Model-based tracking of self-occluding articulated objects. In *ICCV*, pages 612–617.

[Rich Walker, 2003] Rich Walker, S. R. C. (2003). Design of a dextrous hand for advanced clawar applications. In *6th International Conference on Climbing and Walking Robots*, Catania, Italy.

[Rizzolatti et al., 1977] Rizzolatti, G., Fadiga, L., Fogassi, L., and Gallese, V. (1977). The space around us. *Science*, 277:190–191.

[Robins et al., 2004] Robins, B., Dautenhahn, K., te Boekhorst, R., and Billard, A. (2004). Effects of repeated exposure to a humanoid robot on children with autism. In Keates, S., Clarkson, J., Langdon, P., and Robinson, P., editors, *Designing a More Inclusive World.* Springer Verlag, London.

[Rochat, 2002] Rochat, P. (2002). Ego function of early imitation. In Meltzoff, A. N. and Prinz, W., editors, *The Imitative Mind.* Cambridge University Press.

[Rochat et al., 1999] Rochat, P., Goubet, N., and Senders, S. J. (1999). To reach or not to reach? perception of body effectivities by young infants. *Infant and Child Development*, 8(3):129–148.

[Rosen, 1960] Rosen, J. (1960). The gradient projection method for nonlinear programmimg, part i, linear constraints. *SIAM Journal of Applied Mathematics*, 8:181–217.

[Rougeaux and Kuniyoshi, 1998] Rougeaux, S. and Kuniyoshi, Y. (1998). Robust tracking by a humanoid vision system. In *International Workshop on Humanoid and Human Friendly Robotics*, Tsukuba, Japan.

[Samson et al., 1991] Samson, C., Le Borgne, M., and Espiau, B. (1991). *Robot Control: the Task Function Approach.* Clarendon Press, Oxford, United Kingdom.

[Santos-Victor et al., 1994] Santos-Victor, J., van Trigt, F., and Sentieiro, J. (1994). Medusa - a stereo head for active vision. In *International Workshop on Intelligent Robotic Systems - IRS94*, Grenoble, France.

[Sauser and Billard, 2005] Sauser, E. and Billard, A. (2005). View sensitive cells as a neural basis for the representation of others in a self-centered frame of reference. In Demiris, Y., editor, *Third Workshop in Imitation in Animals and Artifacts*, Hatfield, UK.

[Schaal, 1999] Schaal, S. (1999). Is imitation learning the route to humanoid robots. *Trends in Cognitive Sciences*, 3(6).

[Schaal and Atkeson, 1998] Schaal, S. and Atkeson, C. G. (1998). Constructive incremental learning from only local information. *Neural Computation*, 10(8):2047–2084.

[Schaal et al., 2003] Schaal, S., Ijspeert, A., and Billard, A. (2003). Computational approaches to motor learning by imitation. *Phil. Trans. of the Royal Society of London: Series B, Biological Sciences*, 358(1431):537–547.

[Schenatti et al., 2003] Schenatti, M., Natale, L., Metta, G., and Sandini, G. (2003). Object grasping data-set. Lira Lab, University of Genova, Italy.

[Siciliano and Slotine, 1991] Siciliano, B. and Slotine, J.-J. (1991). A general framework for managing multiple tasks in highly redundant robotic systems. In *ICAR'91*, pages 1211 – 1216.

[Soueres et al., 2002] Soueres, P., Cadenat, V., and Djeddou, M. (2002). Dynamical sequence of multi-sensor based tasks for mobile robots navigation. *7th Symp. on Robot Control (SYROCO'03)*, 2:423– 428.

[Taylor, 2000] Taylor, C. (2000). Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *Computer Vision and Image Understanding*, 80.

[Tikhonov and Arsenin, 1977] Tikhonov, A. and Arsenin, V. (1977). *Solution of ill-posed problems.* Washington DC: Winston.

[Valero-Cuevas and Hentz, 1997] Valero-Cuevas, F. J. and Hentz, V. R. (1997). Anatomy and physiology of the human hand. In *Workshop on Human and Machine Haptics*, California, USA.

[van der Meer et al., 1995] van der Meer, A., van der Weel, F., and Lee, D. (1995). The functional significance of arm movements in neonates. *Science*, 267(5198):693–695.

[Vijayakumar and Schaal., 2000] Vijayakumar, S. and Schaal., S. (2000). Locally weighted projection regression: An o(n) algorithm for incremental real time learning in high dimensional spaces. In *ICML*, Stanford, USA.

[Vlassis and Likas, 1999] Vlassis, N. and Likas, A. (1999). A kurtosis-based dynamic approach to gaussian mixture modeling. *IEEE Trans. Systems, Man, and Cybernetics, Part A*, 29:393–399.

[Wampler, 1986] Wampler, C. (1986). Manipulator inverse kinematics solution based on damped least-squares solutions. *IEEE Trans. Systems, Man and Cybernetics*, 16(1).

[Weng, 1998] Weng, J. (1998). The developmental approach to intelligent robots. In *AAAI Spring Symposium Series, Integrating Robotic Research: Taking The Next Leap*, Stanford, USA.

[Wu and Huang, 1999] Wu, Y. and Huang, T. S. (1999). Capturing articulated human hand motion: A divide-and-conquer approach. In *ICCV (1)*, pages 606–611.

[Wu and Huang, 2000] Wu, Y. and Huang, T. S. (2000). View-independent recognition of hand postures. In *CVPR*, pages 88–94.

[Yang et al., 1994] Yang, J., Xu, Y., and Chen, C. (1994). Hidden markov model approach to skill learning and its application to telerobotics. *IEEE Transations on Robotics and Automation*, 10(5):621–631.

[Zöllner and Dillmann, 2003] Zöllner, R. and Dillmann, R. (2003). Using multiple probabilistic hypothesis for programming one and two hand manipulation by demonstration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, USA.