Computer Science and Artificial Intelligence Laboratory
Technical Report

MIT-CSAIL-TR-2007-015 March 2, 2007

Sensitive Manipulation

Eduardo Torres-Jara

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

CSAIL

Sensitive Manipulation

by

Eduardo Rafael Torres Jara

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January 2007

Author . . . .
Department of Electrical Engineering and Computer Science
January 26, 2007

Certified by . . . .
Rodney Brooks
Panasonic Professor of Robotics
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Sensitive Manipulation

by

Eduardo Rafael Torres Jara

Submitted to the Department of Electrical Engineering and Computer Science
on January 26, 2007, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

This thesis presents an effective alternative to the traditional approach to robotic manipulation. In our approach, manipulation is mainly guided by tactile feedback as opposed to vision. The motivation comes from the fact that manipulating an object implies coming in contact with it, consequently, directly sensing physical contact seems more important than vision to control the interaction of the object and the robot. In this work, the traditional approach of a highly precise arm and vision system controlled by a model-based architecture is replaced by one that uses a low mechanical impedance arm with dense tactile sensing and exploration capabilities run by a behavior-based architecture.

The robot OBRERO has been built to implement this approach. New tactile sensing technology has been developed and mounted on the robot's hand. These sensors are biologically inspired and present more adequate features for manipulation than those of state of the art tactile sensors. The robot's limb was built with compliant actuators, which present low mechanical impedance, to make the interaction between the robot and the environment safer than that of a traditional high-stiffness arm. A new actuator was created to fit in the hand size constraints. The reduced precision of OBRERO's limb is compensated by the capability of exploration given by the tactile sensors, actuators and the software architecture.

The success of this approach is shown by picking up objects in an unmodelled environment. This task, simple for humans, has been a challenge for robots. The robot can deal with new, unmodelled objects. OBRERO can come gently in contact, explore, lift, and place the object in a different location. It can also detect slippage and external forces acting on an object while it is held. Each one of these steps are done by using tactile feedback. This task can be done with very light objects with no fixtures and on slippery surfaces.

Thesis Supervisor: Rodney Brooks
Title: Panasonic Professor of Robotics

Acknowledgments

Pursuing a PhD degree at the artificial intelligence laboratory (now CSAIL) has been quite an experience. An experience that was very different from the one I had in mind at the begining of the program. I was lucky to work with whom I consider one of the greatest roboticist, Rod Brooks. He always encourages his students to abandond the status quo of the field to pursue unconventional ideas. However, learning to do that has been a challenge but it has been very rewarding as well. Thank you Rod for all your support and advice during all these years. This thesis would have not been possible without the advice of my committee Deb Roy and Steve Massaquoi. Thank you for encouraging me to develop a key technology that made this work possible and for showing me a complete different angle of my thesis.

During my time at MIT, I got to meet many interesting people that with their friendship made me feel at home. Joaquin, gracias hermano por tu sinceridad, generosidad, integridad y en pocas palabras gracias por tu amistad. Lijin, my third sister, thank you for your loyal friendship, for being there for me in good and bad times, for your extreme generosity, for your advice, and your great humor. Alberto, te conozco de tan poco tiempo y estoy muy agradecido de que seas mi amigo. Gracias por las conversaciones sobre todos lo temas que tocamos y por ser tan confiable “vecina”. Juan, my first friend at MIT, it has always been surprising for me the parallelism in our lifes. There are many things for which I would like to thank you. For proof reading my thesis, for your helpful advice during the writing, for the long conversations, etc. But the most important reason, thank you for being such a great friend. Thank you Ann, for being such a sweet person, for your advice and your positive attitude toward life. Thanks Paul for pushing me to make more sensitive tactile sensors, for your friendship, for the endless conversations at the tea conspiracy meetings, for your integrity, and for your good humor. Thanks Iuliu for your moral support and for putting up with me and my singing, for stopping your work to help me to debug the electronics, for helping to machine parts very late at night when I was discouraged, for providing fresh insides in my technology development, for being a good friend.

Thank you Carrick for your patience and help with my 3D printing. My host family at MIT was a very important emotional support. I feel blessed to have met them. Thanks John, Jery, and Nancy for all your love. I especially want to thank Lorenzo Natale for working with me and for his great contribution to this work. I enjoyed very much to work with him and I hope we continue our collaboration. I also like to thank Charlie Kemp for his encouragement and for contributing with some modules. To Lilla Zollei, thank you for your constant encouragement, your patience and for being a good friend. Thanks to Luis Perez-Breva for the endless discussions about politics, physics and any other random topic while we took coffee breaks. I also like to thank Alex Narvaez, Anna Custo, Carlos Mazariegos, Karen Gonzales-Valentin, Reina Riemann, Marisol Redin, and Robert Poor for their constant support. Thanks Alexis for being a great roommate and a good friend. Thanks for all the conversations about my work that helped to make things clear. Thanks to Una-May O'Reilly for taking care of the relation with the sponsors. My dear friends in Ecuador have been incredibly supportive during these years. Christina, thank you so much for your wonderful friendship and love. I cannot thank you enough for hosting me and my friends at your home, for listening me, for helping me with immigration paperwork, etc. Alvaro, thank you so much for the many interesting and funny conversations and thank you for sharing Christina's time. Santiago, thank you for the challenging conversations and for showing me always a different perspective in almost every subject. Vlady, thank you for you constant encouragement. Marcelo, thank you for always expecting more from me. This whole journey was motivated in part by you. Teresita, muchas gracias por su gran generosidad, cariño y su gran sentido del humor. Marina, thank you for being my moral support during the last stages of my thesis. I really appreciated the conversations, your honesty, and your great humor.

The most important of my thanks goes to my family. To my wonderful sisters, Patricia and Bibiana who had always been there for me and taught me about of strength and courage. To my parents, Cecilia and Teodoro, whose unconditional love and support has given the peace and the strength needed to achieve my goals. Thank you for your constant example and advice.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Current industrial robots are capable of executing many complex manipulation tasks, for example, robots assemble car's parts, such as bodies and engines, with high precision and speed. This is possible because the production line has been carefully engineered to make the robots perform successfully. For instance, the lighting and the placement of objects relative to the sensors are controlled, and the perception is specialized and model-based. However, these robots do not perform as well in slightly less controlled environments as the one at the beginning of the production line. Here, parts have to be picked up and placed in the right position for the robots to work. This task, known as bin picking, is simple for human workers but it is still an open problem for factory robots. If the environment is even less engineered, as is the case in a living room or a kitchen, these robots are not capable of dealing with the variable conditions. Tasks such as manipulating objects with different properties which have not been previously modelled become unfeasible.

Robots can perform more sophisticated tasks when they are teleoperated. For example, teleoperated humanoid robots have performed tasks, such as grasping everyday objects, using a power drill, throwing away trash, and retrieving a drink from a refrigerator. However, teleoperation is slow and demands significant effort on the part of the human operator. This is more noticeable in the robots exploring Mars where simple tasks such as placing a probe in a rock can take a long period of time (about a day). Moreover, some tasks even on Earth are not feasible due to communication delays. Such is the case of adjusting the grip when lifting an object. This is because by the time the teleoperator detects the slippage of the object, this has already fallen and the command sent to adjust the grip is useless.

Humans are much more adaptable. We observe people doing many dexterous tasks with their limbs. For instance, picking up a new object in a dynamic environment is not at problem for us. Because of this high performance, we look for inspiration in human manipulation to build robots capable of manipulating objects in poorly modelled environments.

It is believed that we use many sensory inputs to manipulate objects successfully. Visual input has been considered of paramount importance, therefore, most of the work done in robotic manipulation has taken this direction. That is, manipulation guided by vision. This approach has taken us to the current state of art in robotic manipulation.

However, one can easily think of examples where direct visual input is not used for manipulating object. Imagine, for example, waking up in your bedroom at midnight, reaching for the remote control, reorienting it and turning on the TV. All these steps were done in darkness. The complete task can be executed due to the information provided by the sensors located in the hand and arm. This information permits us to safely explore the environment, extract features to identify objects and control the interaction hand/arm/world. Moreover, experiments performed by Klatzky et al [61] have shown that blindfolded individuals were capable of identifying objects using only their hands with an accuracy of 96%. The dextrous manipulation performed by blind people is even more impressive.

Therefore, we think that tactile sensing is an specially appropriate sensing modality given that manipulation fundamentally relies on the contact between the robot and the world.

Information such as location(s) of contact, local shape of an object, and slippage are some of the parameters that tactile sensors could supply. These data would facilitate making decisions about the motion of a hand while grasping an object or about the force applied while gripping an object to avoid slippage. Those are parameters directly related to the hand/object interaction that would be difficult to obtain in a timely manner using other sensing modalities. Even though obtaining information from the direct physical robot/world interaction seems so useful, if not necessary, its use in robotics research has not been generalized due to technological limitations.

It is important to note that given the nature of tactile sensing, it is useful to actively increase the chances of the sensors coming in contact with objects to extract information. That is, using the tactile sensors to explore the surroundings. We can observe this in animals and humans who use their limbs to actively explore their environment.

Up until this time, we have not seen robots exploring their environments in general. The main reason for that is the design of today's robots. They are designed to position their limbs precisely using only their proprioception feedback. To achieve this high precision, robots limbs have to have high stiffness. High stiffness is not ideal for exploration because if an object crosses the robot's trajectory unexpectedly, the robot will not stop and it will damage either itself or the object. It is true that the robot can have sensors that detect this collision and if the actuators were fast enough the robot could stop in a timely manner. In practice, at the moment of the collision, only the passive elements of the robot determine the response of the system. Because the robot is stiff the collision most likely will not be gentle to the robot or the object.

Human and animals have limbs whose stiffness can be controlled but in general this stiffness is lower than that of a robot. This makes it possible to explore their surroundings in a safe manner which helps to adapt to changes in the environment. Therefore, we consider that having low impedance limbs combined with tactile sensing will allow us to implement more flexible manipulation.

In this thesis, we approach manipulation as a task mainly guided by tactile and force sensing and we use vision as a complement. We call this approach *sensitive manipulation*. In this approach, manipulation is as much about action as it is about perception and it is intrinsically responsive to the properties of the object being manipulated.

However, there are many challenges that need to be addressed to implement *sensitive manipulation.* Our solutions are presented on the following chapters.

1.2 The challenges

In this section we present some of the challenges that we faced while implementing *sensitive manipulation.* We briefly mention some of the solutions implemented in our robot OBRERO. Details of the robot are presented in chapter 2.

1.2.1 Contact vs Non-Contact sensors

Manipulating objects implies coming in contact with them. This is different from navigation, for example, where you want to avoid the obstacles and in general uses non-contact sensors (infrared, ultrasound and cameras). Non-contact sensors allow the robot to analyze the environment without greatly affecting it. In contrast, *sensitive manipulation* uses contact sensors (tactile and force sensors) to extract information. This means that a robotic limb, populated with contact sensors, should be capable of safely coming in contact with the environment to extract information using those sensors.

As mentioned earlier, traditional robots cannot safely come in contact with the environment because of their high stiffness.

In our approach, we deal with this challenge by building our robot with low mechanical impedance actuators. This feature allows the robot to safely come in contact with the environment which enables efficiently use contact sensors.

1.2.2 Coming in Contact with the object

The robot needs to come in contact with an object to interact with it. This event should be safe for the robot and the object since we do not want to damage either of them. Although this might sound like a simple task it is not always simple to perform it.

Coming in contact with an object is a nonlinear problem from the point of view of the control of the arm [104]. It implies to move from position control to force control, from a non-constrained environment to a constrained one. In practice, a combination of position and force control is preferred. To understand this, let's consider the following example. An arm (position control) comes in contact with an object, at this moment transient forces known as impact forces are applied to the arm. The controller is then switched to force control. The impact forces can be large and make the controller unstable. Moreover, when the robot's arm comes in contact with the object and the control is switched to force control, the arm can bounce because of the elasticity of the impact or the object can move. This makes the arm lose contact again but its mode is in force control instead of position control. This produces an oscillatory behavior.

Therefore, a hybrid control that uses position and force combined (compliance) is more adequate to avoid the abrupt switching. A more detailed analysis about this topic and alternative solutions are presented in [104].

OBRERO uses two degrees of compliance to deal with this problem. These degrees of compliance are given by the actuators and by the tactile sensors.

1.2.3 Sensing the hand/object interaction

In order to control the interaction between the hand and the object we need to have sensors capable of detecting parameters related to that interaction. For example, contact, curvature, normal, sheer forces, motion, and slippage are a few. In humans and animals these sensors are in the skin.

However, a major problem in robotic manipulation has been the lack of tactile sensors that actually work beyond the laboratory bench. Many attempts have been made to create tactile sensors (see chapter 3) but in most of the cases the actual conditions present in a robot have not been considered [43]. Consequently, even detecting contact with an arbitrary object is not a solved problem nowadays. Moreover, if we want the robot to manipulate objects, a useful property of a tactile sensor on a finger would be to deform in order to increase the contact surface with an object and as a consequence the friction.

The problem becomes a bit more complicated when the finger has low mechanical impedance. In this case the sensitivity of the sensor should be enough to detect the contact even when the fingers move due to their low impedance.

Certainly, alternative methods for detecting contact are available. For example, one method is to measure the torque on the proximal phalange of a finger using strain gauges instead of having surface sensors. This method works but the information that it can provide is limited. For example, more than one contact point or sheer forces cannot be detected.

As it will be described in more detail later, OBRERO uses a new kind of tactile sensor that provides information to detect contact from arbitrary directions, sheer forces, and rotational and linear slippage.

1.2.4 Using tactile data to guide motion

The traditional approach to manipulation models the interaction hand/object using an small number of contact points. At each contact point a normal force and a frictional cone are considered. This model makes the mathematics tractable.

However, in actual hand/object interactions this model does not always hold. For instance, the hand usually needs to have a flexible surface to distribute the pressure on the object. This is safe for the object because the stress on the contact surface is reduced. More importantly, a wider surface constrains the rotation of the object which is not usually considered in the model. If we include all these variables, obtaining a closed form expression will be complex.

In our approach, the robot hand has more points of contact with the hand and a wealth of information arises from the tactile and force sensors. Making sense of these data is definitely a challenge. Nevertheless, it opens the door to a greater understanding of the hand/object interaction.

Having a great number of data to make sense is a common scenario in artificial intelligence and computer science. Such is the case of Computer Vision where an RGB image of 360 × 240 pixels, 8 bits resolution at 7.5 fps generates 15,552,000 bytes per second. For such amounts of data many analysis tools have been developed. Compared with those numbers, the tactile sensors of our current robot only generates about 1,600 bytes per second.

In chapter 5, we present some of the processing done with these data to extract information useful for the task, literally at hand.

1.3 The approach: sensitive manipulation

We approach manipulation as a task mainly guided by tactile and force sensing and we use vision as a complement. In this approach, we consider that embodiment and behavior-based architectures are important elements as discussed in the following sections.

1.3.1 Embodiment: The robot OBRERO

In simulation, robots have performed sophisticated manipulation tasks such as grasping convoluted objects, tying knots, carrying objects around complex obstacles, and extracting objects from entangled circumstances. The control algorithms for these demonstrations often employ search algorithms to find satisfactory solutions, such as a path to a goal state, or a set of contact points that maximize a measure of grasp quality. For example, many virtual robots use algorithms for motion planning that rapidly search for paths through a state space that models the kinematics and dynamics of the world. Almost all of these simulations ignore the robot's sensory systems and assume that the state of the world is known with certainty. For example, they often assume that the robot knows the 3D structure of the objects it is manipulating.

The results of such simulations are in few cases directly transferable to actual robots. The reason is that the models assumed in simulation to represent parameters such as friction, contact forces, etc., do not hold in the world beyond the simulation.

For instance, tactile sensors are generally not even modelled in simulations. In cases in which they are considered, the contact points between the hand and the object are used as the tactile sensors readings. In actual robots, those contacts most likely will not be detected.

Brooks [13] has pointed out these problems and proposed embodiment as a solution that provides physical grounding. The physical grounding of the robot allows us to work with the actual data produced by the sensors and the actual actions executed by the actuators. Actual sensing and actuation determines in great part the kind of tasks that can be performed.

In the specific case of manipulation, embodied approaches have been limited by the current tactile sensing technology. To solve this problem we have developed tactile sensors (see chapter 3) that present helpful features for manipulation. These sensors have been tested in an embodied system: the robot OBRERO.

In our approach, we are capable of doing manipulation guided by tactile sensing because OBRERO has a body that provides information about the interaction with objects. Therefore, we consider embodiment of great importance to achieve *sensitive manipulation*.

1.3.2 Behavior-based architectures in manipulation

In order to achieve *sensitive manipulation*, we use a behavior-based architecture [1] that lets us deal with unknown environments. Traditionally, the trajectory of the robotic manipulator is completely planned based on a model of the world (usually a CAD model). This renders the manipulator incapable of operating in a changing environment (not to mention an unknown one) unless a model of the environment is acquired in real-time.

The same situation was already faced in mobile robotics and behavior-based architectures were introduced as a successful alternative. However, the transition to manipulation is not straight forward because of the nature of the variables involved. For instance, in a model-based approach the trajectory of the robot's limb is carefully planned and the hand only comes in contact with the environment in a carefully control manner. Unexpected contacts are generally dangerous and avoided.

If we now consider a behavior based approach, we can easily think of behaviors such as: move the finger following the surface of the object, move the hand until it makes contact with the object, or move the hand down to the bottom of the object while maintaining contact assuming the object is supported. All these behaviors rely on coming in contact with the environment.

Therefore, we believe that to take advantage of a behavior-based architecture in manipulation, the robot used should be capable of safely coming in contact with the environment and detecting this contact.

In the robot OBRERO, we use passive elements embedded in every actuator of the robot's limb. These actuators, known as series elastic actuators [88,111], make coming in contact with the environment safe. This makes it possible to use a behavior-based architecture on OBRERO.

1.4 Contributions

1.4.1 Overall contribution

The overall contribution is our approach to manipulation named ***sensitive manipulation***. Sensitive manipulation is about purposely guiding the motion of the robot's limb using mainly tactile information. In this approach, the robot's limb is not only an actuator device but also as a sensing device.

In this thesis we use sensitive manipulation to implement grasping. This implementation is enabled by the development of adequate technology also presented in this thesis.

To implement this approach the following points are considered important.

- The robot needs to have tactile sensors to extract information of the environment.

- The limb has to be compliant to gently interact with the environment. This compliance limits the precision of the limb in general but enables exploration.

- The capability of exploring together with the capability of sensing contact (dense tactile sensing) compensate for the lack of precision. For very many applications knowing that you are in contact with the object is more useful than precisely positioning the limb.

1.4.2 Technological contributions

Most robots are not dexterous in part because the technology needed is not available. In this thesis, we present five technological contributions to robotics manipulation. The first four enable the last contribution.

- The first is a new tactile sensing technology. This technology allows us to easily build **compliant tactile sensors** whose features are convenient for manipulation. Some of those features include: adequate shape to come in contact with objects, mechanical flexibility, high friction coefficient and high sensitivity.

- The second contribution is a new **scalable force actuator**. This actuator is designed to be cheap, easy to assemble, and easy to scale to small dimensions. This actuator operates with the same principle as a series elastic actuator. However, its mechanical simplicity enables its use in very constrained applications such as actuating a robotic hand.

- The third contribution is a new **force controlled robotic hand**. This hand has passive and active degrees of freedom and it can be controlled either using position or force control. Each one of its eight joints uses a version of the actuator above mentioned.

- The fourth contribution is an integrated platform named **OBRERO**. It integrates the tactile sensing technology, the actuators and the hand, mentioned above, with a force controlled arm and a head with two optical degrees of freedom (zoom and focus). The integration allows us to implement a behavior based approach to manipulation. OBRERO means human worker in Spanish.

- The fifth contribution is: the first humanoid robot that actually **integrates tactile information for guiding manipulation**. The robot is capable of coming in contact with an unknown object, use tactile feedback to move the hand around the object, and lift it. This integration is possible because of the first four contributions.

1.5 Road Map

This thesis is organized as follows. Chapter 2 presents the design and the implementation of the platform for *sensitive manipulation*, the robot named OBRERO. Especial attention is given to the robot's hand and its actuators. The tactile sensors used are presented in detail in chapter 3, where we cover the design, implementation and evaluation of these sensors. In chapter 4, we present some observations on humans that inspired part of this work.

An integrated system to approach manipulation using mainly tactile information is presented in chapter 5. The details of the different modules implemented as well as the integration are included. The chapter ends with the results obtained. *Sensitive manipulation* is contrasted with alternative approaches in chapter 6. In this chapter we also summarizes some of the lesson learned.

In chapter 7 we present alternative methods that learn about objects using sound and use shadows to estimate distances.

Chapter 2

OBRERO: A platform for Sensitive Manipulation

2.1 Introduction

In this chapter, we present the design, construction and evaluation of a humanoid platform (OBRERO) suitable for sensitive manipulation. The design of the platform is motivated by human manipulation. Humans are capable of manipulating objects in a dexterous way in unstructured environments. We use our limbs not only as pure actuators but also as active sensors. Human manipulation is so sensitive that many tasks can be accomplished using our hands without any help from vision. In contrast, humanoid robots in general are limited in the operations they can perform with their limbs alone.

However, if we consider tasks such as precise positioning or accurate repeated motion of an arm, we notice that, in general, humans are outperformed by robots because human limbs are clumsier than robotic ones. This apparent disadvantage is overcome by the great number of sensors and actuators present in human limbs which allows us to adapt to different conditions of the environment.

For instance, humans use their hands to touch or grab an object without damaging themselves or the object. This is possible because humans can control the force and the mechanical impedance exerted by their limbs when in contact with an object.

Robots, in general, cannot do this because their components lack the sensing and actuating capabilities needed to control these parameters i.e. force and impedance.

Moreover, the sensing capabilities of human limbs are not limited to force. Humans can also extract many features of an object they are holding [61] thanks to their highly innervated skin. In contrast, robotic limbs have a limited number of sensors, rendering them inadequate for feature extraction.

Motivated by these ideas, we have favored the sensing capabilities over the precision in the design of OBRERO's limb. The limb has force control, low mechanical impedance, as well as position and force sensing.

We use non-conventional actuators for the limb and dense tactile sensors for the hand (special attention is paid to the actuators in the hand because of size constrains). These actuators control the force and reduce the mechanical impedance. These features allow the limb to come in contact with objects in a safe manner. For instance, when contact occurs the platform needs to respond fast enough to avoid damaging itself or the object. In practice, when the limb comes in contact with an object, the passive elements of the system are the ones that determine the response. Therefore, these passive elements must have a low mechanical impedance to achieve contact compliance. This property is especially important when using the limb as an active exploring device.

While tactile information will dominate, *sensitive manipulation* also can benefit from visual and auditory perception. Such information will be used by the robot to improve the efficiency of manipulation, rather than be an essential prerequisite. Vision can give a quick estimate of an object's boundary or find interesting inhomogeneities to probe. Sound is also a very important clue used by humans to estimate the position of an object and to identify it. We have conducted experiments to take advantage of this fact in section 7.1 and [99]. The robot OBRERO has a 2 degree-of-freedom head that includes vision and sound. The camera has two optical degrees of freedom; focus and zoom. Focus is very useful to obtain depth information and zoom helps to obtain fine details of an image. The vision system will try to take advantage of natural cues present in the environment such as shadows, see section 7.2 and [36].

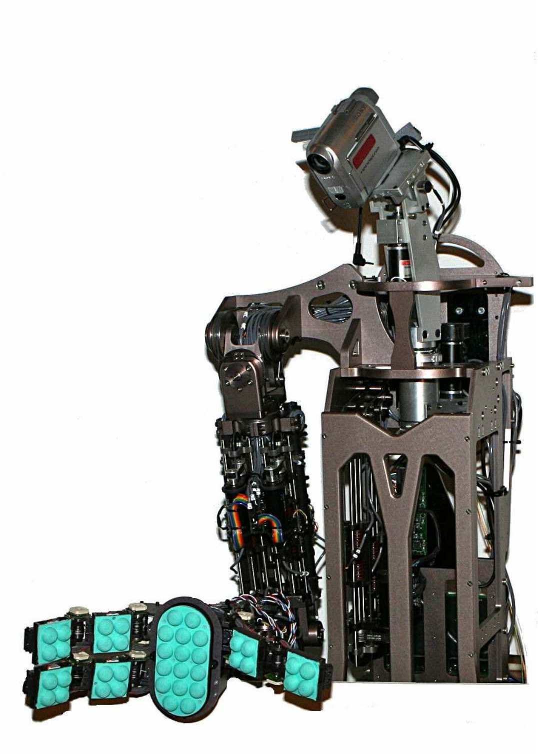

Figure 2-1: Robot OBRERO. The robot has a highly sensitive and force controlled hand, a force controlled arm and a camcorder as a head. OBRERO's hand has three fingers, 8 DOF, 5 motors, 8 force sensors, 10 position sensors and 160 tactile sensors.The head has 2 mechanical DOFs and 2 optical ones (Zoom and Focus).

This robot has been designed to approach manipulation using a behavior-based architecture [1]. A discussion for using behavior-based architecture is presented in section 1.3.2.

In this chapter we present the design and implementation of the robot OBRERO. A review of robotic platforms for manipulation is presented in section 2.2. The overall architecture of the robot is presented in section 2.2.1. The design and implementation of the hand and its actuation system are described in sections 2.2.3 and 2.2.2. The robotic arm and head are described in sections 2.2.4 and 2.2.5.

2.2 Platforms for Robotic Manipulation

In robotics, several researchers have designed and constructed arms with different features depending on the application to address. For example we can mention:

Milacron's arm, PUMA 560, WAN [101], DLR arm [41] and Cardea's arm [28]. The same applies to the design of hands where we can mention: the MIT/Utah's [50], the Stanford/JPL [89], the Barret's [102], DLR's [16] and the Shadow's [23] hand. There is also a wealth of work in the area of wrists. However, there are only a few platforms that have been constructed to research manipulation as a whole. Not surprisingly most of these platforms are humanoid robots. In this section, we will pay attention to these platforms.

- **Dexter** is a humanoid platform which has two Whole Arm Manipulators (WAM) [101], two Barrett hands [102], and a BiSight stereo head.

  The WAM arms have 7 direct-drive DOFs and cable transmission for force control. Each of the hands has 3 fingers and 4 DOFs. One DOF for each finger and one for rotating the fingers. The tips of each finger have an ATI load cell for force sensing. The BiSight stereo head can pan, tilt, and independently verge each camera. The cameras have control of focus, iris, and zoom. The head also has a binaural acoustic sensor consisting of four microphones. A VME architecture is used for computation.

  The work implemented in this platform [54, 56] shows an extensive use of force sensing in the fingers to deal with objects of unknown geometry. The speed of operation is limited, in part because of the lack of compliance in the fingers.

- **Robonaut** is a humanoid robotic platform designed to operate in space. It consists of a 2 DOF head (pan/tilt) and stereo cameras, two 7 DOF arms with force/torque cells at each shoulder (16 embedded sensors at each DOF), and two 14 DOF hands [70] whose design is based on the MIT-UTAH hand. The tactile sensing is still in development (miniature force cells for the fingertips) but currently uses FSRs and QTC resistors. This robot was designed to manipulate the same kinds of tools that humans do in space, controlled by teleoperators. However, due to the time-delays in communication the platform is becoming more autonomous.

Autonomous and semi-autonomous manipulation uses force-sensing from a few force cells in the shoulder and wrist. The arms are designed for high stiffness and consequently the harmonic geardrives are prone to damage. To solve this, the robot is covered with padding. Therefore, this platform is not fully designed to conduct exploration with its limbs.

- **Cog** is a humanoid robot designed to study embodied intelligence and social interaction. Cog has twenty-two mechanical DOFs: two 6 DOF arms, a 3 DOF torso, a 4 DOF neck, and 3-DOF combined in its eyes. The actuators in the arm are series elastic actuators [111]. Its design allows the robot interact safely with its environment and with people. These capability have been exploited in [112], [113] and [31].

- **Babybot** is composed of a head, an arm and an anthropomorphic hand. This robot was built to study sensory motor development and manipulation [81].

- **Domo** is a humanoid robot designed to study general dexterous manipulation, visual perception, and learning [27]. It has two arms and hands and a head. Its limbs use series elastic actuators [111].

- **Saika** is a humanoid robot [62] that consists of a two-DOF neck, dual five-DOF upper arms, a torso and a head. The hands and forearms used were designed according to the tasks to perform. The control used was behavior-based. Some of the goals of the robot were: hitting a bouncing ball, grasping unknown objects and catching a ball [63].

2.2.1 Robot Hardware Architecture

The robot OBRERO is shown in figure 2-1 where we can observe the hand, arm(originally created for the robot Domo [27]) and head. OBRERO's overall hardware architecture is presented in figure 2-2. In this figure we can observe that the hand and head controllers connect to different linux boxes using RS232 and that the arm controller uses CAN. The details about the hand, arm and head controllers are explained in

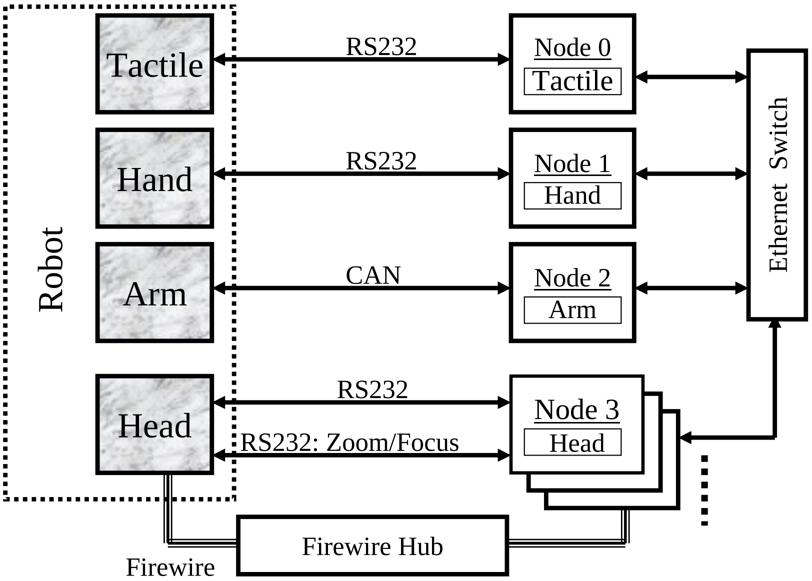

Figure 2-2: Overall architecture of OBRERO. The motor controllers of the Hand, Arm and Head are connected to different linux nodes via a serial RS232 or CAN. The boxes inside nodes 0,1,2,and 3 represent processes that handle the communication between a linux node and a motor controller or the tactile hardware. A network of microcontrollers reads the data from the tactile sensors and send the data to a linux node via RS232. The head is also connected to all the computers using a firewire network for acquiring images/sound and via RS232 to control zoom and focus.

sections 2.2.3, 2.2.4 and 2.2.5, respectively. The linux computers are interconnected by an 100 Mbps ethernet network. One of these nodes connects to the head via RS232 to control the zoom and focus of the camera. The camera is also networked with all of the computers by a firewire network to acquire images and sound. The details about these connections are described in section 2.2.5.

In this architecture, the information from each of the low level motor controllers (hand, arm, head) and from the tactile sensors is collected by linux nodes and made available to any software module. This feature gives flexibility for the software implementation because a program can run in any node independent of the feedback information needed.

2.2.2 Small and compliant actuator

In order to have a compliant hand, we need to have compliant actuators in its joints. An actuator that complies with this requirement is a series elastic actuator (SEA)

[88, 111], however, it presents problems (described later) when they are to be used in small mechanisms. Consequently we started by designing an actuator that fits our specifications.

We start by defining SEA. SEAs are comprised of an elastic element, i.e., a spring, in series with a motor (see figure 2-3). By measuring the deflection of the spring, one can determine the force being applied by the system. Given that the spring is the only connective element between the actuator and the output, SEAs effectively reduce the mechanical impedance of the system. This can be better explained with an example: Imagine a robotic link actuated by an SEA. Any external force applied to the link will only be resisted by the flexible spring as opposed to the high inertia projected by the gearhead reduction. Therefore, the mechanical impedance of the whole system is defined by that of the spring.

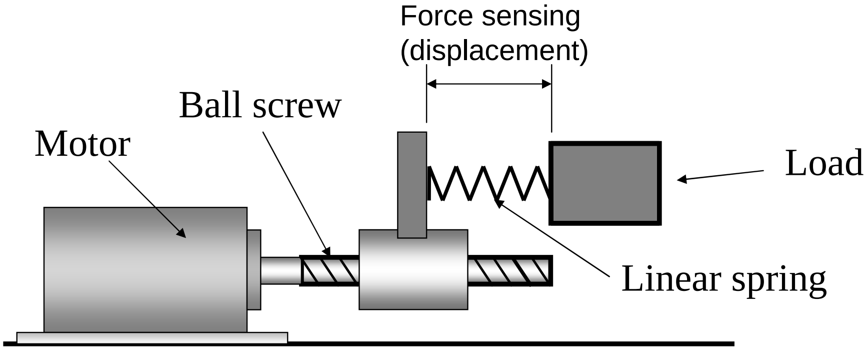

Figure 2-3: Conceptual depiction of an SEA, comprising a spring in series with a motor.

Although the spring also affects the reaction speed, or bandwidth, of the system, speeds still fall within an appropriate operational range for control applications. As a physical shock absorber, the spring also makes the robotic system less susceptible to and inherently reactive to unexpected impacts.

Traditionally there are both linear and rotary SEAs. The linear version requires precision ball screws to control the spring deflection. Although allowing for good mechanical transmission reduction, this constraint makes the system expensive and puts a limit on how small it can be. Conventional rotary SEAs require custom-made torsional springs, which are hard to fabricate and very stiff. This stiffness practically obviates the benefits of an elastic element. Furthermore, the torsional spring deflection is generally measured by strain-gauge sensors that are cumbersome to mount and maintain. Both of these linear and rotary SEAs present joint integration problems.

Therefore, we designed and built a different actuator that is compact, easily-mountable and cheaper to fabricate while maintaining the features of SEAs [98].

Design

Figure 2-4 is a CAD rendering of the actuator. As will be explained, the module is comprised of: an actuated part, a transmission mechanism, and a mobile part. The actuated part is connected directly to the motor apparatus, while the mobile part is integrated into the moving output. The transmission mechanism transmits the motion from the actuated to the mobile part.

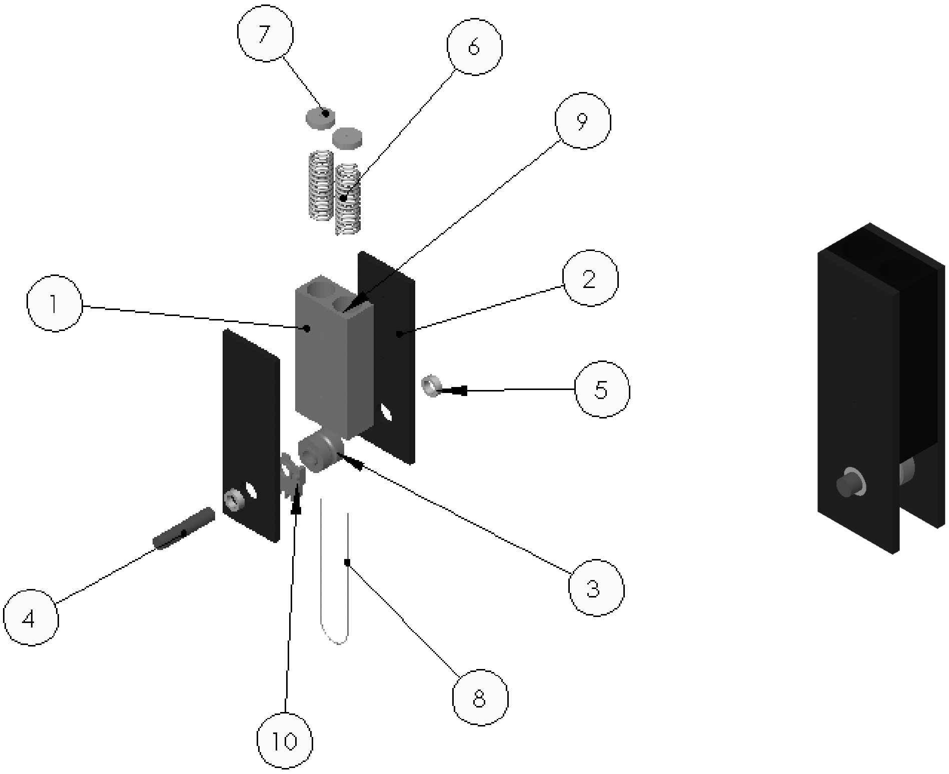

Figure 2-4: The force control actuator as a whole and an exploded, annotated view.1.Springbox 2.Plate 3.Wheel 4.Shaft 5.Bearing 6.Spring 7.Lid 8.Cable 9.Sping housing 10.Potentiometer

Description of the module functionality is easiest when we first consider the relationship between these subparts. The mobile part is composed of a spring box (1) sandwiched between two side plates (2). The actuated part consists of a wheel (3) fixed to a shaft (4) (which may be the motor shaft) and a potentiometer (10). The shaft/wheel assembly is mounted on bearings (5) in the two side plates. Therefore, the mobile part and the actuated part can rotate freely with respect to one another-at this point in the discussion, they are connected only by bearings. In other words, if a motor rotates the shaft, the wheel will move but the mobile part will not. The function of the potentiometer will be described below.

The transmission mechanism consists of two compression springs (6), two lids (7), and a cable (8). Each spring sits within a cylindrical, close-fitting shaft (9) in the spring box. An opening at one end of the shaft allows for spring and lid insertion.

The other end of the shaft has a flat bottom (against which the spring compresses) with a hole for the cable to pass through (see figure 2-5). Starting from its termination on one of the lids, the cable then runs through the center of one spring, out of the spring box, around the wheel and similarly back up through the other spring to terminate on its lid. The cable is fixed on the wheel to prevent it from slipping. Therefore, the springs can be effectively compressed between the bottom of their shafts and their respective lids through the tension in the transmission cable. Furthermore, both of the springs are initially compressed half way (described below). Due to this arrangement, any compression in one spring is reflected as an extension in the other. Because of this pre-compression of the springs, the routing of the cable during assembly may seem complicated; however, we present an elegant solution to this fabrication problem in Section 2.2.2.

In order to describe how the actuator works, we refer to the diagrams in figure 2-6. In the first diagram, we have a depiction of the wheel (of radius R) mounted on an axle and the cable. The cable is fixed to the wheel at Point A. The torque of the motor T_m is applied directly to the wheel's axle such that

$$T_m = R(F_1 - F_2) \tag{2.1}$$

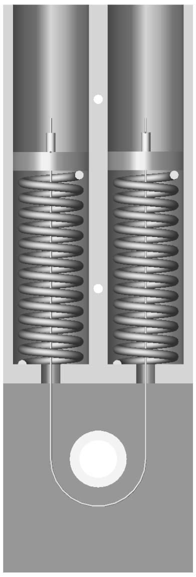

Figure 2-5: A cut-away view of the springbox showing the embedded springs (in their initial equally pre-compressed state) with their shafts and lids. As you can see, the cable terminates above each lid, runs down through the axis of each spring, and proceeds out the holes in the bottom of the shafts.

where F_1 and F_2 are the forces transmitted to the cable by the wheel .

The second diagram includes the springs. Each spring has rest length L and spring constant k, is capable of maximum compression C, and is initially pre-compressed to

$$L - x_i = C/2 \tag{2.2}$$

where x_i is the pre-compressed resting height of the springs. In order to keep the response of the actuator linear, its range must be maximally constrained to $\pi/2$ radians on either side of Point A. Pre-compressing the springs to $C/2$ ensures that the springs will not escape their shafts over the operational forces of the module. In other words,

$$C/2 = \phi_{max} R | 0 < \phi_{max} \leq \pi/2. \tag{2.3}$$

When the module (i.e., the mobile component) encounters a force such that it is held rigidly in place, we get a displacement of the springs (as shown in figure 2-7). That is, one of the springs is compressed a distance x and the other is extended by the same amount. Because the force on each spring is given by $F = k(x_i + x)$ and

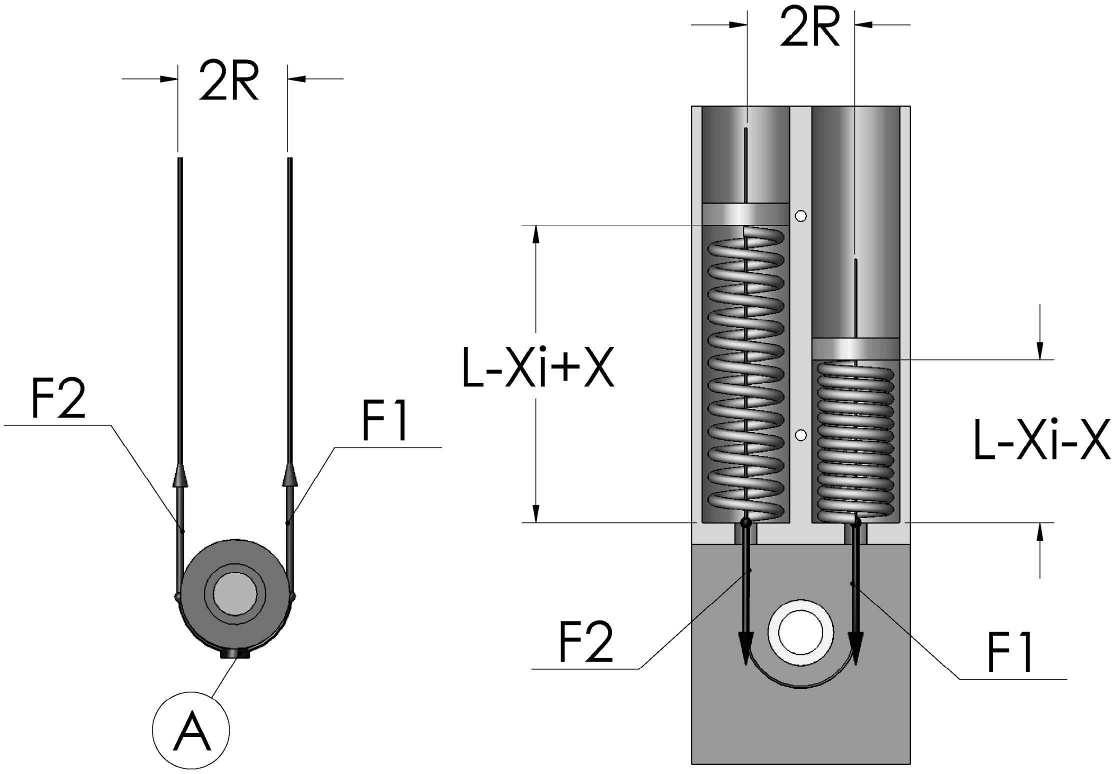

Figure 2-6: Free body diagrams of the actuator.

because this linear displacement is directly related to the rotation of the wheel by $x = \phi R$ (see figure 2-7), we end up with an output torque T_o:

$$T_o = 2k\phi R^2. \tag{2.4}$$

The potentiometer measures the relative displacement between the wheel (essentially the motor shaft) and the mobile part (essentially the moving link) to give us ϕ. Thus the actuator provides an effective measurement of the applied torques by measuring linear displacement of springs.

In order to control the torque applied by the actuator, a proportional controller is used. The controller acts on the position of the wheel with respect to the mobile part. This position control is enough to control the torque because of equation 4.

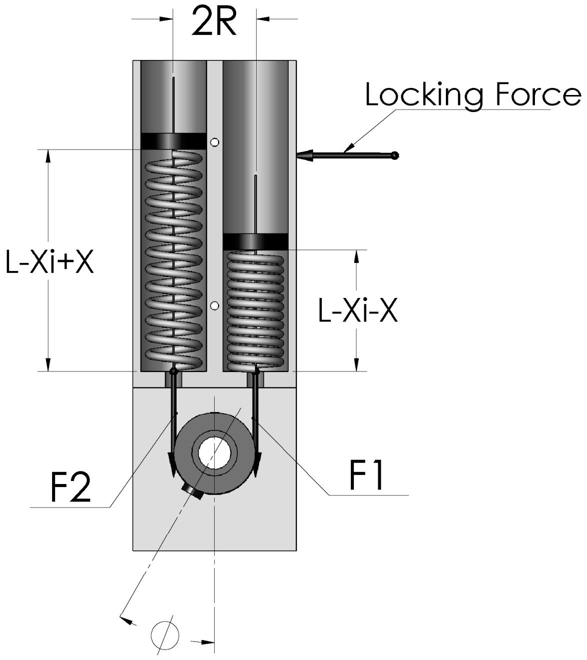

Figure 2-7: Actuator configuration with applied force holding it in place.

Mounting

One particularly important aspect of this module is its fabrication process. The whole actuator was designed to enable the pre-compression of the springs. Figure 2-8 depicts the assembly process which we describe in steps:

1. The wheel is fixed to the shaft which is mounted into one of the side plate bearings.

2. The springs are inserted into their shafts in the spring box and covered with their lids. The spring box is then temporarily placed tangent to the wheel (a distance of C/2 from its operational position).

3. A cable is threaded through the hole in one of the spring lids, down through the center of that spring, out of the spring box, around the wheel and similarly up the other side. The cable is then fixed to the wheel directly across from the spring box to keep it from slipping.

4. Next, cable terminators are crimped to the ends of the cable, flush with the outside of each lid. This is the configuration depicted in figure 2-8.

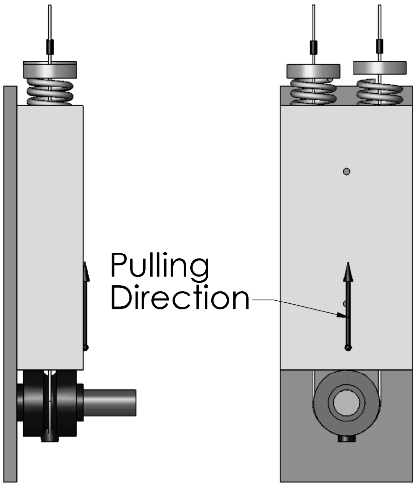

Figure 2-8: Mounting of the actuator.

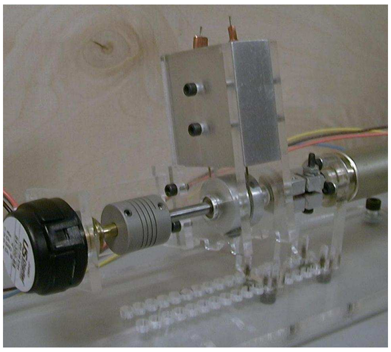

Figure 2-9: A testing prototype of the actuator made of acrylic and aluminum. The motor is off to the right.

5. Pulling the spring box away from the wheel effectively compresses the springs. The distance between the wheel and the operational position of the spring box was intentionally designed to be half the total compression of the springs. Thus by then fixing the spring box to the side plates, in its operational position, we have equally compressed both springs half way (see figure 2-5).

6. The modified potentiometer (or an encoder) is then attached to the wheel and the other side plate, which is finally mounted to the assembly.

Characterization

The prototype depicted in figure 2-9 was used to evaluate the linearity and the time-response of the actuator.

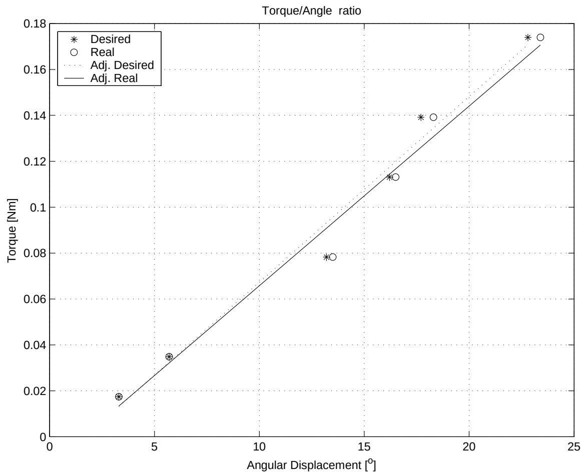

Figure 2-10: The figure shows the linear relation between the torque applied and the angular displacement in the actuator. We can also observe that the controller produces a torque close in value to the desired torque.

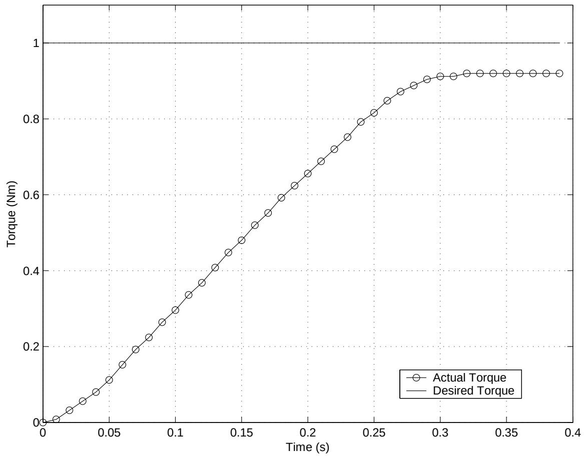

Figure 2-11: Time-response. We can observe that the actuator takes some time to deliver a desired torque.

The linearity of the actuator was determined by measuring the actual torque applied to the load and recording both the desired and actual displacement angles. The ratio between the torque and the angle yields the spring-constant of the actuator. The results of this test, shown in figure 2-10, demonstrate that the ratio between the actual displacement and the applied torque is fairly linear and confirm the expectations of the design. The actuator constant determined was 0.008 Nm/degree.

In order to determine the time-response of the actuator, we lock the mobile part and then apply a step function to the input of the actuator. The step function is applied by sending a desired force to the actuator at time zero. The controller of the actuator will move the motor until the desired angle is achieved. Given the current conditions, the motor will deform the springs and by measuring the displacement of the shaft with respect to the mobile part of the actuator, we determine the applied torque. In figure 2-11 we can observe the time it takes for the actuator to apply the desired torque to the load. We also observe that the damping of the system introduced by the springs is quite considerable. The rise time, defined as the time it takes the actuator to go from the 10% to 90% of the desired output, is around 270 ms.

2.2.3 Hand Design

General Description

In designing the hand we considered the following features important:

- Flexible configuration of the fingers
- Force sensing and mechanical compliance
- High resolution tactile sensing

As we describe the parts of the design, we will discuss the implementation of these features.

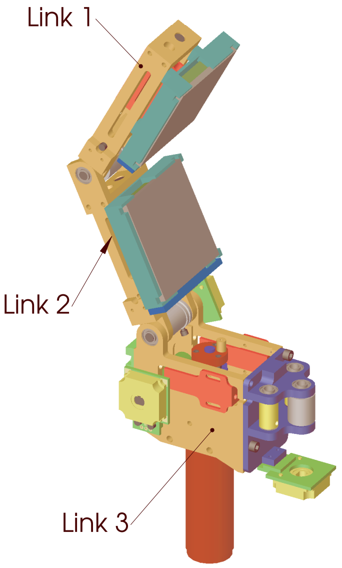

Figure 2-12: CAD rendition of a finger. It is comprised of three links. Link 1 and 2 have tactile sensors and their movement is coupled. Each of the three links is actuated using SEAs.

Finger Design

We start with the description of a finger. The size of each phalange of the fingers was imposed by the size of the tactile sensors originally considered. Each finger consists of three links as depicted in figure 2-12. Links 1 and 2 are coupled with a ratio of 3/4. The axes of these two links have an actuator, which is described in section 2.2.2. This actuator has several functions: reading the torque applied to the axes, reducing the mechanical impedance of each link, and allowing the two links to decouple their movement.

The first two functions are common features of this kind of actuator and the last one is a consequence of the actuator construction.

This decoupling is useful to do grasping as described in [102]. For instance, we can observe in figure 2-13 that when link 2 contacts an object, link 1 can still keep moving to reach the object, while link 2 continues to apply force on the object.

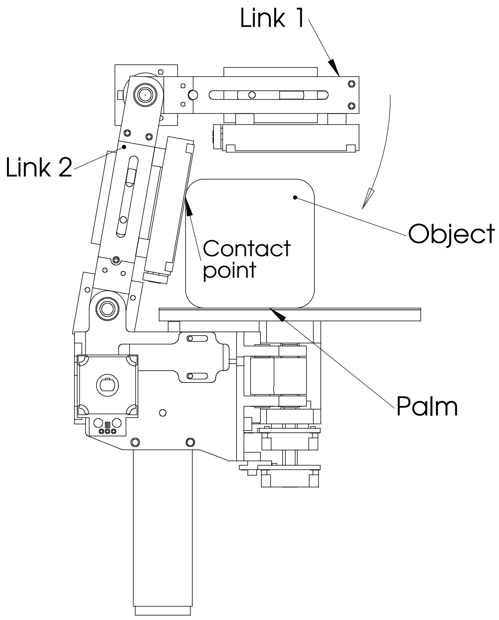

Figure 2-13: Link 2 has made contact with an object and stopped moving but keeps pressing against the object. Link 1 continues moving.

In order to move links 2 and 1, there is a motor located on link 3. The torque is transmitted using a cable from the motor to the two actuators on their respective links (see figure 2-14). Cable is used as a transmission mechanism because unlike gears it does not have backlash problems. The different diameters of the wheels of the actuators determine the transmission ratio.

An important consideration when we are working with cables is the tension mechanism. The size of the tension mechanism in this case had to be small so that it could fit inside link 3. The mechanism used is described in figure 2-14.

In this figure, we can also observe the presence of an idler wheel that helps to route the cable and has a potentiometer attached to its axis to determine the absolute position of the links when they are coupled. When they are decoupled we need to consider the information available in the actuators to get the absolute position.

Tactile sensors are mounted in links 2 and 1. The details of these sensors are described in section 2.2.3. The sensor used are sensitive to small forces in any direction,

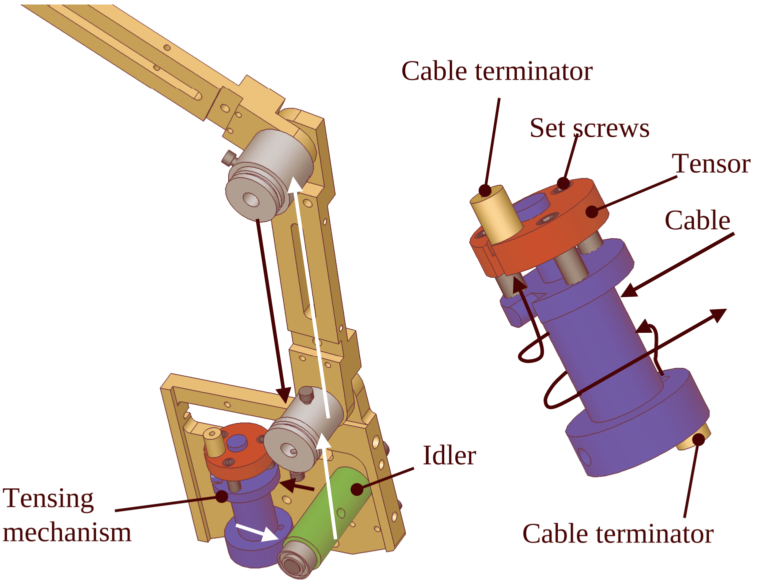

Figure 2-14: On the left we can observe the cable routing in a finger. The cable comes from the tensing mechanism, goes under the idler wheel and continues to the wheels on each axis. On each of these wheels the cable is wrapped around and clamped using the screws shown on the wheels. The cable wrapped on the top wheel goes down, wraps around the lower and the idler wheels and ends on the tensing mechanism. A detail of the tensing mechanism is shown on the right of the figure. It consist of a wheel that goes connected to the motor and a lid that slides on a shaft. The cable with a terminator comes from the bottom of the wheel, continues its trajectory as described before and ends with another terminator on the lid. The lid tenses the cable by increasing the distance between itself and the wheel using the setscrews. The setscrews fit in holes that avoid rotation of the lid.

conform to the objects that is being touched and increase the friction between the object and the hand.

An extra feature of the finger, derived from the actuator, is the possibility of bending to push objects, as described in figure 2-15. This feature is a consequence of the low mechanical impedance of the actuator and is very important when the hand comes in contact with an object. This deflection allows OBRERO to detect the collision, conform the finger to the object, and minimize the chances of damage.

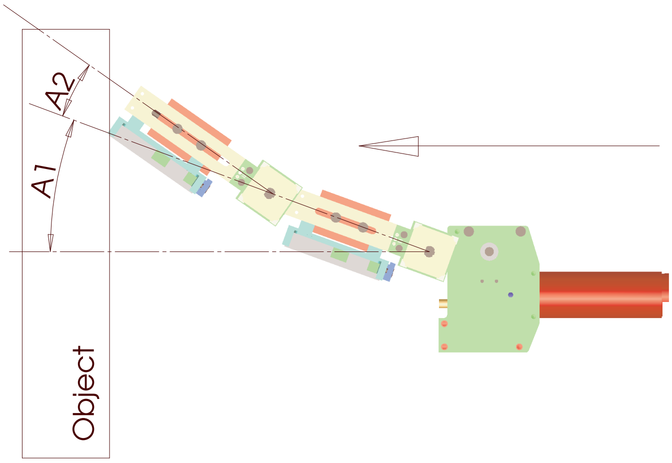

Figure 2-15: When a finger pushes against an object, it passively bends and does not break thanks to the mechanical compliance of the actuators.

Palm and Three Fingers Design

The hand is comprised of three fingers, each like the one described above, arranged around a palm as shown in figure 2-16. In this configuration, finger 2 is fixed with respect to the palm but fingers 1 and 3 can move in the direction shown by the arrows. Fingers 1 and 2 can be opposed to each other as a thumb and index finger in a human hand. Fingers 1 and 3 can also be opposed as shown in figure 2-17. The degree of freedom of the fingers 1 and 3 around the palm allow the hand to arrange the fingers to obtain an adequate configuration for grabbing objects with a variety of shapes.

The rotation of fingers 1 and 3, respect to the palm, is driven by a variation of the actuator described in section 2.2.2. This provides these fingers with the advantages described earlier. The torque for each axis is provided by a DC motor which transmits movement through a cable mechanism. However, the cable tensing mechanism is simpler than the one on the fingers because it does not have to move coupled links. Therefore, the tensing mechanism of the actuator is enough. This mechanism is shown in figure 2-18.

The palm has the same kind of tactile sensor as the one on the fingers and is described in section 2.2.3).

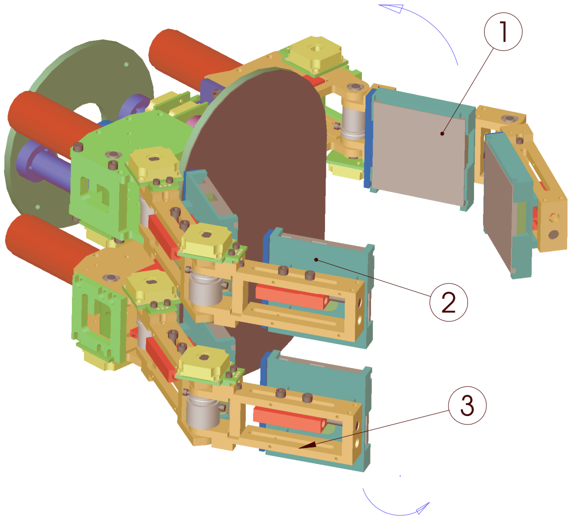

Figure 2-16: The arrangement of the fingers around the palm. Finger 2 is fixed to the palm while fingers 1 and 3 move up to 90° in the directions indicated by the arrows.

Tactile Sensor

In the original design we considered tactile sensors with high spatial resolution. The best option was using a mouse pad composed of force sensing resistors (FSR). A touch pad from Interlink Electronics provides an array of FSRs whose density is 200 dots/inch and 7 bits magnitude of the force/pressure applied. These sensors report

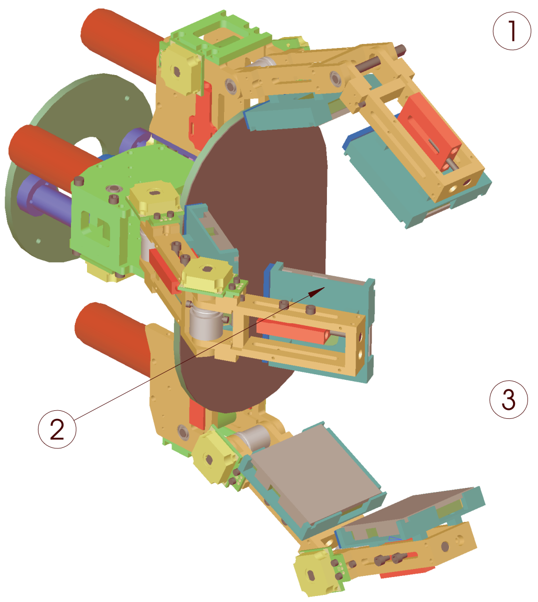

Figure 2-17: The hand is shown with fingers 1 and 3 rotated to their angle limit(90°).

the coordinates and the force of a point of contact. When there is more than one point of contact with the pad, it reports only the average force at the center of mass of the points of contact. The original application of these pads is reading pen strokes from human users, therefore, the spatial resolution is high.

However, these sensors, as many others tested, do not provide the information needed for manipulation tasks. In order to have a reading from these sensors, an object has to come in contact at the right angle. This does not work well in a robotic finger where we want to detect contact independently of the force direction. Moreover, these sensors are not mechanically flexible and their surface does not provide much friction when in contact with an object.

The solution to this problem was to develop a new kind of tactile sensor that has some of the properties mentioned above. A detailed explanation of this tactile sensor is presented in chapter 3. In figure 2-19 we can observe the two tactile sensors described.

On each phalange of the finger there are four domes (see figure 2-20) and each dome

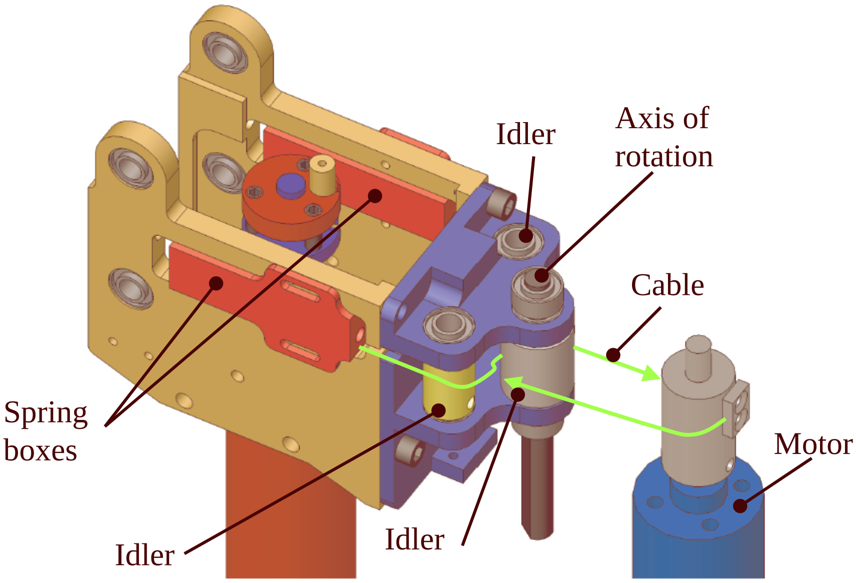

Figure 2-18: We observe that the cable comes out from the spring box and turns around two idlers before going to the motor. The idlers help to route the cable. From the motor, the cable returns to the idlers in the axle and goes towards the other spring box. The spring boxes are pulled by screws placed in the back. These screws are not shown in the figure.

has four electrical signals. The 16 signals are sent to a 16F876 PIC microcontroller. On the palm there are 16 domes whose 64 signals are connected to a microcontroller. The 7 microcontrollers (two per finger and one on the palm) are connected to 7 microcontrollers on the forearm of the robot via RS232. The latter 7 microcontrollers are networked via SPI to another microcontroller that connects to a linux node as shown in figure 2-2. The communication with the linux node is via RS232.

Additionally, on the lower side of the palm the robot has an infrared sensor to detect proximity to objects. This sensor is used instead of a touch sensor because, in

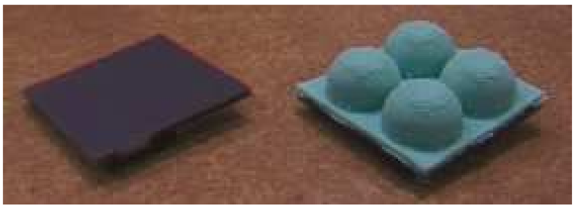

Figure 2-19: Left: A commercial FSR tactile sensor with high spatial resolution (200 point per square inch), flat surface, rigid ,and low friction coefficient. Right: OBRERO's tactile sensor with low spatial resolution (about 16 points per square inch), dome-like surface, flexible, and higher friction coefficient.

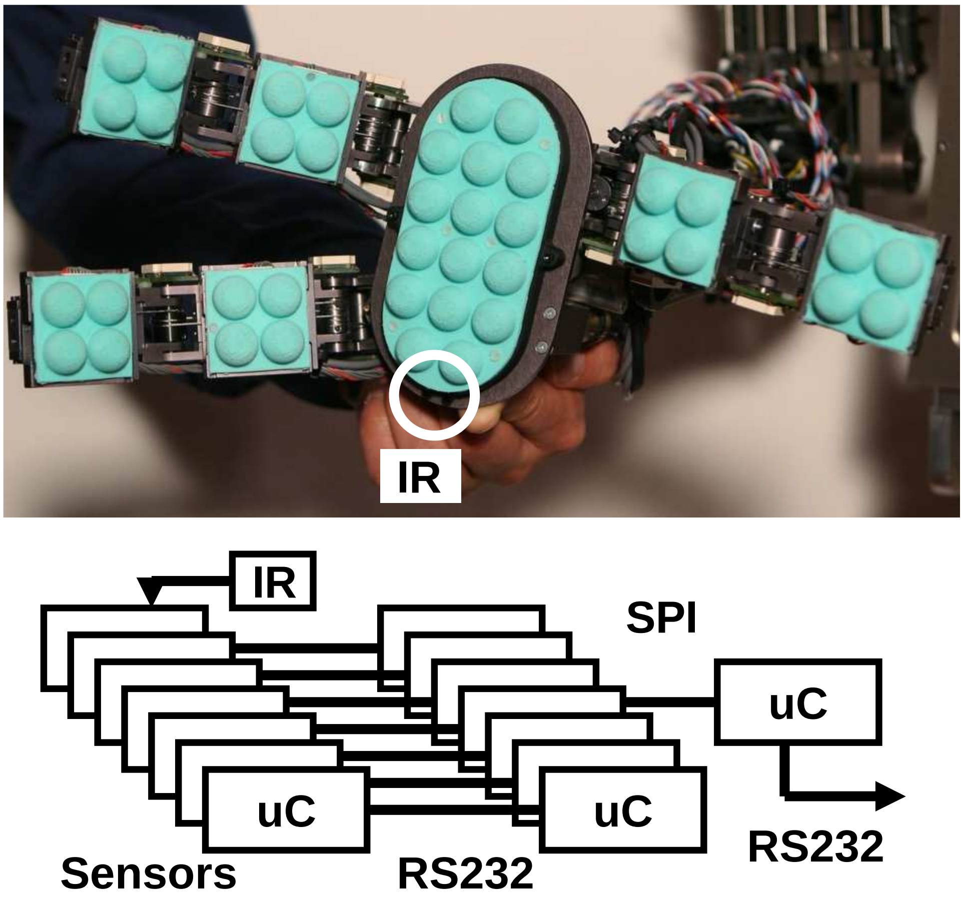

Figure 2-20: Top: Picture of the hand with all its tactile sensors and the infrared(IR) sensor. Bottom: Hardware architecture that networks all the sensors. The RS232 connection goes to a linux box. The microcontroller (uC) used is a PIC 16F876.

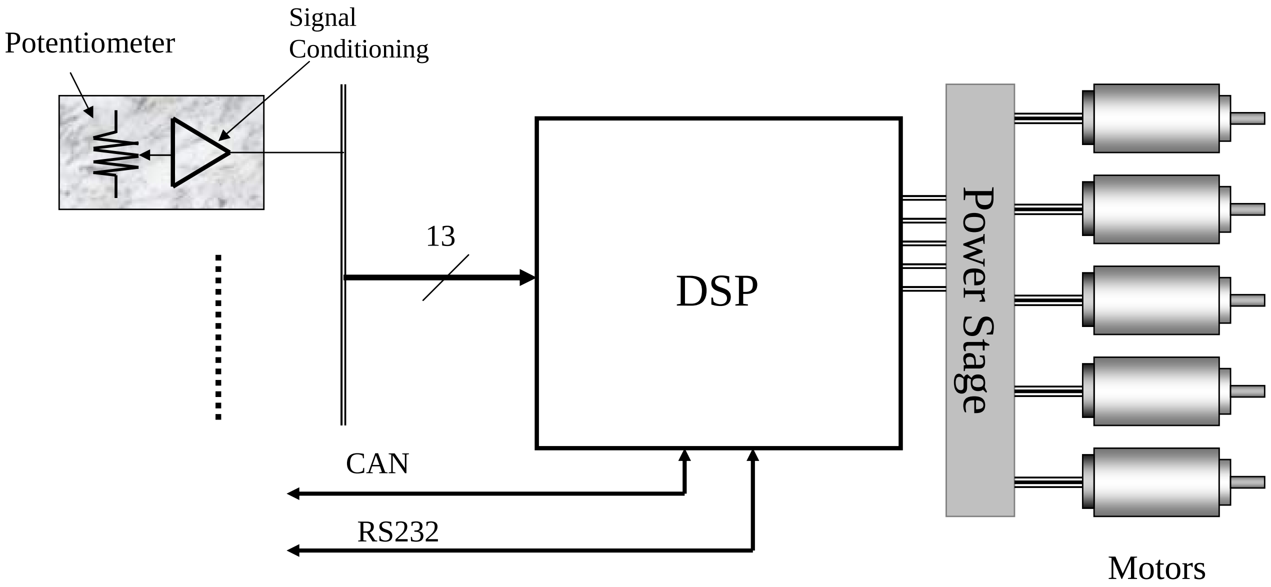

Figure 2-21: Hardware architecture to control the hand. A DSP interfaces with 5 motors, 13 potentiometers (5 for position and 8 for force).In the current implementation the CAN interface is not used. In figure 2-2, we can observe this motor controller as part of the overall architecture.

Hardware Architecture

The hardware architecture for the hand consists of a DSP Motorola 56F807 that reads 13 potentiometers and drives 5 motors. A schematic of this architecture is shown in figure 2-21.

The voltages from each of the potentiometers are amplified and filtered before being connected to the Analog-to-Digital (A/D) converters in the DSP.

The encoder signals from the motors are connected to an LS7266R1 chip which decodes the signal and counts the ticks. These decoders are connected to an 8 bit bus implemented by I/O lines in the DSP.

The five motors are powered by H-bridges that receive direction and PWM signals from the DSP: the direction from I/O ports and the PWM from internal generators. The connections between the DSP and the H-bridges are opto-isolated.

Motor Control

The low level motor control deals with force and position control of the links. Every motor that controls a finger can use the force feedback from either one of the joints or position feedback from the base of the finger. For the rotation of the fingers, the feedback can come from either the position or the force feedback potentiometers. The PWM outputs were calculated using simple PD controllers updated at 500Hz. The setpoints for these controllers come from a higher level controller.

2.2.4 Force controlled arm

The arm used in OBRERO is a copy of the arm created for the robot DOMO [27]. The arm has 6 DOFs: 3 in the shoulder, 1 in the elbow and 2 in the wrist. All the DOF's are force controlled using series elastic actuators.

The motor controller for OBRERO's arm is implemented using a DSP Motorola 56F807 that communicates using a CAN interface as described in section 2.2.1.

2.2.5 Head: Vision and audio platform

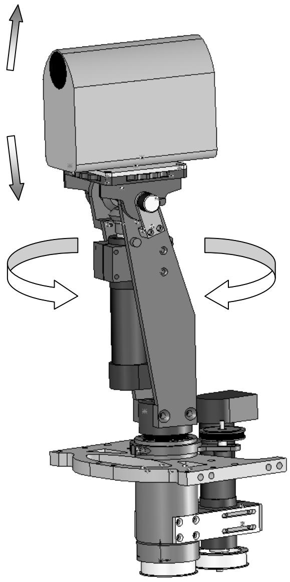

Figure 2-22: Robotic Head. The head has two mechanical DOFs - pan and tilt - and two optical DOF's - zoom and focus

The vision system developed is specialized for manipulation. The system was designed to take advantage of features such as zoom and focus that are not commonly used but are very useful. Focus gives an estimate of depth which is computationally less expensive than stereo, for example. This depth information helps to position the limb. Zoom allows greater detail of an image. For example, we can look very closely at objects to get texture information.

The camera used is a Sony Camcorder model DCR-HC21 which has an optical zoom of 10 times and a resolution of 720 × 480 24 bit pixels. The audio system is integrated in the camcorder and provides 2 channels sampled at 44Khz. The sound and the images are transmitted to a computer using an IEEE 1394 (firewire) cable. The zoom and the focus are controlled using an RS232 port. The RS232 connects to a microcontroller PIC 16F877 that interfaces to the camcorder via LANC (Sony standard).

The camcorder is mounted on a two degree of freedom platform that has pan and tilt as shown in figure 2-22. The head is mounted in the robot torso as shown in figure 2-1. The motors are controlled by a microcontroller PIC 16F877 that communicates using RS232. The architecture of the vision and audio system is shown in figure 2-23.

2.3 Evaluation

We have evaluated the whole robot and each of the parts on different tasks. In this section we present some of these evaluations.

In figure 2-24, we can observe a sequence of pictures of the hand closing on an air balloon by controlling the force exerted by the actuators with no feedback from the tactile sensors. We have used the same control to grab unknown objects. We can observe in figure 2-25 that the hand conforms to different objects. These results show how the task of grasping unknown objects can be simplified by using a force controlled hand.

In figure 2-26 we show a situation in which low mechanical compliance is very

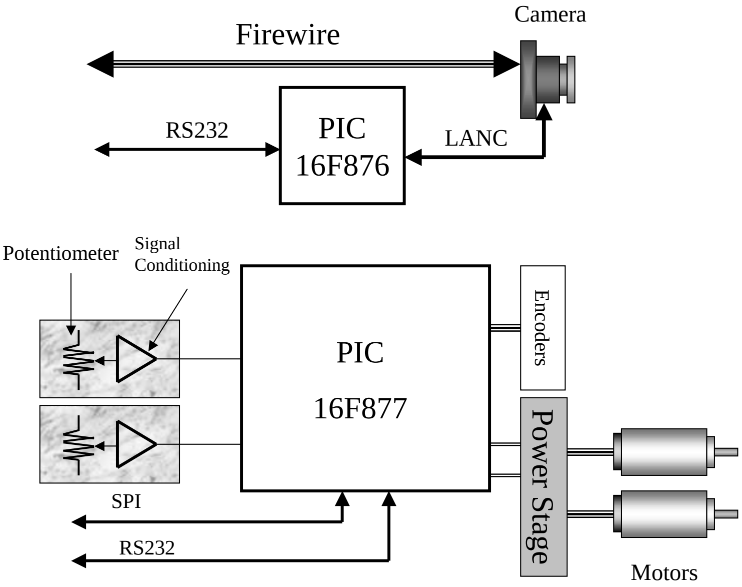

Figure 2-23: Hardware architecture of the robotic head. There are two motors, one for panning and one for tilting. The motors have encoders for precise position feedback and potentiometers to avoid calibration. The SPI communication is not used in this implementation. In figure 2-2, we can observe this motor controller as part of the overall architecture.

useful. OBRERO is blindly moving its limb to explore its environment. When its hand comes in contact with an object, the hand stops without knocking over the object. In this specific case the object is an empty glass bottle. This action is possible due to the low mechanical impedance of OBRERO's fingers. When a finger touches the bottle, it bends and does not knock over the bottle despite the differences in mass and acceleration. The angle that the finger bends is measured by the potentiometer in the actuator and consequently the collision can be detected.

Once the robot comes gently in contact with the object, other features can be extracted. We have used this approach in [99] where the robot taps the object to hear the characteristic sound of its material.

2.4 Summary

In this chapter we have presented the design of OBRERO, a humanoid platform built for addressing sensitive manipulation. The robot consists of a sensitive hand, a force controlled arm, and a vision and audio system.

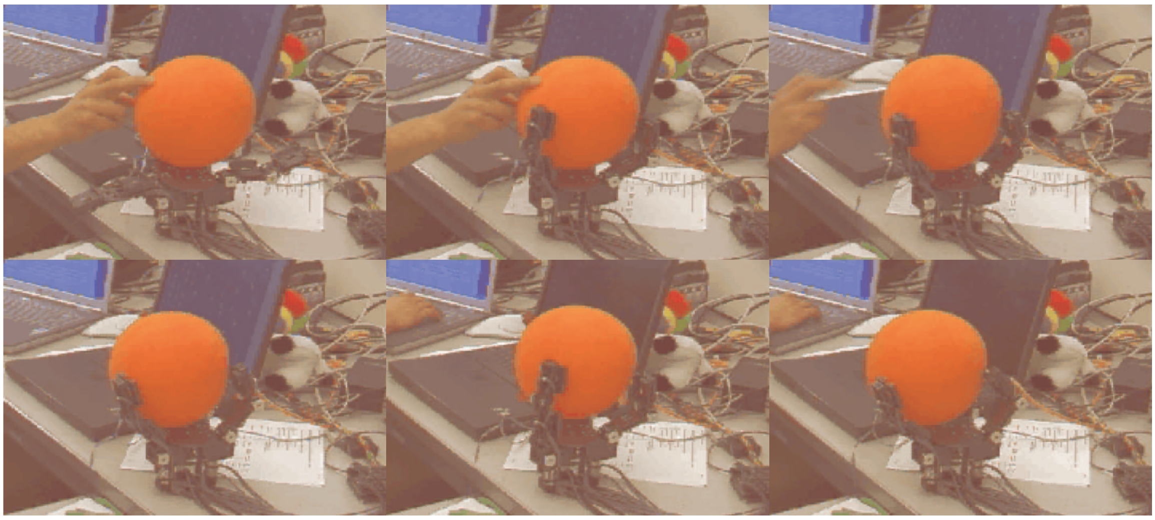

Figure 2-24: Hand closing on an air balloon. The pictures are organized from left to right. On the first two pictures (top-left) we observe the hand closing over an air balloon. When the person's finger is moved, the robotic fingers and the balloon find a position of equilibrium. In the lower row, we observe that the finger in front pushes harder on the air balloon and then returns to its initial position. During that motion the other fingers maintain contact with the balloon.

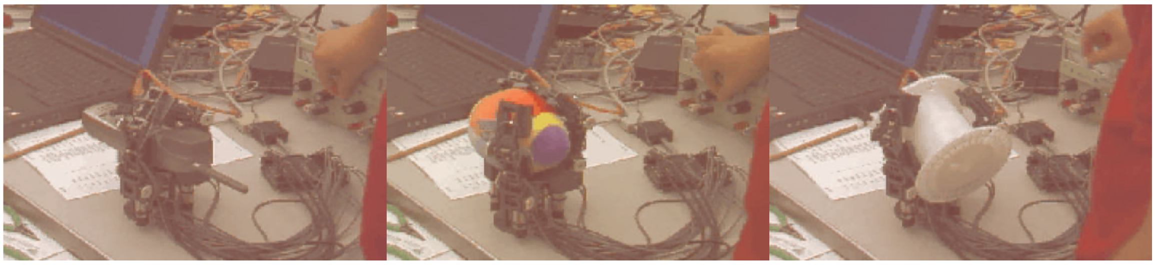

Figure 2-25: Hand closing and conforming to different objects.

The hand and the arm are force controlled and present low mechanical impedance. They are driven by series elastic actuators. New small series elastic actuators have been designed to fit the dimensions of the hand. The hand is covered with newly designed tactile sensors that are presented in chapter 3.

The vision system is intended to be a complement to the sensors in the limb and not the main perceptual input. It consists of a camera with control of zoom and focus. These two optical DOFs are helpful in extracting information from the environment. For example, focus provides depth information while zoom helps to extract small details from an image. Sound is also used to provide additional information for

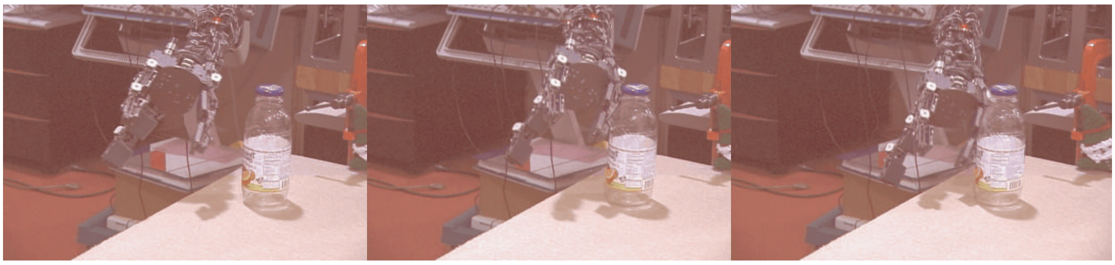

Figure 2-26: When the hand comes in contact with the glass bottle, the finger deflects due to its low mechanical impedance. This deflection allows it to detect the contact. Neither the object nor the hand are damaged.

manipulation.

The features of the robot enable the implementation of behavior-based architecture to perform manipulation tasks. We use this architecture because it has allowed mobile robots to successfully operate in unstructured and dynamic environments. Finally, we have evaluated these features in tasks such as basic grasping, exploring and tapping.

Chapter 3

A soft touch: Compliant Tactile Sensors for Sensitive Manipulation

3.1 Introduction

In this chapter we present the design, analysis and construction of a biologically inspired tactile sensor. We are motivated to build tactile sensors that are useful for robotic manipulation in unmodelled environments [100]. These sensors should be capable of detecting when a robotic part (i.e. finger, hand, arm, etc) comes in contact with any type of object at any incident angle. This feature is very important because in general a robot will not have any prior model of the object and thus have to use its limbs to come into contact and learn about the object. Moreover, the sensor should have a shape that makes it prone to contact.

Current tactile sensors are not good at detecting generalized contact. For example, consider a computer touch pad that uses force resistor sensors (FSR). These pads have high spatial resolution, low minimum detectable the force (about 0.1N) and a good force range (7 bits). These features make the sensor work very well when a human finger, a plastic pen or another object with a pointy shape comes in contact. However if you place a larger object or the same pen at a small incident angle, it is very unlikely that these contacts are detected unless the applied force is very large.

The detection is even more difficult when this pad is mounted in a low mechanical impedance robotic finger (such as the one in OBRERO described in chapter2 and [100]). This is because the low mechanical impedance makes the finger deflect when it comes in contact with an objet. A tactile sensor on this finger should be able to detect this contact with only a little deflection of the finger, because if the deflection is large, undesired forces are already being applied to the object. Therefore, the sensor needs to be very sensitive and able to detect forces applied from different directions.

Another factor that we consider important for the functionality of the sensor is its shape. The shape should be such that it is physically reachable from a range of directions. In other words, the shape should make the sensor prone to contact. We can observe that in the human skin, where there are hairs and ridges that stick out. That is opposed to the design of a traditional tactile sensor whose shape is planar and only normal forces to its surface will be detectable.

Moreover, after the initial contact with an object, the fingers of a robot exert high forces to handle objects. Consequently, the tactile sensors also need to deal with this condition by either handling saturation or having a large operating range. Lastly, a tactile sensor should be able to conform to the object to increase the friction and facilitate manipulation.

In this chapter, we present a biologically inspired tactile sensor which has been designed to deal with the issues described above. Using human skin as inspiration, we designed and built sensors that are prone to contact, can detect contact from different directions and can deform to increase friction. These sensors have been mounted in the fingers of the humanoid robot OBRERO. A short description of human tactile sensing is presented in section 3.2. In section 3.3 we present a survey of the related work. The design and evaluation are presented in sections 3.4 and 3.5. A summary of the chapter is presented in section 3.6.

3.2 Human Tactile Sensing

The human body is completely covered by a sensitive skin which allows us to get information from the environment. The information comes in four modalities [57]:

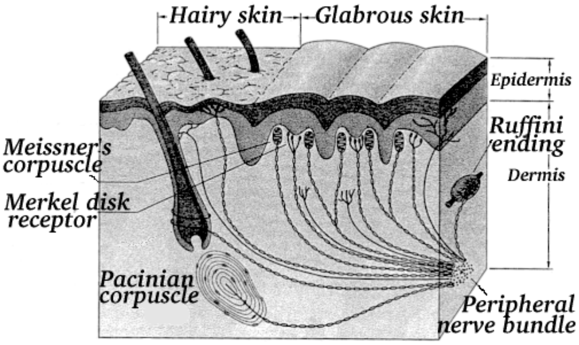

Figure 3-1: Mechanoreceptors present in hairless (glabrous) and hairy human skin.

discriminative touch, proprioception, nociception and temperature.

These modalities supply the following kinds of information:

- Discriminative touch. Shape, size, texture and movement of an object.
- Proprioception. Position and movement of the limbs and body.
- Nociception. Tissue damage.
- Temperature. Perception of cold, cool, warm and hot.

All this information is collected by mechanoreceptors which are different in the hairless and hairy skin as can be observed in figure 3-1. These mechanoreceptors are classified as superficial and deep receptors.

The superficial receptors are known as Meissner's corpuscles and Merkel's disks. Meissner's corpuscles are rapidly adapting sensors that have fine mechanical sensitivity. On the other hand Merkel disks are slowly adapting sensors that measure strain.

The deep receptors are larger than the superficial receptors. They are known as Pacinian corpuscules and Ruffini endings. The Pacinian corpuscules respond to rapid indentations of the skin but not to steady pressure. They also detect vibration. Ruffini endings are slow adapting sensors that link subcutaneous tissue to foldings in the skin in places such as the joints and finger nails.

In the hairy skin there are similar receptors. There are two different types: the hair follicle receptor and the field receptor. They detect hair movement and skin stretches respectively. Both receptors are rapid adapting ones.

The information collected by a group of mechanoreceptors is collected by a dorsal root ganglion neuron. This group of receptors only represents a small area of skin and is known as a receptive field.

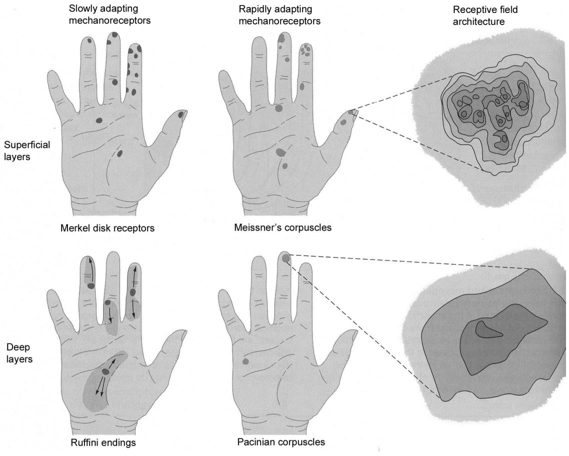

Figure 3-2: We can observe that different mechanoreceptors have different receptive fields. The receptive fields are different in shape and size which make them appropiate for detecting specific features. For instance, Meissner corpuscules detect fine features with circular and small receptive fields. Ruffini endings and Pacinian corpuscules detect coarse features with greater receptive fields.

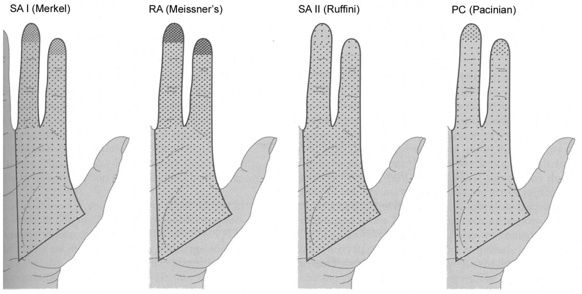

Figure 3-3: Distribution of mechanoreceptors in the human hand. Different mechanoreceptors have different densities.

Receptive fields for different mechanoreceptors are different as we can observe in figure 3-2. For example, the receptive fields of Meissner corpuscules or Merkel's disk are composed of 10-25 mechanoreceptors. In the case of Meissner corpuscules, this corresponds to circular receptive fields with diameters of 2-3mm in the finger's tip and 10mm in the palm. The size of the receptive fields makes these mechanoreceptors ideal for detecting fine features in an object. On the other hand receptive fields of the Ruffini endings and Pacinian corpuscules cover a wider area, which makes them good at detecting coarse features.

All these corpuscules are distributed in different densities in the skin. This gives different spatial resolution in different parts of the body. The hand is one of the richest areas regarding tactile sensing. We can observe the distribution of the mechanoreceptors in the human hand in figure 3-3.

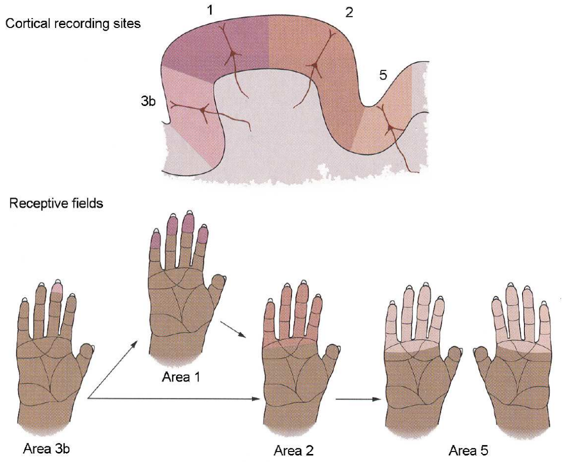

Figure 3-4: We can observe the non-uniform mapping of receptive fields into the sensory cortex.

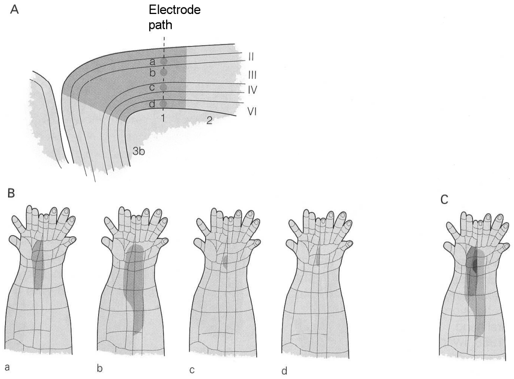

Figure 3-5: The receptive fields in the Brodmann's area are organized in columns that map into skin stripes whose width varies between 300-600 μm.

The information from these mechanoreceptors is sent to the somatosensory cortex. The signals go through several relay regions. When the signals arrive at the cortex, the cortical neurons have larger receptive fields than the ones of the dorsal root ganglion neurons. That allows the detection of more complex features. We can observe how the size of the receptive fields change in figure 3-4.

All the somatosensory inputs that arrive at the cortex from a somatosensory map known as the homunculus. The cortex itself is organized in columns about 300-600 μm wide where each column corresponds to only one location and modality as can be observed in figure 3-5. The space occupied in the homunculus by the different inputs is not proportional to its physical size but to its density of innervation. For example the space used by a finger is greater than the one used by the abdomen as can be observed in figure 3-6. However, the spaces occupied by the different parts of the body are not fixed and they can be changed by experience.

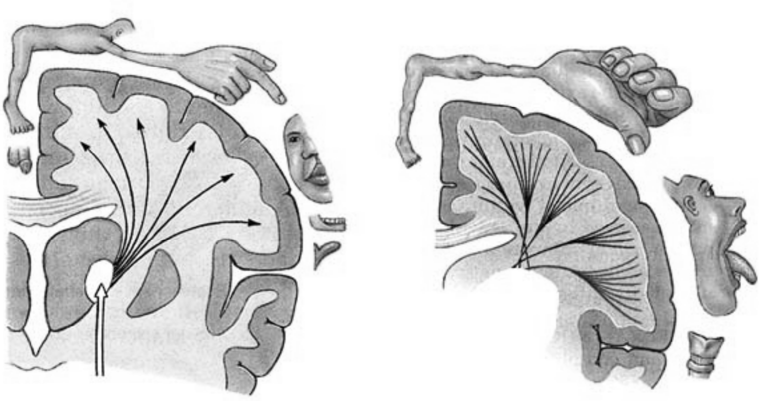

Figure 3-6: Sensory and Motor Homunculus. Each part of the body is represented in proportion to the amount of cortical area dedicated to sensing and controlling a given part. Reprinted without permission from http://www.wirescript.net/magazine/lv9901/lvfig5.jpg

The patterns detected by the mechanosensors are faithfully reproduced in the cortex maps up to the first stage of the cortical map in area 3b [57]. In higher cortical levels, neurons are activated by specific combination of receptive fields. This allows detecting specific features of objects. For example, researchers have identified neurons that are sensitive to: orientation, direction, texture and shape. Sensitivity to orientation and direction is illustrated in figure 3-7.

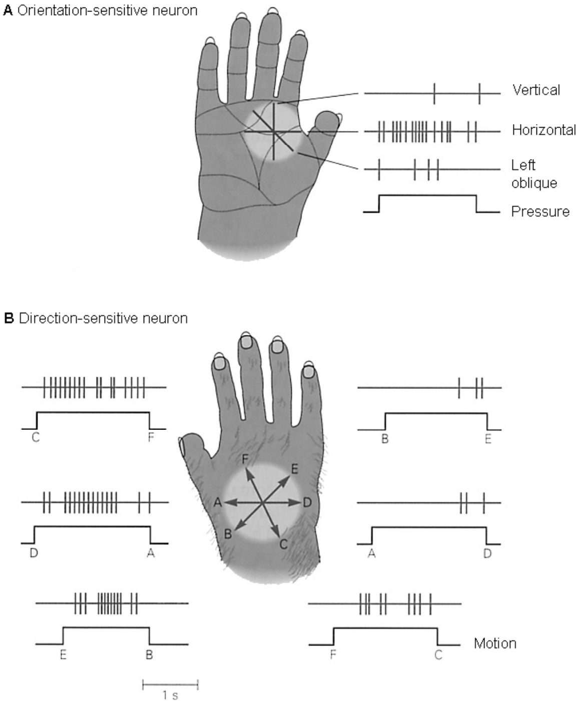

Figure 3-7: Responses of feature-detection neurons. **A.** Orientation sensitive neuron. **B.** Direction sensitive neuron.

3.3 Related Work

Many attempts have been made to implement tactile sensing in robots. There are many technologies used to build sensor arrays: conductive elastomers [93], elastomer-dielectric capacitive [114], optical sensors (surface motion and frustrated internal reflection) [65], piezoelectric [51], acoustic, magnetoelastic, electromagnetic dipoles, silicon micromechanical (mems), and force sensing resistors. A complete review of these technologies can be found in [43]. The performance of these sensors has been measured according to the parameters mentioned in a survey study by Harmon [40]. Those parameters include: spatial and temporal resolution, measurement accuracy, noise rejection, hysteresis, linearity, number of wires, packing, and cost. However, it is not clear if many of these designs are useful for manipulation because little attention has been given to the data produced by these sensors [43].

Most sensors are essentially a flexible elastic skin, coupled with a method of measuring the deformations caused by the applied force. In [39], an IR emitter and receiver are placed inside cavities of an elastic skin. As the skin deforms, the cavity shape changes resulting in change of the received signal. However, the results on pressure sensitivity are not shown, and the sensor is not able to distinguish the direction from which the pressure is applied. In [65] the idea is taken one step further, a full matrix of such sensors is developed and analyzed, light being routed in and out of the cavities using a matrix of optical fibers, taking out some of the bulk of the sensors. Again, only scalar data is obtained.

An interesting idea for a tactile sensor is described by both Lang [66] and Hristuy [25]. Here the finger is a white flexible membrane with a pattern of black dots or lines drawn on the inside. A light source and a CCD camera are placed inside the finger facing the patterns. Pressure on the membrane result in deformations, which are detected by analyzing the CCD images. Significant processing power is required, and overall the sensor is quite big and expensive.

Another approach for sensing tactile forces has been to use joint torque and force information to recover the normal forces [9] instead of using superficial sensors. Nevertheless, this approach is only able to detect resulting forces as opposed to distributions. Tactile sensors also have been developed using organic materials to print circuits [93]. This approach creates flexible transistors that can be used to develop a flexible skin with high density of touch sensors that can easily be wrapped around manipulator's fingers. The sensor used is a rubber sheet that changes conductivity with deformation, the transistors role is to locally amplify the signal and connect it to the matrix of wires that routes the signals to the controller. While the idea has great potential to be developed, so far the results are modest in terms of sensitivity, they only detect force applied in on direction and the organic technology still needs to be perfected (they are sensitive to oxidation).

A promising tactile sensor is described by Chu [114]. Essentially the skin is padded with rubber cones each placed on top of four capacitive force sensors. Having four sensors under each cone makes the detection of both the perpendicular as well as the sideways force the possible. The results were very encouraging with sub gram forces detected. However the process requires a custom silicon wafer to be made (for the capacitive sensors) and therefore might be prohibitively expensive.

A unidirectional capacitive sensors is described by Voyles [55]. Here an electrorheological gel is placed between the inner and outer membrane padded with electrical contacts. The inner membrane is a hard material (like plastic) while the outer material is elastic (rubber). Again, pressure changes the relative position of the contacts which causes a change of the capacitance. One interesting property of this fluids is the change in viscosity with the electric field applied, opening doors for an adaptive skin strength to the task. Unfortunately no quantitative data on the sensor are published.

Most recently (after this work was implemented), Maheshwari and Saraf [71] have implemented a tactile sensor with very high spatial resolution. The resolution is high enough to recover the indentations on a coin when pressed against the sensor. They build a thin-film (100-nm-thick) device composed of metal and semiconducting nanoparticles. This film has the property that its current density and its electroluminescent light intensity are proportional to the pressure applied. The technology is still experimental and a bit complicated since it requires a camera and a lens to work. However, it suffers from the some of the problems described earlier: it is flat, it has only been tested with normal forces, and it has not been tested in an actual gripper.

3.4 Analysis and Design

In designing the sensor we considered the lessons learned in the robot OBRERO. OBRERO's hand originally used commercial FSR's pads of high spatial resolution (1000 dot per square inch), however, these sensors presented the problems described in section 3.1: flat shape, rigidness and low friction. Moreover, the tactile sensors did not provide any information when they came in contact with a given object in general. Even though the minimum force to produce a reading is $0.098N$ (as stated in [49]), this only works when applied in a particular direction and with a particular probe (i.e. a finger).

In order to address these problems, we analyzed the human skin as a source of inspiration. In figure 3-8 a sectional cut of a the skin is presented. We can observe,

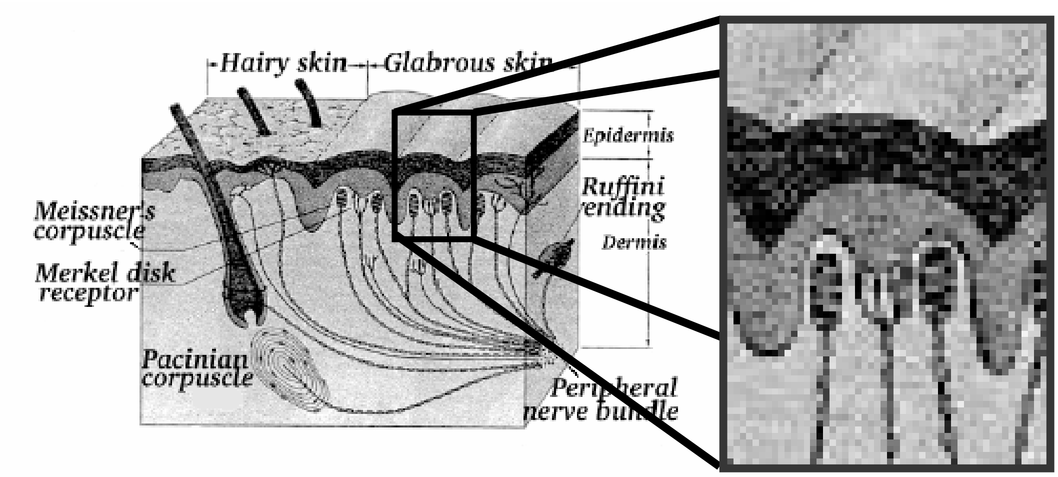

Figure 3-8: Detail of the skin. In the magnified view we can observe a curved shape and the lateral sensors that detect its deformation.The innervations on the sides detect fast changes (oscillations) while the ones on the center detect slow changes (constant pressure).

in the amplified view, that the skin shape is curved. This curve deforms when it comes in contact with an object; the innervations on the sides of the shape detect fast deformations which are correlated with the changes on the force applied. The deformation also increases the surface of contact between the skin and the object, therefore increasing the friction.

Based on this mechanism, we build dome-like shapes using silicon rubber and measure magnitudes related to the deformation to determine the forces applied. We started by analyzing the behavior of the dome in simulation.

3.4.1 Simulation analysis

In order to determine the relation between the load applied to the dome and its deformation, a model to do finite element analysis (FEA) was implemented in ADINA 8 (See figure 3-9a). The specifications of the dome are shown in figure 3-9b. The dome has a flat surface on top and a circular profile on the sides. Under the flat surface there is a magnet that it will be used for detecting the deformation. The magnet was modelled as a rigid object.

Two types of forces acting on the dome were evaluated: a normal force acting

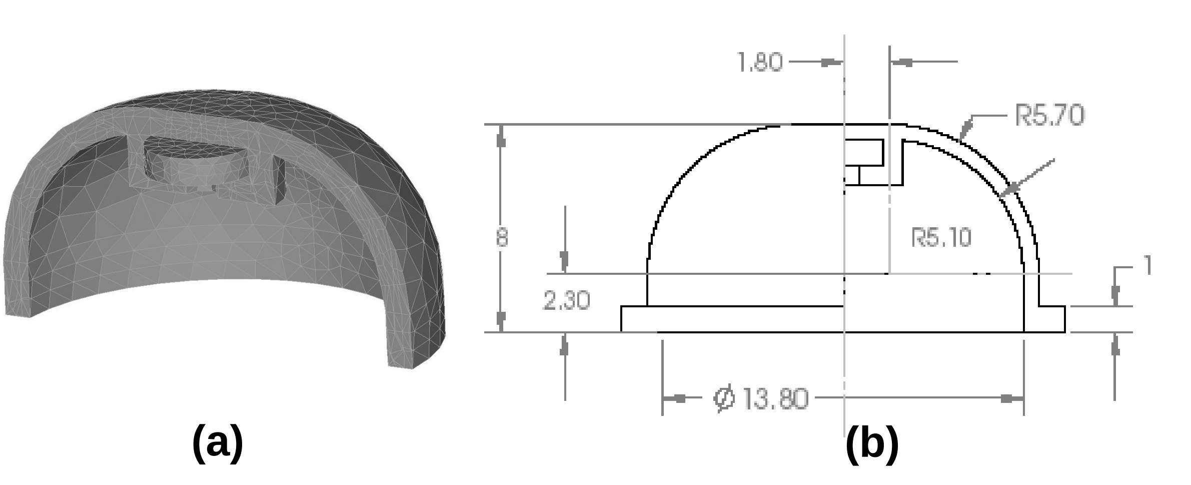

Figure 3-9: (a) Meshed geometry for the FEA. (b) Dimensions in mm of the rubber dome.

on the top of the dome and a normal force to the surface acting at 45 degrees away from the top of the dome. The base of the dome was fixed. The sensor material was assumed to be isotropic. The stress and strain constitutive relation was assumed to be linear due to the small loads applied to the dome. The values for the material properties were modulus of Young $1.5 \times 10^6 Pa$, Poisson ratio 0.45, and density of $1100 kg/m^3$. For the first load case, normal force applied on the top and the dome downwards displacement as a function of the applied load was determined. The maximum dome displacement is constrained by the base of the sensor which is at $5mm$ from the top of the dome. It was determined that the maximum displacement is achieved under a load of $0.147N$. The resulting dome displacement due to a load is showed in figure 3-10.

Similarly, for the case where the force acts 45 degrees away from the top of the dome, the relations of the horizontal displacement vs. load and the vertical displacement vs. load were determined. The horizontal displacement is measured from the center of the dome. The vertical displacement is measured from the top of the dome (when no load is applied) downwards. The displacements are normalized by the maximum horizontal displacement of $7.5mm$ and the maximum vertical displacement of $5mm$. The load is normalized by the maximum vertical load of $0.147N$. Figure 3-11 illustrates the deformed geometry and the load/displacement relations.

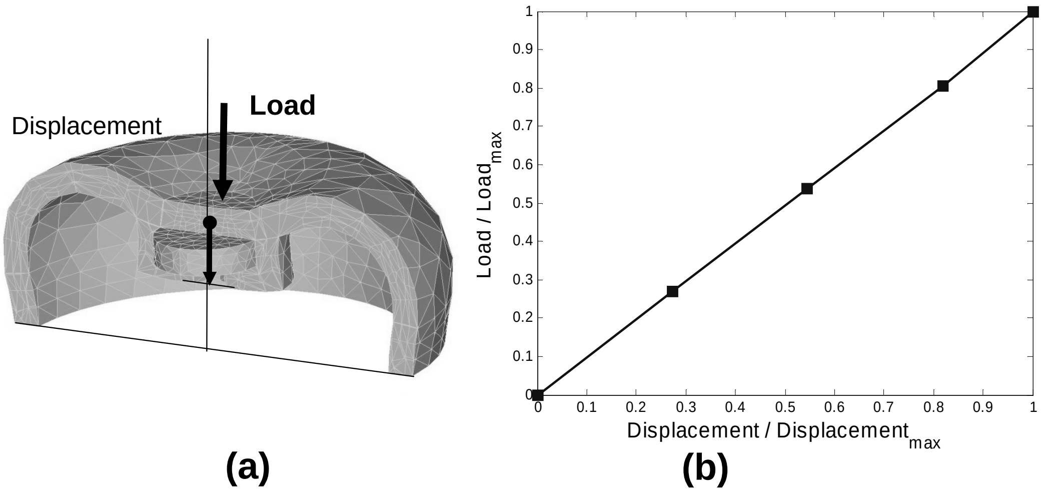

Figure 3-10: (a) Deformed mesh under normal force acting on top of the dome. (b) Load versus displacement when force acts on top of the dome. The displacement of top of the dome is measured from its initial position (no load) to its final position (equilibrium with the load applied). The load is applied to the center of the magnet holder. The displacement is normalized by the maximum dome displacement of $5mm$ and the load is normalized by the maximum load of $0.147N$.

The FEA shows that there is a linear relation between the displacement and load and that the dome can detect forces acting at various locations of its surface. Parameters that have not been explored are the dome thickness and curvature; these parameters should allow tuning of the sensor to detect a force.

3.4.2 Design

The dome shape shown in figures 3-9 and 3-12 was used as the basic shape for the sensors. This dome deforms when either lateral or normal forces are applied as shown in the section 3.4.1. The force needed to deform the dome is determined by the structure and the material used to build the sensor. According to the FEA, there is a linear relation between the displacement and the load. Given that the FEA does not cover all the possibilities, this linear relation might not hold for all the forces applied. However, we can assume that measuring the displacement of the center of the dome will give us a reasonable estimation of the forces applied.

In the design process, we considered a number of methods for determining the displacement of the center of the dome, i.e. using magnetic, optical, and pressure

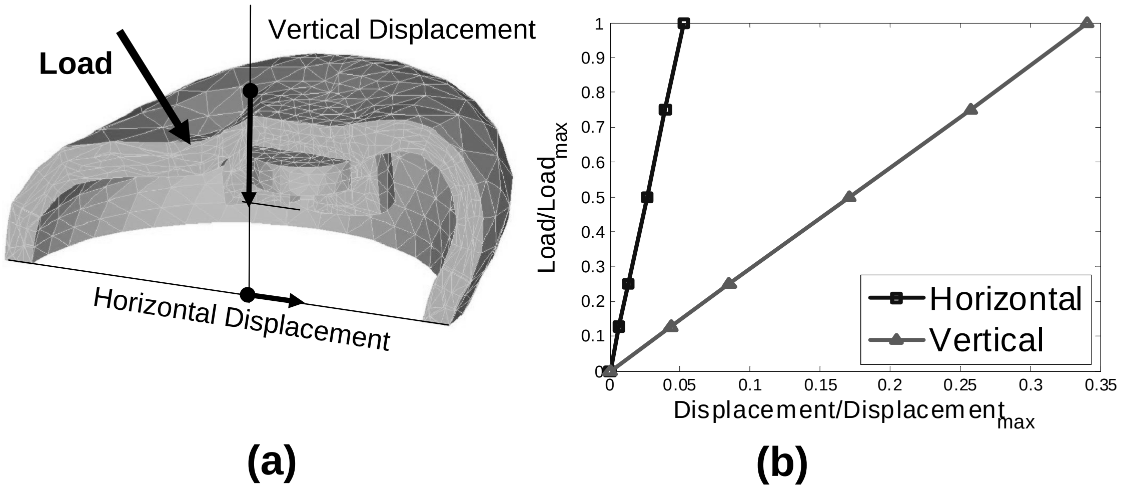

Figure 3-11: (a) Deformed geometry under a load acting normal to the surface 45 degrees away from the top of the dome. (b) Load versus displacements. The horizontal displacement is measured from the center of the dome. The vertical displacement is measured at the top of the dome as in the previous case. The displacements are normalized by the maximum horizontal displacement of $7.5mm$ and the maximum vertical displacement of $5mm$. The load value on the plot corresponds to the total force applied. The load is normalized by the maximum vertical load of $0.147N$

sensors. We chose to use the magnetic approach in our primary design and evaluation. We also implemented the optical method and showed some preliminary evaluation in section 3.5. Lastly, we implemented and discussed the pressure method. In the following sections, we describe each method.

Magnetic

In this approach, we embed a magnet in the top of the dome. The base of the dome is glued to a printed circuit board (PCB). The magnetic field of the magnet is detected by 4 hall effect sensors on the PCB (figure 3-12b). The difference on the signal detected by the sensors will give a rough approximation of lateral displacement and the average of the signals will give the vertical displacement. When the top of the dome reaches the base of the dome, no more displacement will be allowed and the sensor saturates. Additional sensors can be placed in the PCB to measure beyond this saturation limit. Each one of the hall effect sensors is multiplexed and read through and A/D converter. This converter is part of 16F876A PIC microcontroller that transmits the information via RS232 using ASCII code. Four of these domes

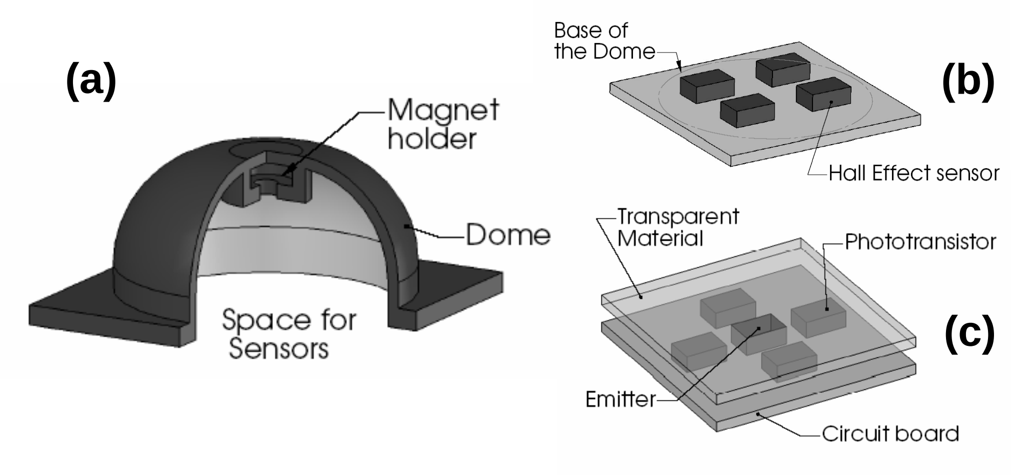

Figure 3-12: (a) Cut of the silicon rubber dome. The base of the dome is glued to a circuit board with electronics that measures the deformation. (b) Diagram of the circuit board using hall effect sensors to measure the deformation. This board goes under the dome. (c) Diagram of the circuit board using light to measure the deformation. For this case a dome without the magnet holder was built.

were used in each phalange of the fingers in the robot OBRERO (see figure 3-18). OBRERO's hand has 24 domes in its fingers and 16 in its palm.

It is important to note that the sensitivity of the sensor can be controlled by changing the shape of the domes. More easily deformable structures can be created to satisfy a given application. In figure 3-13, we present an example of a more sensible sensor that consists of only one dome as opposed to four as in the previous case. There are four magnets embedded in the structure which allows detect the deformation. These structure is easily deformable because a larger surface is supported by farther away walls. However, independently of the shape, the problem with this sensing method is that it is affected by metallic objects. Therefore we also experimented using light and pressure for detecting the deformation.

Optic

The dome used for optical detection is like the one shown in figure 3-12 except that the magnet holder was removed. The base of the dome is glued to a printed circuit board (PCB) which has one light emitter and 4 phototransistors as shown in figure 3-12c. The total power reflected gives the vertical deformation of the dome and the

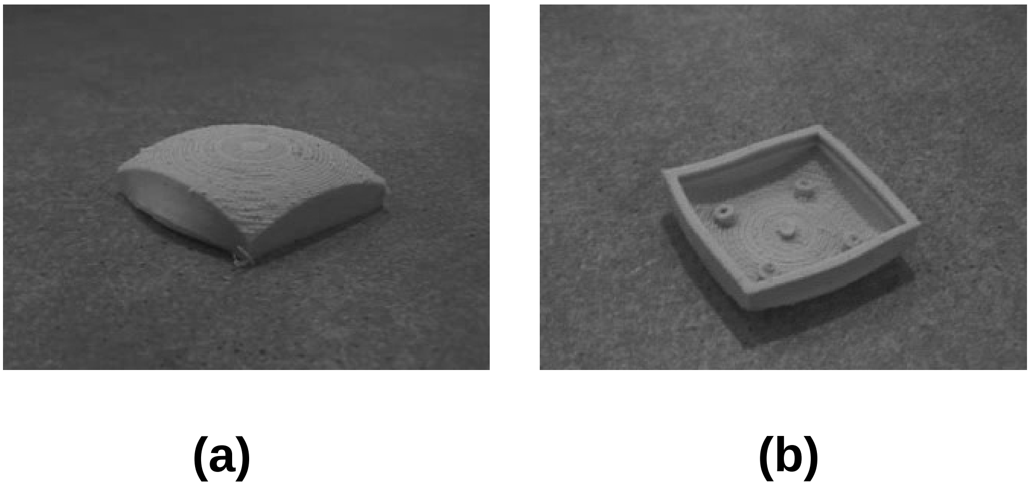

Figure 3-13: (a) The shape of this dome is a spherical surface intersected with a cube. This surface deforms with less force than the previous shape considered. (b) We can observe the magnets on the other side of the dome.

difference between the sensors signals gives the lateral deformation. A transparent layer was added to prevent the dome from completely blocking the light emitter and guarantee a monotonic reading of the light reflection. This layer will also be the limit of the vertical displacement of the dome. The force that produces such deformation is the saturation threshold of the sensor.Therefore, this layer will deal with forces greater than the saturation limit. As in the previous case, the output of each of the phototransistors goes to a multiplexor connected to an A/D converter which is part of a 16F876A PIC microcontroller.

Pressure

Another implementation was to seal the domes with air in their interior and place a pressure sensor on their base. The pressure was measured using a ICS1451 sensor. The drawback of this approach is that it does not differentiate horizontal from vertical displacement. Moreover, pressure sensors need more complicated and more expensive electronics than the previous methods. Therefore, we did not completely characterize this type of sensor.

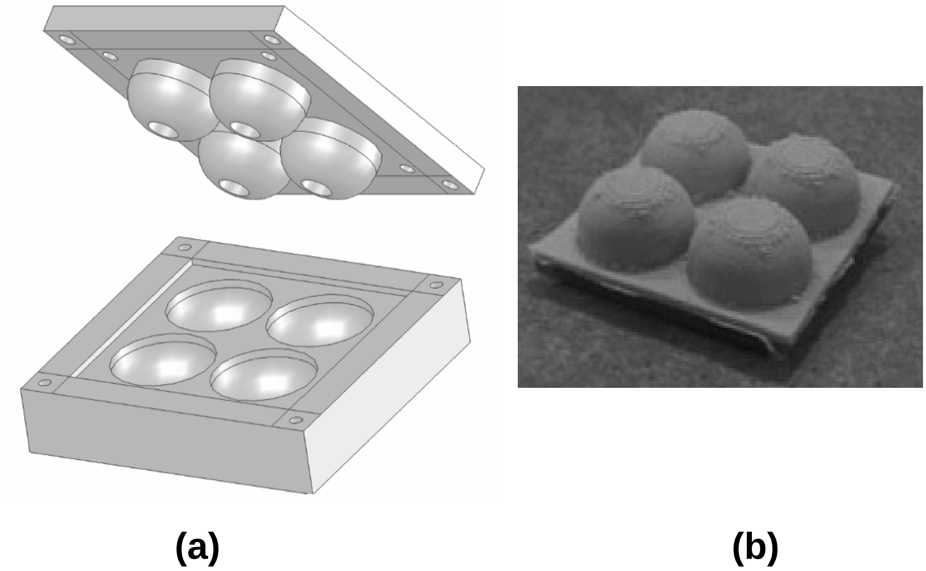

Figure 3-14: (a) CAD rendering of the mold used to build 4 sensors. The dimensions were chosen to fit the fingers of the robot OBRERO. (b) Four domes molded using silicon rubber.

3.4.3 Material and molding

The sensors were built using Freeman V-1062 room-temperature, condensation-cured silicone, which is very simple to use without expensive tooling. This material was chosen because of its very low hardness (14 Shore A), good tensile strength (545 PSI) and tear resistance (120 ppi). The molds were constructed using a 3D printer. In figure 3-14 we can see one of those molds.

3.5 Evaluation

As mentioned above, we implemented and evaluated two approaches for determining the displacement of the center of the dome, i.e. using magnetic and optical sensors. In this section, we will first present the experimental results of our primary design using the magnetic approach. We then show some preliminary results of the sensors using optical sensors.

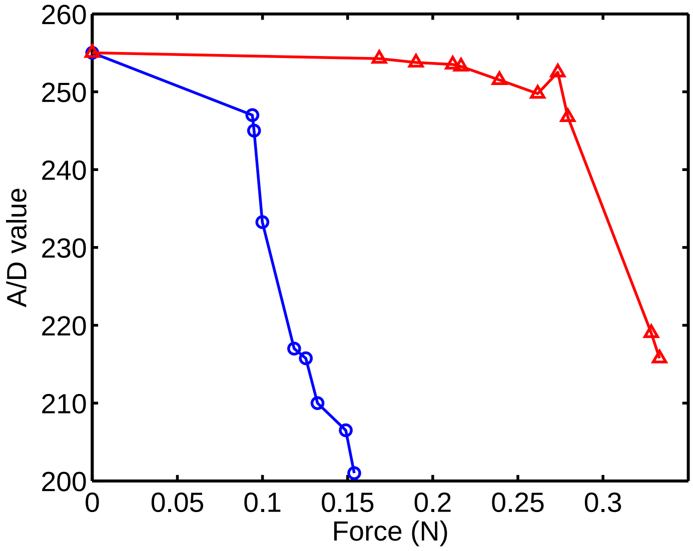

Figure 3-15: Response of the magnetic sensor to a vertical force. The figure shows the average of the readings from 4 hall effect sensors as the force applied changes.Each hall effect sensor is read by a 8 bit A/D converter. The dome is composed of spherical part with a flat circular surface on the top (figure 3-9b). The behavior of the dome is different depending on the relative size of the object in contact with respect to the circular surface. The lines with circular and triangular marks show, respectively, the responses to probes smaller and larger than the circular surface of the dome.

Magnetic

We apply normal and lateral forces and record the displacement of the center of the dome. The displacement is determined by measuring the intensity of the magnetic field that reaches the hall effect sensors in the base of the dome. The configuration of the hall effect sensors are shown in figure 3-12b.

We first evaluated the response of the sensor to normal forces. Different weights were applied to the top of the dome using plastic holders. Two cases were considered; holders with contact surfaces smaller and larger than the top surface of the dome. Figure 3-15 shows the average of the reading of 4 hall effect sensors as the normal forces applied change. The reading of each hall effect sensor is obtained using a 8 bit A/D converter. The lines with circular and triangular marks show, respectively, the responses to weights whose contact surface are smaller and larger than the circular surface of the dome.

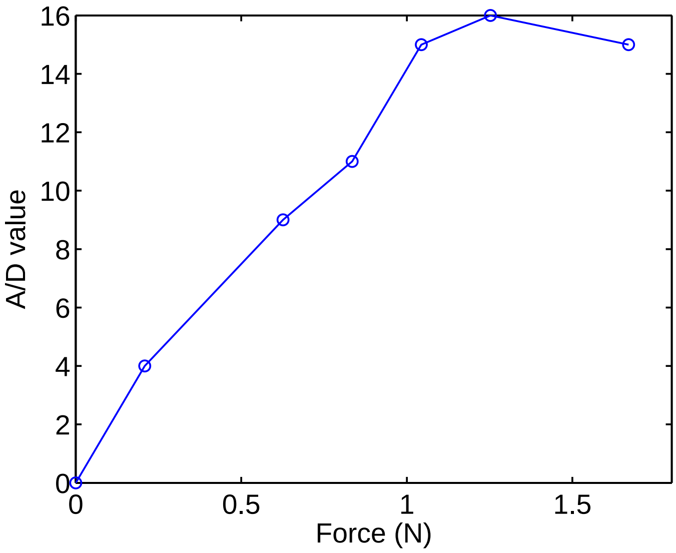

Figure 3-16: Response of the magnetic sensor to lateral forces. As shown in figure 3-12b, the hall effects are organized in a 2 by 2 array. The data presented in this figure is the difference of the two columns. The value of each column is the summation of the readings in that column. The readings come from a 8 bit A/D converter that digitalizes the multiplexed sensor outputs. The direction of the force applied was normal to the columns of hall effects.

We observe that the readings decrease monotonically in each of the two cases. The relation is not linear as shown in the simulation because the relation between the intensity of the magnetic field and the distance to the hall effects is not linear.

These two different behaviors are determined by the shape of the dome which is composed of a spherical part with a flat circular surface on the top. If the sensor shape were a semi-sphere it would be more difficult to deform and if it were a plane it will be easier. The current one is a combination of both of them. These shape makes the deformation of the sensor greater when it comes in contact with pointy objects. In the simulation analysis, only the case with a contact surface smaller than the top of the dome was considered since we wanted to have a general idea of the behavior of the structure.

In order to determine the response to lateral forces, an array of four domes (see figure 3-14b) were placed upside down against a flat surface and weight was applied on top of them. These provide a minimum normal force that remained constant. The

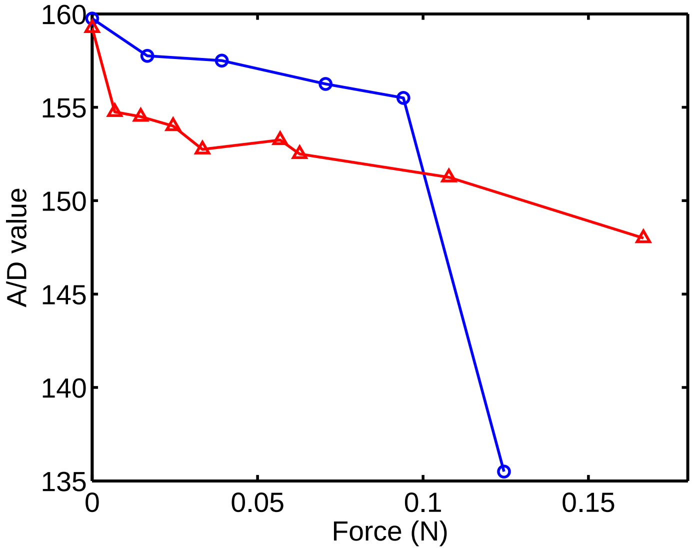

Figure 3-17: Response of the optical sensor to a vertical force. The figure shows the average of the readings from 4 photo transistors as the force applied changes. Each phototransistor effect sensor is read by a 8 bit A/D converter. The dome is composed of spherical part with a flat circular surface on the top (figure 3-9b with no magnetic holder). The behavior of the dome is different depending on the relative size of the object in contact with respect to the circular surface. The lines with circular and triangular marks show, respectively, the responses to probes smaller and larger than the circular surface of the dome.

motion of the array was restricted to one direction by placing fixed plates on two opposed sides of the array. A lateral force was applied by a cable attached to the side of the array of 4 domes in the unrestricted direction. The cable was routed through a pulley and attached to a weight. The weight applied determines the pulling force.

The lateral displacement is determined by computing the difference among the readings of the hall effect sensors. The sensors whose output is greater have the magnet closer to them. Figure 3-16 shows the computed difference between the readings of two groups of two sensors for a lateral force of known direction. We can observe that the response curve is roughly monotonic. The minimum normal force detected by the magnetic sensor is $0.094N$. The maximum deflection is obtained when a force of $0.147N$ is applied to the center of the circle in top of the dome and the weight was smaller than the dome.

Certainly many tactile sensors have good sensitivity but they may not necessarily

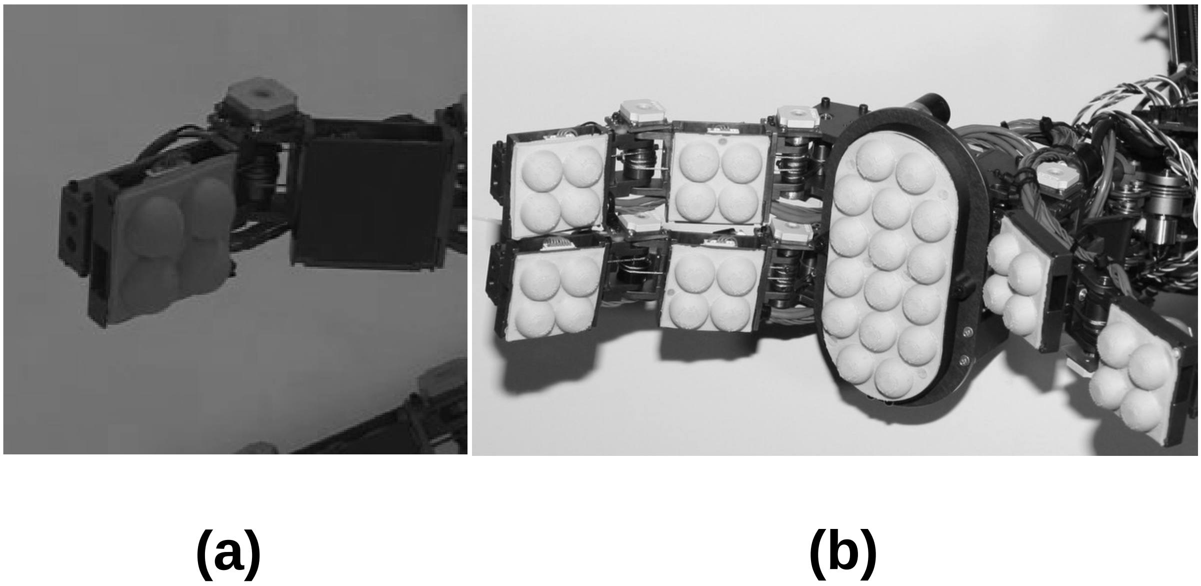

Figure 3-18: (a) A four dome tactile sensor mounted in the finger tip of the robot OBRERO. (b) Hand of the robot OBRERO with 8 domes mounted in each of its 3 fingers and 16 on its palm.

work well on a real robot. These sensors were mounted on the fingers of OBRERO's hand (see figure3-18). In figure 3-19 we show a sequence of the robot approaching a cylindrical object whose physical characteristics are: $mass = 0.179Kg$, $diameter = 92mm$, $height = 216mm$. When OBRERO performs a blind (no vision involved) exploration task and touches the object, it is not pushed or tilted. The tactile sensors deform and conform to the surface of the object. Consequently the hand touches the object gently. Regretly, it is difficult to see all these effects in the figure.

Optic

As mentioned above, the second method used to measure the deformation was optic. A LED and 4 phototransistors were placed in the base of the dome (figure 3-12c). The same two cases as in the magnetic method were considered. Figure 3-17 shows the results of this evaluation. The values on the vertical axis are the average of the readings from 4 photo transistors. As in the magnetic case, the lines with circular and triangular marks show, respectively, the responses to weights smaller and larger than the circular surface of the dome. The response to lateral forces was not evaluated for this version.

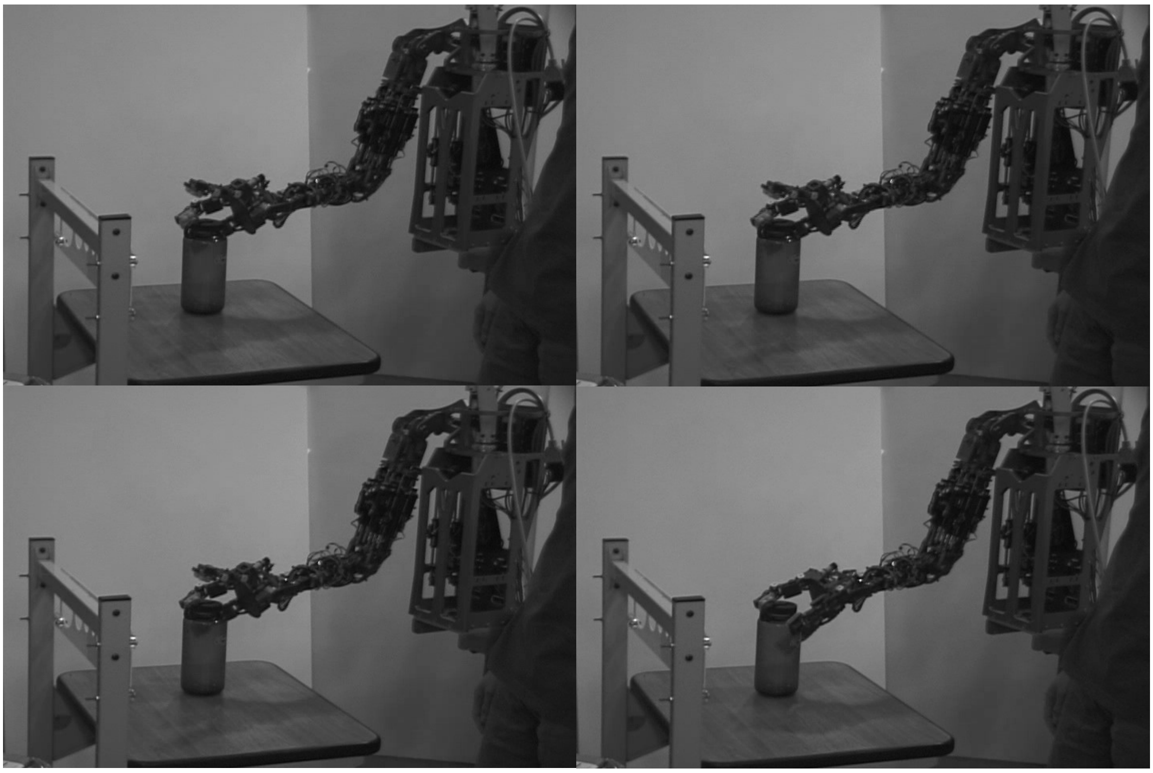

Figure 3-19: From left to right and from top to bottom, we present the robot OBRERO moving its hand until a contact with a bottle occurs. The task is performed with no help from vision. In the top-left picture, the hand is approaching to the bottle. In the top-right picture, the sensor touches the bottle. In the bottom-left picture, the sensor is being deformed further.In the last picture, the thumb of the robot is moved to the side as a response to the touch. We should notice that the plastic bottle has little mass and it has not been tilted by the contact with the hand.

3.6 Summary

We have presented in this chapter the design, analysis, constructions and experimental evaluation of a biologically-inspired tactile sensor. Our sensor has five key properties that make it better suited for manipulation than previous designs. It conforms to the surfaces with which it comes in contact by elastic deformation, enabling the grabbing of a wider range of objects. Its shape make it prone to contact which create the opportunity to sense the environment. It is sensitive to both normal and lateral forces, providing better feedback to the host robot about the object to be grabbed. It has a high sensitivity, enabling its use in state of the art manipulation fingers, which typically have low mechanical impedance - in order to be very compliant. We showed in this chapter the integration of our sensor with OBRERO's hand [100]. Last but not least the construction of the sensor is simple, using inexpensive technologies like silicon rubber molding and standard stock electronics.

While some of these feature are present in previous designs, none of previous sensors has encompassed all of them. Additionally most of the previous sensor are not fully characterized experimentally, and therefore estimating there usefulness in robotic manipulation is difficult.

Chapter 4

Observations on humans

4.1 Introduction

Experimental results suggest that from a very early age, arm movements in infants are influenced by vision. For example, van der Meer and colleagues found that sight of the hand allows infants to maintain the posture of the hand when pulled by an external force [103]. Von Hofsten compared the arm movements of two groups of infants in the presence and absence of an object and found that in the former case arm movements were significantly more frequent. When the infants were fixating the objects the movements were directed closer to them [105].

Taken together, these results suggest that in children some sort of eye-hand coordination is already present soon after birth. But on the other hand, continuous visual feedback from the hand is not required for infants to reach for an object [20, 21]. Indeed, it is only at 9 months of age that children seem to be able to exploit visual feedback from the hand during the approach phase [5]. A possible explanation for this could be that in the first months of development the visual system of infants is still rather immature: visual acuity is limited and perception of depth has not developed yet [15]. Later on during development the role of vision is certainly crucial to control the correct preshape of the hand according to the object's shape and orientation; however, tactile feedback from the contact with an object is an alternative source of information that could initially substitute for the visual feedback.

In adults, several studies have revealed the importance of somatosensory input for manipulation; for example human subjects with anesthetized fingertips have difficulty in handling small objects even with full vision [52]. Humans use a set of strategies collectively called *exploratory procedures* [67] in their perception of the world around them, such as tracing object outlines with a finger.

Motivated by these studies about exploration and tactile sensing, we look at human manipulation as inspiration for our work. In this chapter we present an experiment were we closely observed people performing tasks related to what our robot will face.

4.2 Simulating our robot with humans

Human haptic perception is impressive, even under serious constraints. In [68] we can find a review of different experiments done with humans to determine how well the subjects can identify objects using only haptic information. In the experiments mentioned, the individuals wore headphones and a blindfold to make sure that sound and vision did not provide extra information about the objects. Haptic information was also systematically interfered with to explore different aspects of manual exploration. The constraints included: reduced number of end effectors, compliant covering, application of rigid finger splints, rigid finger sheathes, and rigid probes. These constraints reduced either one or many aspects of the cutaneous (spatial, temporal and thermal) and kinesthetic information available to the subjects.

The results showed that by reducing the type of sensing available in the human hand, the subject's recognition performance is reduced. The lowest recognition accuracy for objects was around 40% when the subjects used a probe to explore the object. This recognition task took around 80 seconds. For the researchers who did this work, these numbers may seem low – but for a robotics researcher, they are a cause of envy, and show that human haptic perception is indeed very impressive even under unusually-constrained situations.

To get an "upper bound" of what we could expect from our robot, we evaluated the performance of human subjects when wearing thick gloves that reduced their sensitivity and dexterity to an extent that would be comparable to our robot. We blocked their vision, since we know our robot cannot compete with human visual perception, but let them hear.

We sat 10 subjects in front of a padded desk covered with various objects – a wooden statue, a bottle, a kitchen glove, a plastic box, a paper cup, a desktop phone, a tea bag and a business card. (See figure 4-1);

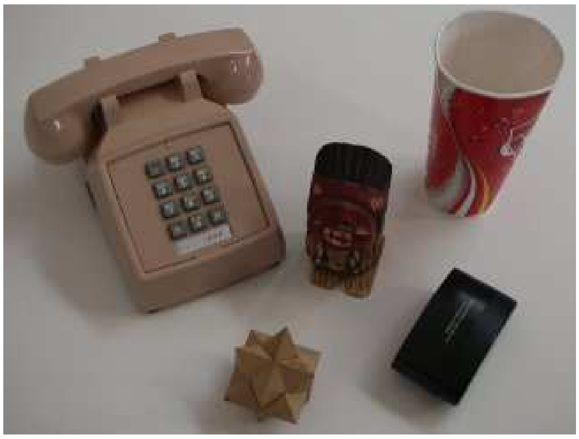

Figure 4-1: Some of the objects used in our experiment. The wooden statue, the wooden star and the black oval box are not considered common objects. These objects were not correctly identified.

The subjects wore a blindfold and a thick glove which reduced their haptic sensitivity and the number of usable fingers. The glove only allowed them to use their thumb, their index and middle finger.

A goal of the experiment was to determine how much and in what way humans can manipulate unknown objects in an unknown environment with capabilities reduced to something approximating our robot (described in Chapter 2).

Our subjects were instructed to perform certain tasks starting from a constant initial position, sitting straight with their right arm relaxed and close to their waist. The first task was to find and (if possible) identify objects on a desk. This task was repeated with multiple set of objects. When changing from one set of objects to another, the subjects were moved away and turned around so that their back was facing the desk. Some examples can be observed on figures 4-2, 4-3, 4-4, 4-5, and 4-6.

The next task extended the challenge further. Along with locating and identifying the objects (an arbitrary name was assigned when an object was not recognized), the subjects were instructed to remember the object's position. Later, they were instructed to move their hand to a named object starting from the initial position (see figures 4-7 and 4-8).

For the final task, a few objects and a desktop phone were placed on the desk. The handset and the phone base were disconnected – the phone cord was removed, and the two parts of the phone were placed in separate locations. The subjects initially had no idea a phone was present. They were instructed to find, identify and remember the position of the object on the desk. If they identified the two parts of the phone, they were instructed to grab the handset and place it in the correct position on the phone base. An instance of this task can be observed in figure 4-9.

In the following section, we summarize some of our observations from these experiments.

4.3 Observations

4.3.1 Examples

In this section, we present a few examples that illustrate some of the skills that people use to manipulate objects.

Exploration

In figure 4-2, we observe a sequence of frames showing a subject grabbing a plastic oval box. It should be noted that she has had no previous experience with this specific box in this experiment. In frame 1, the initial contact with the plastic box is only done with the exterior side of the thumb. After this contact, the person is able to reposition her hand over the object. Then, she further explores the object by moving her hand around and closing it over the object (frame 4). This seems a good position for the hand to lift the object. However, we observed that she reorients her hand

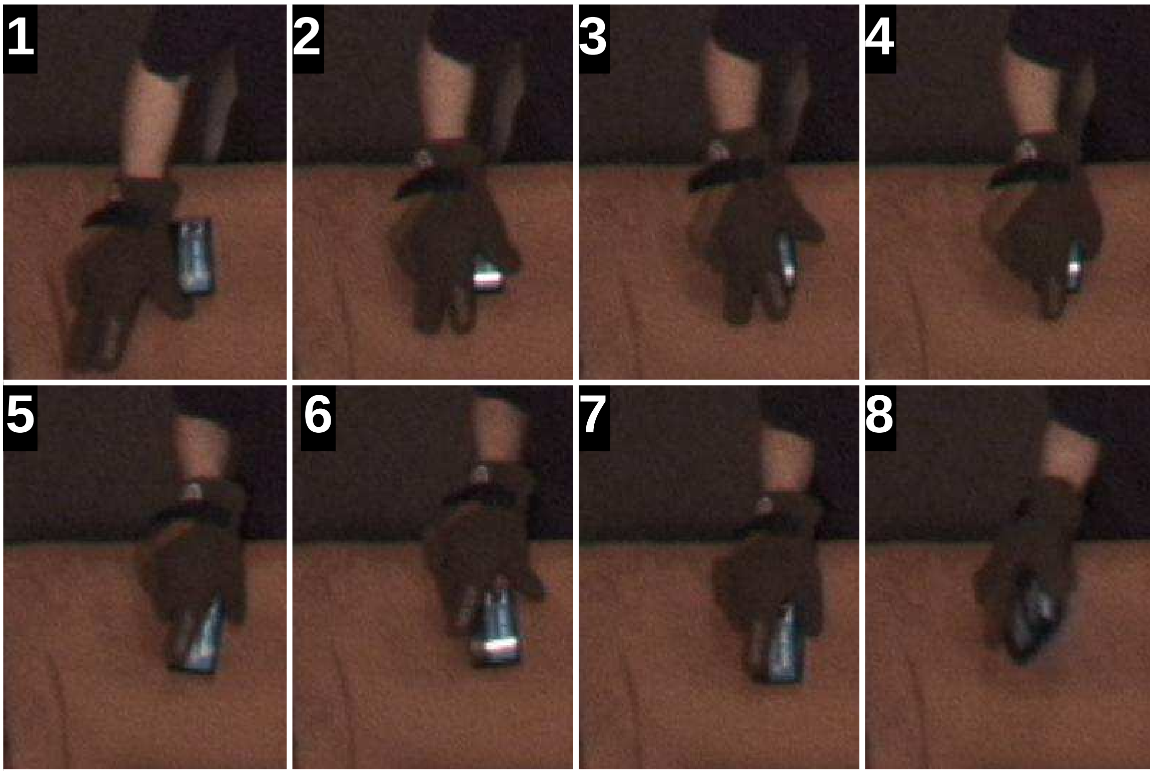

Figure 4-2: Grabbing a plastic oval box. 1. On a random exploration, the box is touched with the external side of the thumb. From 2. to 4. the hand is replaced around the box. From 5. to 7. the hand is placed at a different position for lifting the object. In this process the object is moved. In 8. the object is lifted.

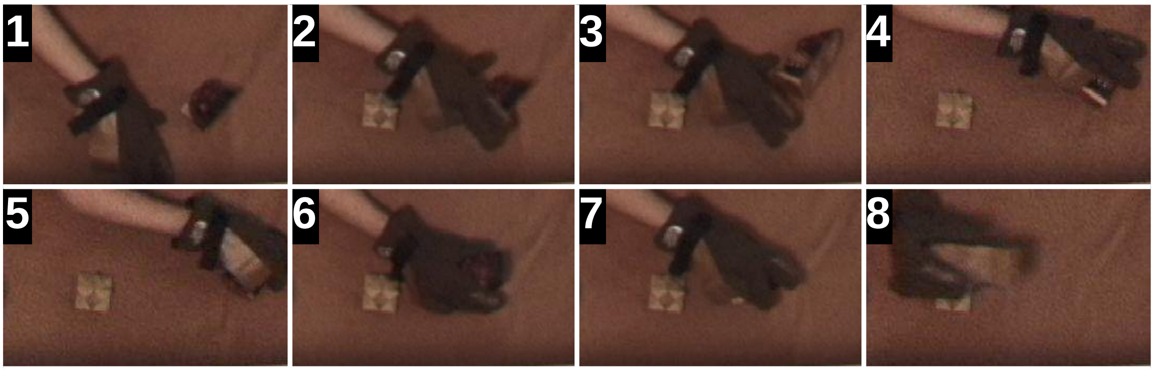

Figure 4-3: Grabbing a wooden statue. From 1. to 3. we observe that during exploration, the statue is hit and knocked over. However, in spite of this, the blindfolded person continues the exploration and comes in contact with the statue on 4. In 5. the statue is placed back to its original position. From 6. to 7. the hand is reoriented to grab and lift the object. In 8. the object is lifted.

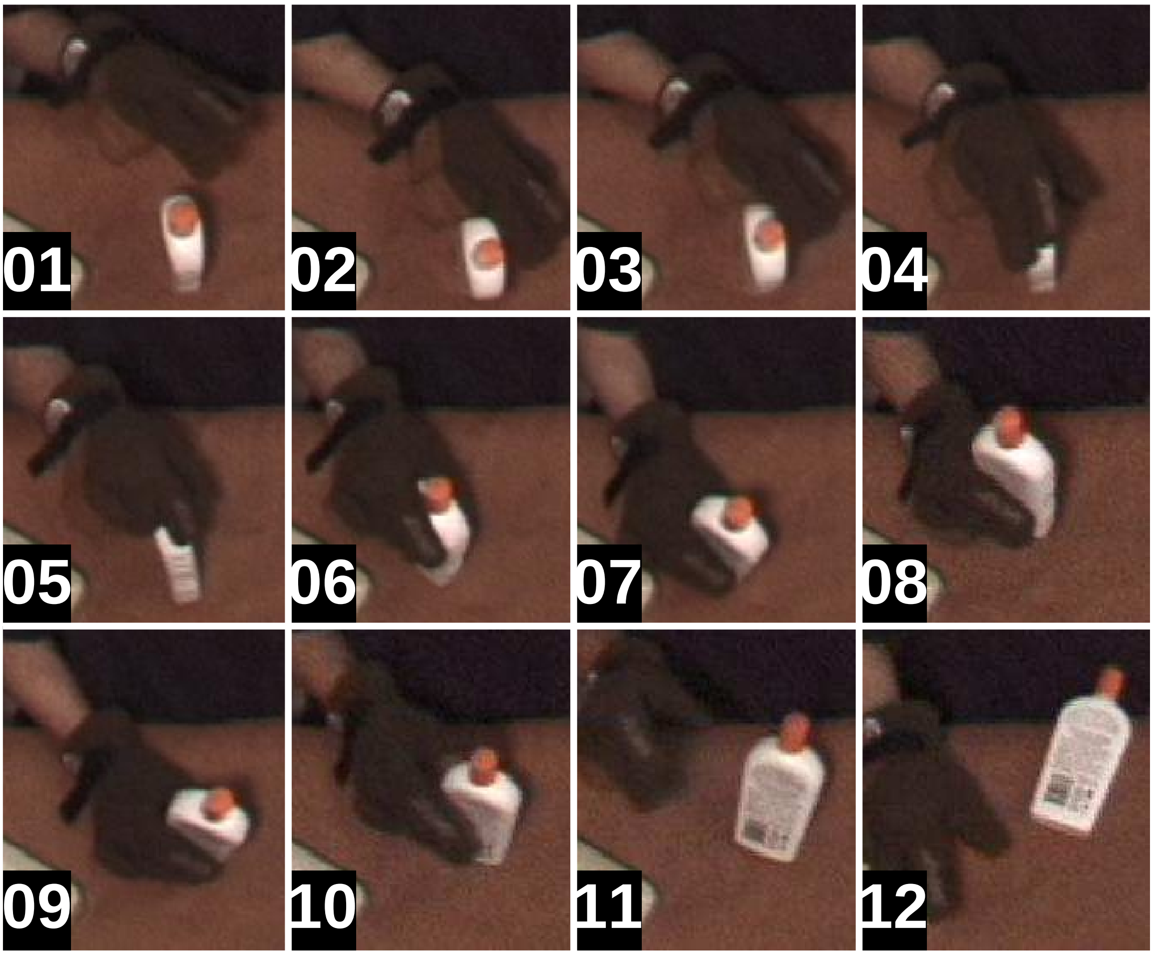

Figure 4-4: Grabbing a plastic bottle. From 1. to 3. We observe that the hand hits the bottle, moves it, but does not knock it over. The hand movements are gentle enough to get the bottle back to the original position. From 4. to 7. the hand is moved from grabbing the bottle by the tip to grabbing it by its bottom. In 8. the bottle is lifted. From 9. to 12. the bottle is placed on the desk. We observed that even though the person released the bottle in 10., presumably assuming stability, the bottle fell.

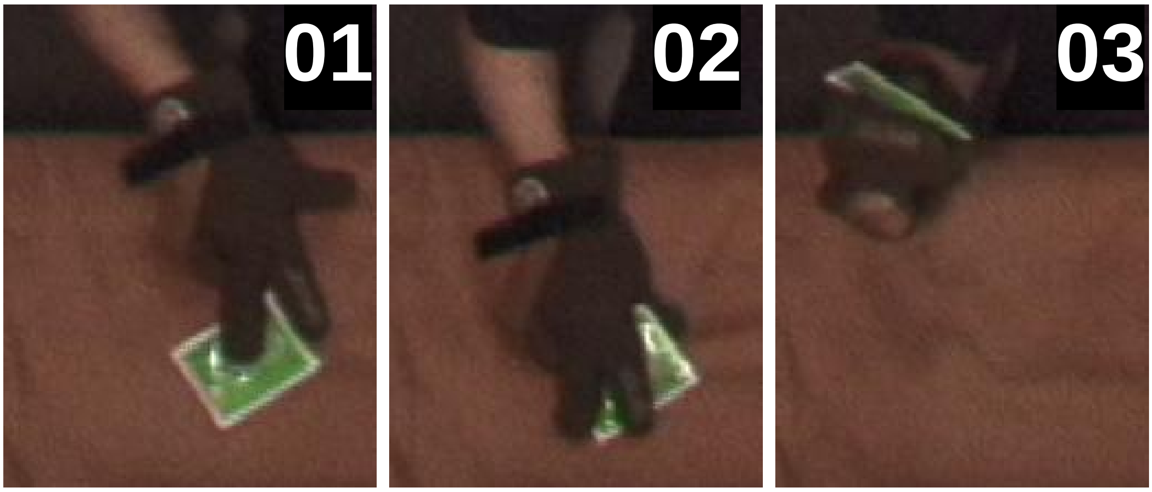

Figure 4-5: Grabbing a tea bag. 01. The hand deforms the bag. 02. The thumb gets underneath the bag to be lifted in 03.

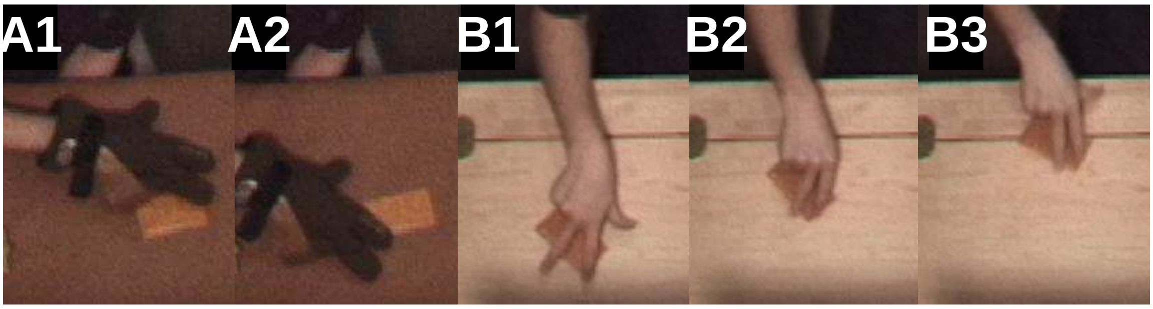

Figure 4-6: Grabbing a business card. In A1 and A2 we observe that the person wearing the glove was not able to detect the presence of the business card, even though, the hand came in contact with the card a number of times. However, we can see that the detection was done in B1 when a glove was not worn. In B2 and B3, we can observe that the lifting needed to be done by sliding the card to the edge of the desk.

once again before grasping the object. We can see that the person is quite effective in repositioning her hand to explore and grasp using mainly tactile feedback and proprioception.

The effectiveness of exploring can be confirmed when the subjects handle even more complex conditions. If we observe figure 4-3, we can see that the object is knocked over when the person's hand hits it. Nevertheless, the person can predict the new position of the object, re-grasp it and place it in its original orientation (respect to gravity) to finally lift it.

Another example can be observed on figure 4-4. We observe that the hand hits the bottle nearby its lid and the bottle leans over without falling. Subsequently, the person retracts her hand up to the point where the bottle is somewhat stable. Then, she moves her hand to the top of the bottle and grasps it. Nevertheless, to lift the bottle, her hand is moved to the bottom. We believe this maneuver is done to increase the friction surface which makes the grasp more stable. Another event to note is the falling of the bottle. On frame 10, the person believes that the bottle is stable on the surface and she opens her hand. When her hand moves away, the bottle falls but this event goes unnoticed by her tactile sensing.

More complex manipulation skills are shown when the objects have small dimensions. For example in figure 4-5, we can observe that a tea bag is deformed in order to later position the persons's thumb under it.

Nevertheless, there is a limit to sensing. In figure 4-6 we can observe that the person in the experiment, wearing a glove, can not detect the presence of a business card. The detection is very simple when the glove is not worn. However, lifting the business card required additional steps. The person slides the card to the edge of the surface to be able to grab it by the sides and lift it. We think that this is an interesting case that shows that without the right sensing resolution some tasks are not too simple to execute.

These observations give us an idea of the possibilities and the limitations expected when making the robot OBRERO explore its surroundings. For instance, it is not always possible to avoid leaning or knocking over an object when vision is not used.

Knocking over an object would not necessarily impede the robot of picking up the object if this condition can be detected. We are assuming that the object does not break if knocked over. And an important consideration is the resolution of the tactile sensors because it limits the kind of tasks that can be accomplished by the robot.

Spatial memory

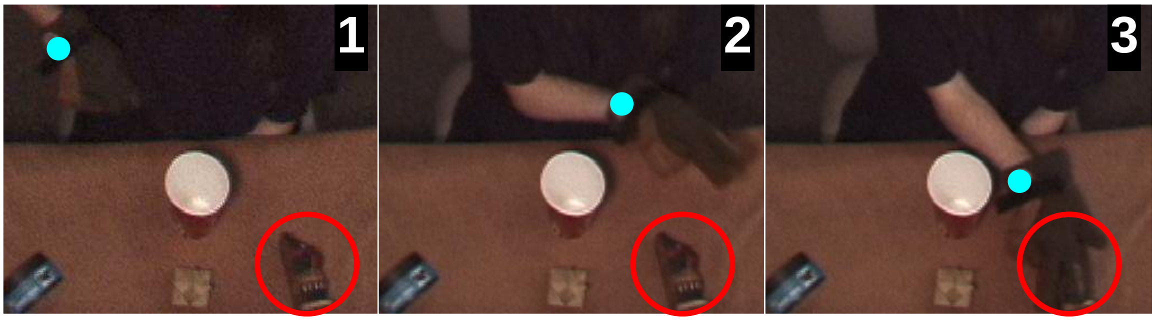

Figure 4-7: Reaching for the wooden statue. From 1. to 3. we can observe the trajectory of the hand/arm of the person to reach the wooden statue. The person have previously explored the environment and identified the objects on the desk. The trajectory to reach the target without collision is executed without vision. For clarity, the statue has been marked with a red circumference and the wrist with a blue circle.

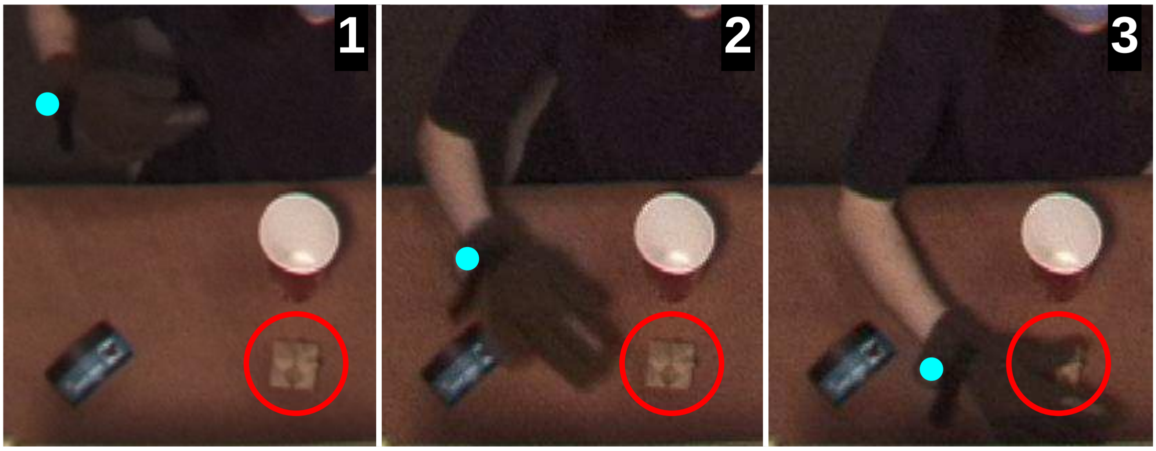

Figure 4-8: Reaching for the wooden star. From 1. to 3. we can observe the trajectory of the hand/arm of the person to reach the wooden star. The person has previously explored the environment and identified the objects on the desk. The trajectory to reach the target without collision is executed without vision. For clarity, the star has been marked with a red circumference and the wrist with a blue circle.

After exploring the environment, people are capable of moving their hand in a trajectory that avoids obstacles. This can be observed in figure 4-7. The person's hand is moved behind the cup and to the right to avoid hitting the cup. This trajectory is generated using previous tactile and proprioceptive information but not direct visual information.

The trajectory is generated for different objects in the environment as we can see in figure 4-8. In this figure, the same person reaches for the wooden star while avoiding the cup.

The spatial memory of people seems to be very powerful to reposition the hand and to generate trajectories. However, when the person is moved away from the desk and then back, they do not perform as well, presumably due to short term memory in proprioception.

OBRERO could take advantage of this observation by storing the arm/hand position of an object whose presence has been detected after exploring. This information would work as an initial reference for the arm's position to explore searching for the object. When many objects are present, this information could also help to find a trajectory free of obstacles for reaching.

Basic part matching

An example of the final tasks we asked our subjects to perform, included, finding the handset of a phone and placing it on the base of the phone, as shown in figure 4-9. In this example, we can observe that when the person moves the handset, it is not necessarily aligned with respect to the phone's base. When the parts come in contact, the person changes her grip of the handset and uses the base as support for the handset. That is, most of the handset weight is held by the base and the hand only guides the motion. We do believe that the performance on this task is highly influenced by previous experience of people handling the phone. Some people reported that their plan was to get close to the hook of the phone and just let the handset fall into place.

This observation shows how the environment itself is used to guide the task. OBRERO does not have the same dexterity in its hand as a person does; however, it should be capable of using the environment in its favor. For example, if it were to

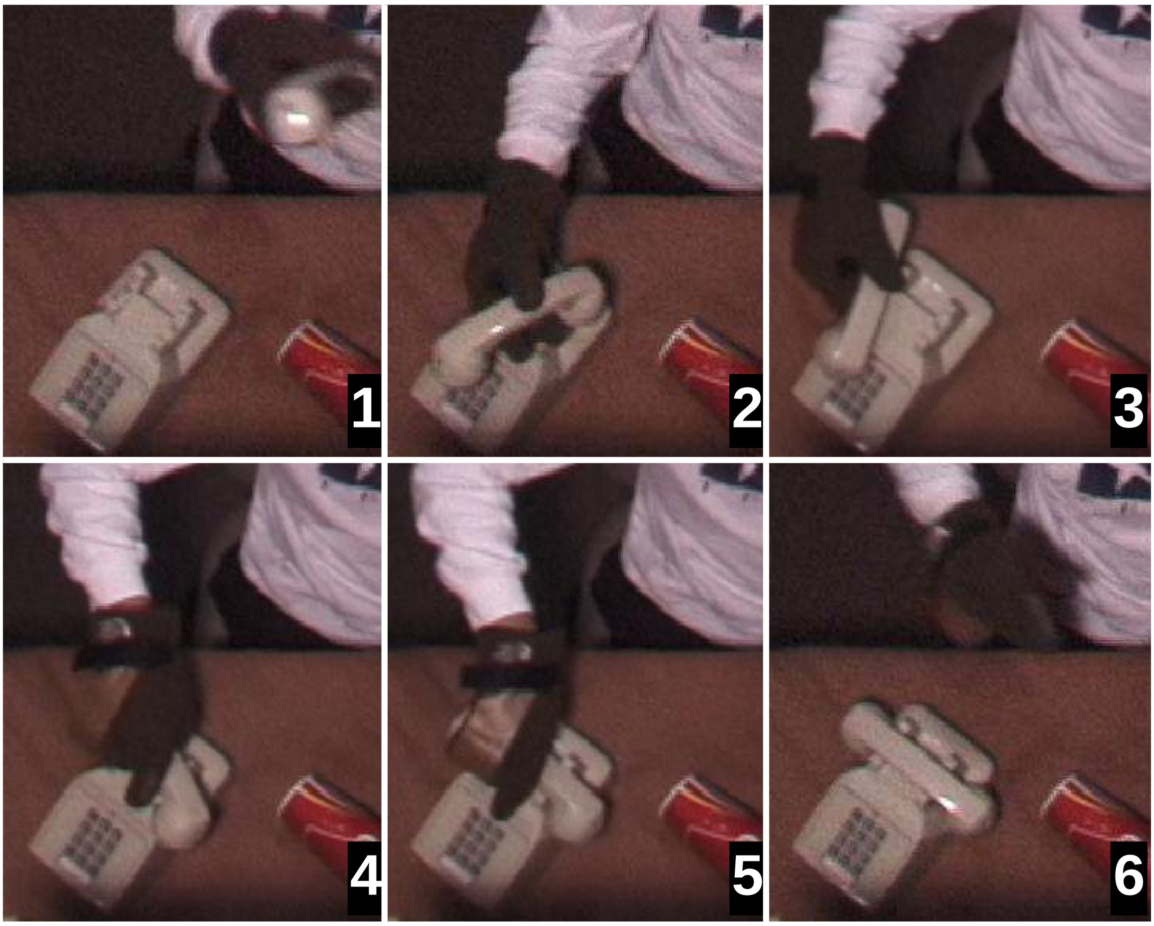

Figure 4-9: Matching the handset and the base of a phone. In 1. the handset is brought close to the base but the orientation is not the correct one. In 2. the handset comes in contact with the base. From 3. to 5 the grip is changed to reoriented the handset. In 6. the task was concluded.

place a cylindrical bottle in a rectangular box. The robot could move the box until the bottle touches an inner side of the box, then let the bottle slide guided by the side of the box.

4.3.2 Overall observations

- Exploration strategies vary. Some subjects face their palm in the direction of motion, others towards the desk. The speed at which people swing their arm is generally slow and cautious, with occasional contact with the table.
- Very light objects were consistently knocked over.

Table 4.1: Reaching results

Sub. No.	Reach 1	Reach 2	Reach 3
01	Yes	Yes	No
02	No	No	Yes
03	Yes	Yes	Yes
04	Yes	Yes	Yes
05	No	Yes	Yes
06	Yes	No	No
07	Yes	Yes	Yes
08	Yes	No	Yes
09	Yes	Yes	n/a
10	Yes	Yes	Yes

- Subjects quickly reorient their hand and arm for grasping if either their hand or their wrist makes contact with an object.

- Subjects exhibited a short-term but powerful memory for object location. This was observed when people were asked to reach an object that they have already explored. The operation was considered successful when the subject touched the object at the first attempt. In table 4.1, we can observe the results of each subject reaching for three different objects. The 76% of the reaches were successful. When a subject failed to make contact at the first attempt, she explored around and reached the object always.

- Sounds produced by objects and surfaces were used to identify them, compensating partially for the reduction in tactile sensitivity. This was occasionally misleading: one subject unwittingly dragged a tea bag over the desk, and thought from the sound that the surface was covered in paper. This observation inspired the work presented in section 7.1.

- Reducing the subjects tactile sensitivity affects the capability of detecting objects as shown in table 4.2. All subjects were capable of detecting a business card on the table when no glove was wore. However, when wearing a glove only one subject was able to detect the card. The tea bag was detected most of the time because of the noise it made.

- Subjects were not always capable of identifying objects that they did not see earlier. This is reported in table 4.3 where we can observe that in several occasions subjects were not able to identify some of the objects considered uncommon.

Table 4.2: Object detection

Sub. No.	Dominant hand	Detect(glove) Tea bag	Detect(glove) Buss.Card	Detect BussCard
01	Right	n/a	No	Yes
02	Left	Yes	No	Yes
03	Left	No	No	Yes
04	Right	No	No	Yes
05	Right	Yes	No	Yes
06	Right	No	No	Yes
07	Right	Yes	No	Yes
08	Right	Yes	Yes	Yes
09	Right	Yes	No	Yes
10	Right	Yes	No	Yes

Table 4.3: Object recognition

Sub. No.	Wooden Statue		Wooden Star		Plastic oval box	
	Answer	Y/N	Answer	Y/N	Answer	Y/N
01	Doll with feet	Yes	Star	Yes	n/a	n/a
02	I do not know	No	Star	Yes	Boat	Yes
03	A statue	Yes	Circle with points	Yes	n/a	n/a
04	A toy	Yes	Star	Yes	mouse	Yes
05	I do not know	Yes	I do not know	No	I do not know	No
06	I do not know	No	I do not know	No	I do not know	No
07	Something edged	No	Star	Yes	Oval type of thing	Yes
08	n/a	n/a	Star	Yes	Rectangular box	No
09	Indian face	Yes	Star	Yes	Soap box	Yes
10	I do not know	No	Pointy thing	Yes	I do not know	No

4.4 Summary

In this chapter we have observed people doing manipulation tasks without visual feedback and with coarse tactile information. The environment had several objects some of which were uncommon. People were effective at grasping objects, remembering their position and matching some simple parts.

This success at manipulating objects with no vision motivates our approach to robotics manipulation, in the sense that even course tactile sensing provided enough information to make manipulation possible. We believe that tactile sensing should be the main sensing modality given that manipulation implies to come in contact with objects. We agree that vision is important as well but we have taken an extreme position in these experiences to learn the importance of tactile feedback and how far one can go without vision.

Although, the idea of using tactile information in robotic manipulation has been present since its beginnings, the sensing technology available has not allowed its use (See chapter 3).

We believe that robotic manipulation would greatly benefit from extensively using tactile information. Our first step in this direction is presented on chapter 5.

Chapter 5

A sensitive approach to grasping

5.1 Introduction

Recent work in developmental robotics has emphasized the role of action for perception and learning [74,78,79]. Developmental psychology, on the other hand, recognizes that motor activity is of paramount importance for the correct emergence of cognition and intelligent behavior [15,37,95,106]. All embodied agents, either artificial or natural, have numerous ways to exploit the physical interaction with the environment to their advantage.

In robotics, actions like pushing, prodding, and tapping have been used for visual and auditory perception respectively [74,99]. More articulated explorative actions or grasping might increase these benefits, as they give direct access to physical properties of objects like shape, volume and weight. Unfortunately, all these aspects have not been extensively investigated yet. One of the reasons for this is that controlling the interaction between the robot and the environment is a difficult problem [104], especially in the absence of accurate models of either the robot or the environment (as it is often the case in non-engineered environments). The design of the robot can ease these problems. We know for example that having a certain degree of elasticity in the limbs helps to “smooth” and control the forces that originate upon contact. Another approach is to enhance the perceptual abilities of the robot. Traditional robotic systems in fact have perceptual systems that do not seem adequate for grasping. Haptic feedback in particular is often quite limited or completely absent. This is because, unfortunately, most of the tactile sensors commercially available are inadequate for robotics tasks: they are only sensitive to forces coming from a specific angle of incidence, rigid and almost frictionless.

OBRERO [100], described in chapter 2, was designed to overcome these limitations. It is equipped with series elastic actuators, which provide intrinsic elasticity and force feedback at each joint. The hand is equipped with tactile sensors (chapter 3) which provide a deformable and sensitive interface between the fingers and the objects.

In this chapter, we report a series of experiments where OBRERO exploits its sensing capabilities to grasp a number of objects individually placed on a table. No prior information about the objects is available to the robot. The use of visual feedback was voluntarily limited. Vision is used at the beginning of the task to direct the attention of the robot and to give a rough estimation of the position of the object. Next, the robot moves its limb towards the object and explores with the hand the area around it. During exploration, the robot exploits tactile feedback to find the actual position of the object and grasp it. The mechanical compliance of the robot and the control facilitate the exploration by allowing a smooth and safe interaction with the object. Results show that the haptic information acquired by the robot during grasping carries information about the shape of the objects. Further analysis of the tactile data allows us to detect slippage and sense forces applied over the object being held by the robot.

This chapter is organized as follows. Section 5.2 briefly reviews the importance of haptic feedback for manipulation in infants and adults. Section 5.3 provides some implementation details and describes the grasping behavior. The latter is evaluated in Section 5.4. Additional analysis of the data provided by tactile feedback is presented in sections 5.5 and 5.6. The limitations of the implementation are presented in section 5.7. In section 5.8, we compare the current implementation with the observations presented in chapter 4. Finally, section 5.9 draws the conclusions of this work.

5.2 Haptic feedback, perception and action

In adults, several studies have revealed the importance of somatosensory input. For example Johansson andWestling [53] have studied in detail what feedback is provided by the skin during object lifting tasks and how it is used to control the movements of the fingers. The results of these experiments proved the importance of somatosensory feedback: they showed that human subjects had difficulties avoiding object slipping when they had their fingertips anesthetized, even with full vision [52].

Haptic feedback has an important role for object perception as well. Lederman and Klatzky [67] identified and described a set of motor strategies named *exploratory procedures* used by humans to determine properties of objects such as shape, texture, weight or volume.

Little is known concerning how infants use tactile sensing for manipulation [95]. In some circumstances children exploit tactile feedback to learn about objects [96]. Streri and Pêcheux measured the habituation time of newborns (2 months and 5 months old) during tactile exploration of objects placed in their hands. In this experiment children spent more time exploring novel rather than familiar objects, even when they did not make visual contact with the hand.

Motor abilities of children are quite limited during the first months of development. This does not prevent infants from using their hand to engage interaction with the world. The importance of motor activity for perceptual development has been emphasized in developmental psychology [37, 106]. Researchers agree on the fact that motor development determines the timing of perceptual development. In other words the ability of infants to explore the environment would determine their capacity to perceive certain properties. Accordingly, perception of object features like temperature, size and hardness is likely to occur relatively early in development, whereas properties requiring more dexterous actions like texture or three dimensional shape would emerge only later on (see [15] for a review).

5.3 Grasping behavior

In this section we describe the implementation of a grasping behavior in the robot OBRERO. The grasping is implemented using our approach, that is sensitive manipulation. In this implementation we assume that OBRERO has a table in front of it where objects will be presented by a person. The table is reachable by the robot's hand. The robot does not have a model of the objects presented. The robot's attention is captured by moving an object in the table. The robot's response is to move its hand toward the object to explore it and grab it. The object can be placed anywhere within reach of the robot's hand. In the following sections, we describe a few perceptual and motor competencies required for the robot to be able to control the body in a meaningful and safe manner: this includes a simple attention system to spot the objects to be grasped and the ability to control the body to reach out for them. At the end of the section we describe how these capabilities are integrated in the grasping behavior.

5.3.1 Attention System

Motion is a simple yet powerful cue to select points of interest in the visual scene; for an active camera system this is still true assuming we can estimate the motion of the background and account for it. In this thesis, we use the algorithm proposed by [58], which uses a 2D affine model to robustly estimate the image motion resulting from the background. In short, the algorithm measures the motion of each pixel with a block matching procedure, and performs a least square fitting of the global affine model. Using the affine model the algorithm predicts the motion of each edge, and marks as foreground those edges that poorly match this prediction. Under the assumption that the majority of the image motion is due to the background, these edges can be used to build a saliency map to direct the attention of the robot. Figure 5-1 shows the implementation of this system.

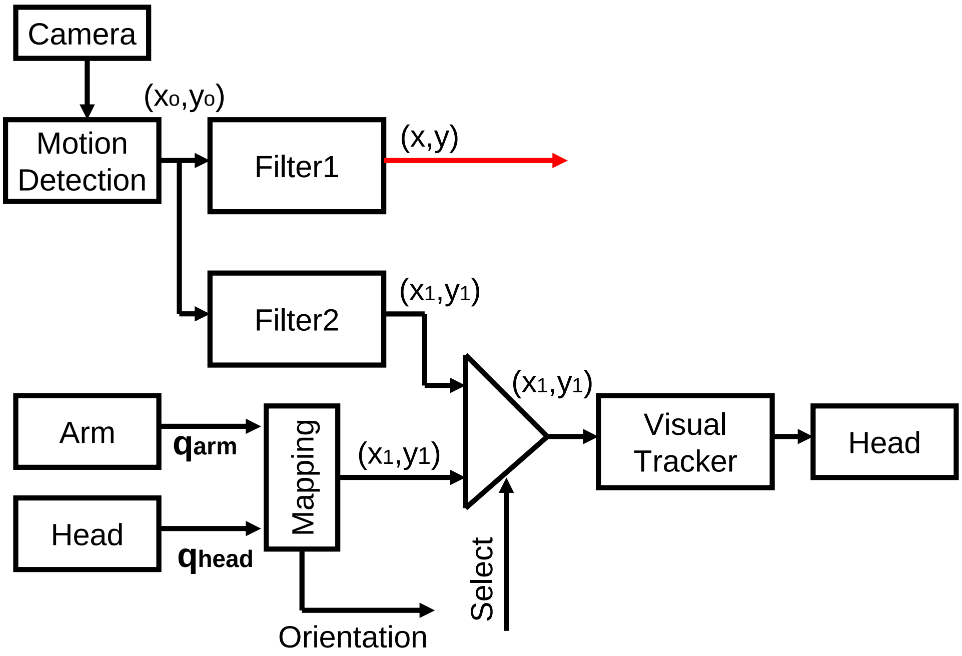

Figure 5-1: The position, in retinotopic coordinates (x_0, y_0), of the motion detected(section 5.3.1) is used to build time distributions during two different time periods (Filter1 and Filter2). The output of Filter2 together with a visual tracker are used to orient the head toward the object. The robot can also orient its head toward its hand by using the select input. The output of Filter1 is used to indicate the presence of an object and trigger behaviors as explained in section 5.3.6

5.3.2 Eye-hand coordination

We decided to focus on explorative actions rather than precise, goal directed, actions towards the target objects.

However, we cannot hope to program the robot to perform a blind exploration of the entire workspace. A possible solution is to constrain the exploration to the area of the workspace where the object is detected visually. Since the 3D location of the object is not available, reaching is performed in 2D; the exploration procedure allows the robot to find the actual position of the object. The motor skills required for reaching and exploring can be learned from the visual ability to localize the hand and compute the orientation of the arm.

5.3.3 Hand Localization

A visual module detects the hand and computes the orientation of the arm in the image. The initial step of the hand detector consists in running a high frequency filter. All points whose frequency is below a certain threshold (fixed a priori) are discarded. A blob detector is run on the resulting image and the biggest blob is selected as the arm. The orientation of the arm is computed as the orientation of the line passing through the top-most and bottom-most pixels of the arm area. Next, specific features (the small circular black and white potentiometers on the fingers) are searched on the arm area (figures 5-2.1 and 5-2.2). The hand is identified if more than two of these features are found. The detection just described proved reliable enough for our purposes and was used as a short-cut in place of other, more general, methods [74, 79].

The visual feedback of the hand could be used for closed-loop control. However, closed-loop control is not always suitable. This happens for example in presence of occlusions or when the hand is not within the visual field. Open-loop control is an alternative solution. A possible open-loop control consists of a mapping between the fixation point of the head and the arm end-point [73]. The advantage of this approach is that the mapping can be easily learned if the robot is able to look at the hand.

Another approach uses the output of the hand detector to learn a direct mapping between the arm proprioception (encoder feedback) and the position of the hand in the image [79]. The direct (forward) mapping can be inverted locally to control the arm to reach for a visually identified target. The solution we adopt here is similar: in a discovery phase the robot tracks the hand as the arm moves to randomly explore the workspace. This behavior allows the robot to acquire samples in the form:

$$\begin{pmatrix} x & y & \alpha & q_{head} & q_{arm} \end{pmatrix}_{0,1...,k}$$

where x, y and α are the coordinates of the hand and the orientation of the arm in the image, q_{head} and q_{arm} are the position of the head and arm respectively. Given q_{head} it is possible to convert x and y into an egocentric reference frame:

$$\begin{bmatrix} \theta_h & \phi_h \end{bmatrix}^T = g_{head}\left(\begin{bmatrix} x & y & q_{head} \end{bmatrix}^T\right) \tag{5.1}$$

θ_h and ϕ_h represents the polar coordinates of the hand in the reference frame centered at the base of the head (azimuth and elevation). Basically g_{head} includes knowledge of the inverse kinematics of the head and the parameters of the camera. The opposite transformation maps polar coordinates into the image plane:

$$\begin{bmatrix} x & y \end{bmatrix}^T = h_{head}\left(\begin{bmatrix} \theta_h & \phi_h & q_{head} \end{bmatrix}^T\right) \tag{5.2}$$

Given these two transformations a neural network can be trained to learn the following mapping:

$$\begin{bmatrix} \theta_h & \phi_h & \alpha \end{bmatrix}^T = f\left(q_{arm}\right) \tag{5.3}$$

which links the arm posture q_{arm} to the polar coordinates of the hand $[\theta_h, \phi_h]^T$ and the orientation of the arm α. This mapping was learned online by using the neural network proposed by [91].

The mapping of equation 5.3 allows computing the polar coordinates of the hand with respect of the robot from the encoders of the arm. Whenever required equation 5.2 maps the polar coordinates back onto the image plane.

5.3.4 Reaching

Suppose we want to move the arm towards a location of the workspace identified visually. Let $\begin{bmatrix} x_t & y_t \end{bmatrix}^T$ be such position. Knowing q_{head} from equation 5.1 we can convert the target position into the body centered reference frame $\begin{bmatrix} \theta_t & \phi_t \end{bmatrix}^T$. The reaching problem can now be stated as a the minimization of the following cost function:

$$\min_{q_{arm}} (C) = \min_{q_{arm}} \left\| \begin{bmatrix} \theta_t & \phi_t \end{bmatrix}^T - \begin{bmatrix} \theta_h & \phi_h \end{bmatrix}^T \right\|^2 \tag{5.4}$$

where θ_h and ϕ_h are computed from equation 5.3.

Assuming a stationary target the minimum of equation 5.4 can be found by gradient descent. The gradient of C is proportional to the Jacobian transposed of the manipulator, that is:

$$\nabla C = -2\nabla f\left(q_{arm}\right) = -2J^T\left(q_{arm}\right) \tag{5.5}$$

$\nabla f\left(q_{arm}\right)$ was approximated by partial differentiation of equation 5.3. Because the basis functions used by the neural network are gaussians this was easily done analytically (another approach is to perform numerical differentiation).

To summarize we have described a method to compute the arm commands required to reach for a visual target. The method employs the forwards kinematics of the arm. The direct kinematics is learned by the robot as described in the previous section. The reaching problem is solved iteratively by using an approximation of the arm Jacobian. The latter is obtained by differentiating the basis functions of the neural network approximating the direct kinematics. This procedure is carried out online without using the real visual feedback of the hand.

In the robot, visual information (and hence the mapping of equation 5.3) is two-dimensional and does not carry any information about distance. The solution found by descending the gradient of the direct kinematics takes care of minimizing the distance between the target and the hand *on the image plane*, and as such, is not concerned with the third dimension R (the distance between the hand and the head,

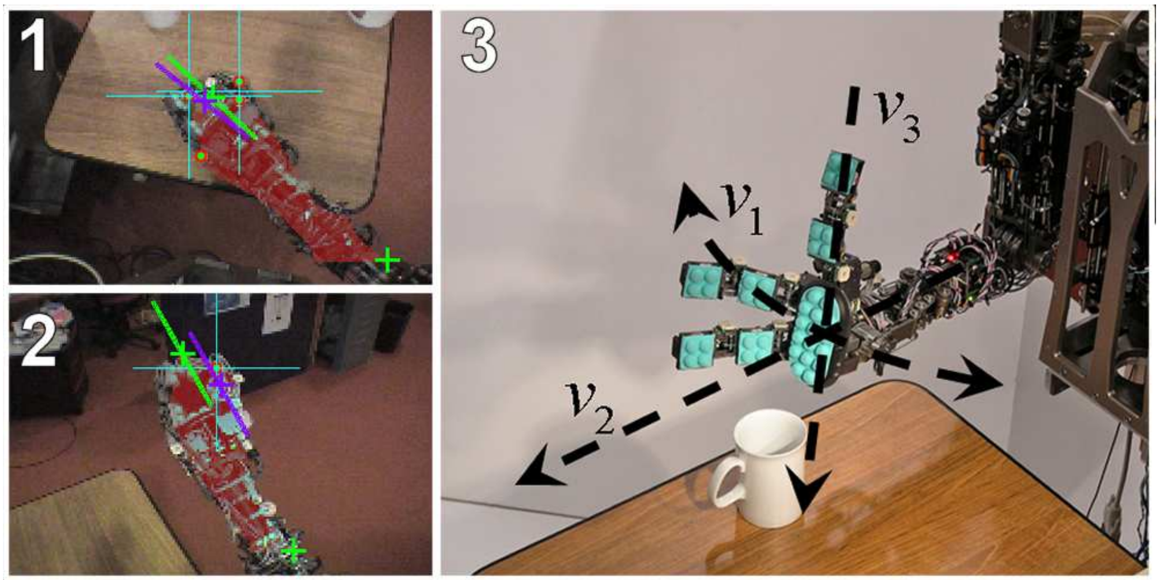

Figure 5-2: Left, frames 1 and 2: hand localization and arm orientation. Right, frame 3: exploration primitives. Primitives v_1 and v_2 are perpendicular and parallel to the arm orientation. v_3 is along the null space of the arm Jacobian. For simpler understanding these primitives are here sketched in the cartesian plane, but they are actually computed in the joint space (see Section 5.3 for more details).

along the optical axis of the camera). In practice however the components of the gradient along R are small compared to the others. The value of R at the end of the reaching movement depends on the initial position of the arm; we chose this value so to keep the hand above the table.

5.3.5 Exploration primitives

Starting from the direct mapping of the hand position and arm orientation we can identify a set of explorative primitives, that is a set of vectors in joint space that allows the robot to explore the arm workspace. We chose three vectors v_1, v_2 and v_3, as follows (see also Figure 5-2):

v_1: moves the hand along the direction perpendicular to the arm. It is computed by planning a reaching movement towards a point a few pixels away from the hand along the line perpendicular to the orientation of the arm.

v_2: moves the hand along the direction of the arm. It is computed by planning a reaching movement towards a point a few pixels away from the hand along the arm.

$v_3 \in \ker\left(J\left(q_{arm}\right)\right)$: v_3 lays in the null space of the arm Jacobian; in our case the null space of the Jacobian consists of those vector that do not affect either the projection of the hand onto the visual plane or the orientation of the arm. These vectors produce a movement of the hand along the optical axis of the camera, or, in other words, along R.

5.3.6 Implementation of the robot's behavior

In this section, we describe the implementation of a behavior that allows the robot to grasp objects. A diagram of this implementation is presented in figure 5-3 where we can observe the interconnection of five behaviors: *reaching*, *depth exploration*, *hovering*, *pushing*, and *grasping*.

We will refer to figure 5-4 to explain the interaction between behaviors. It should be noted in the figure that a behavior can be inhibited or enabled. When a behavior is inhibited, it will not send commands. When a behavior is enabled, the behavior can be either active or inactive. When, it is active, it is sending commands because the programmed conditions have been meet.

In figure 5-4.1, the robot's arm is in a safe position. A safe position above the table is defined for this implementation. We start by describing the *reaching behavior*. The attention system, previously described, detects when a experimenter waves an object in front of the robot. The head tracks the object until it remains stationary within the workspace of the arm. If the arm is in a safe position, the robot moves the arm towards the object (figure 5-4.2). Reaching is not accurate enough to guarantee a correct grasp. Since no three dimensional information is available the arm reaches a region above the object (see Section 5.3.4). At this point the *reaching behavior* enables the other behaviors (figure 5-4.3). The robot computes the directions v_1, v_2, and v_3 and explores using the following three behaviors:

- *depth behavior*, moves the hand "downwards" along v_3;
- *hovering behavior*, moves the hand back and forth along v_1;
- *pushing behavior*, moves the hand along v_2;

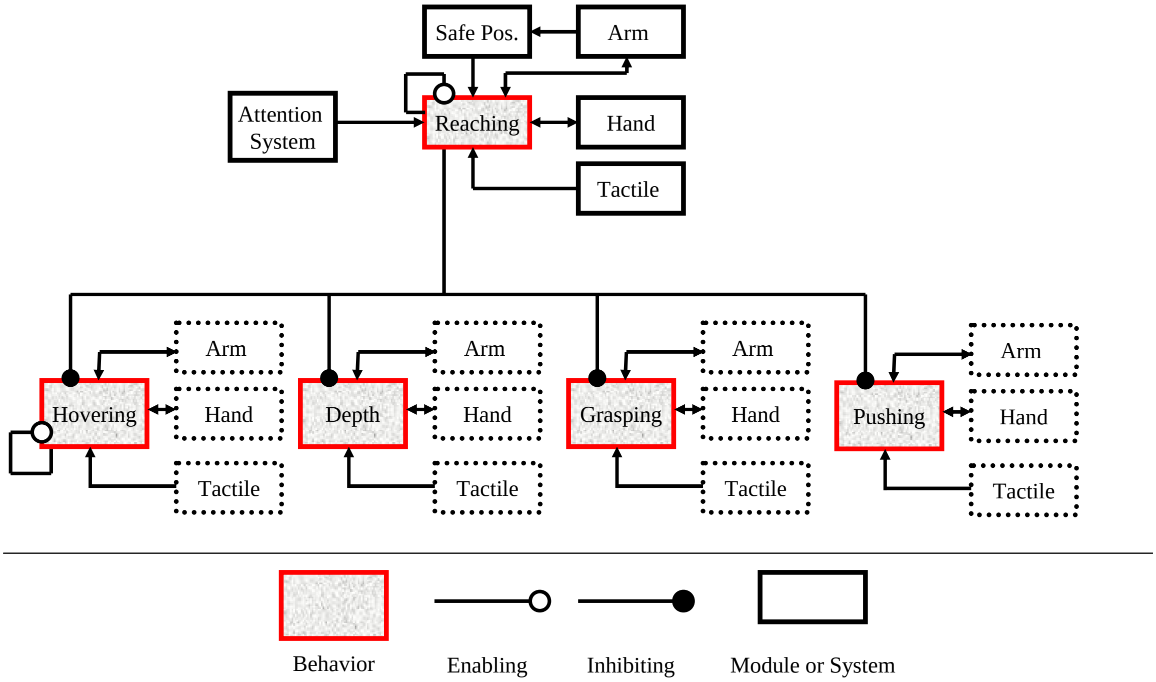

Figure 5-3: We can observe the behaviors *reaching*, *hovering*, *depth*, *grasping*, and *pushing*. We can also observe the modules tactile, arm, and hand that interface with the hardware of the robot. The sensing and actuation modules are repeated in dashed lines for clarity. These modules are also presented in the overall architecture for reference (figure 2-2 inside the linux nodes). The modules arm and hand provide sensing information (i.e.positions and forces) and receive commands (i.e. positions' setpoints). The tactile module only provides information, it does not receive commands. Each of the behaviors are connected to the sensing and actuation modules as shown by the arrows. The line with double arrows indicates that the behavior can receive sensing information and send commands. The line with one arrow indicates that the behavior only receives sensing information. The Safe Position module receives the angles of the arm and outputs a binary output that indicates if the arm's position is between the range considered safe or not. The attention system output is the coordinates of an object detected. Details of the attention system are presented in figure 5-1. The *reaching* behavior receives the outputs of the attention system and the safe position module. The connections represented by lines terminated on a circle represent the capability of enabling or inhibiting a behavior. In the case of the *hovering* behavior, it has the capability of self-inhibition and it runs when both connections are enabled.

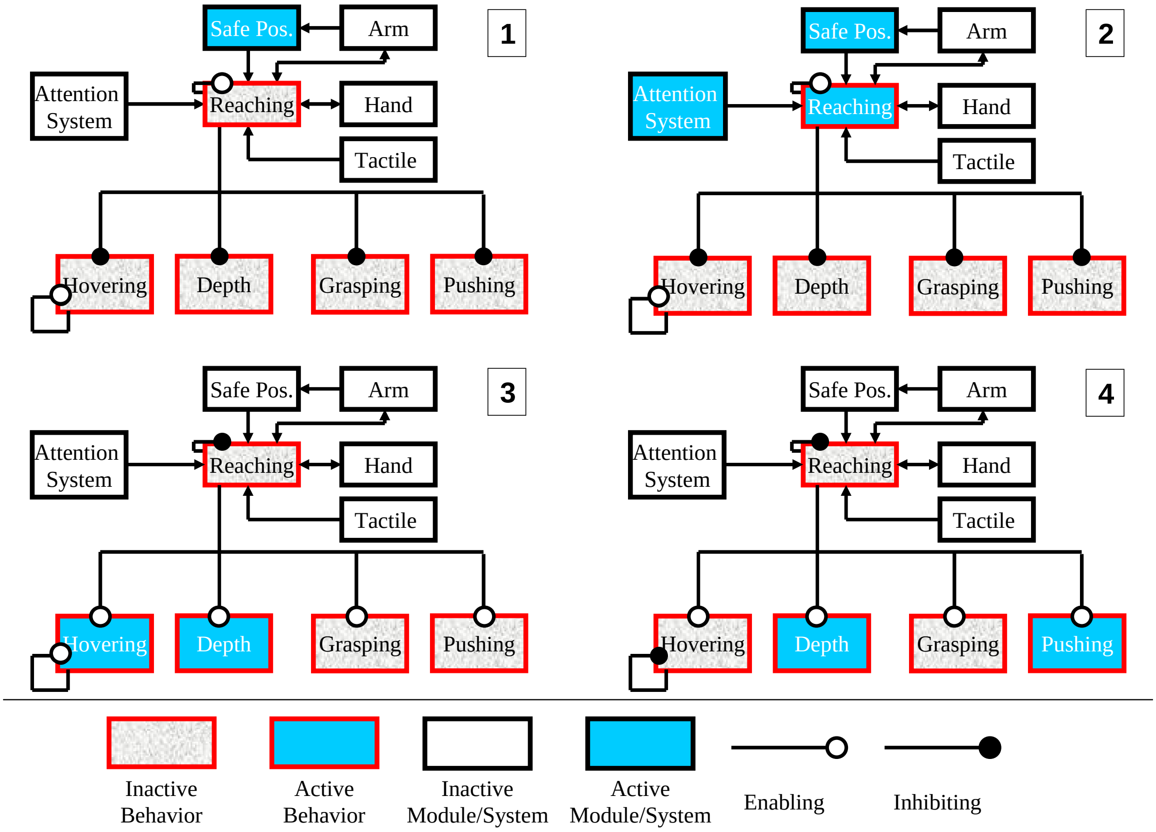

Figure 5-4: Behaviors interaction. 1. All the behaviors but *reaching* are inhibited as shown by the color of the circles. We assume that the arm is in the safe position. Therefore, the output of the safe position module indicates this condition (colored blue) to the *reaching* behavior. 2. Repetitive motion of an object in front of the robot is detected by the attention system (colored blue) and 2D coordinates of the object are sent to the *reaching* behavior. The *reaching* behavior, knowing that the arm is in a safe position and the goal position, sends commands to the arm to execute this motion (colored blue). 3. The output of the safe position module changes when arm moves. We are assuming that there is no more repetitive motion and that the output of the attention system also changes. When the arm gets to the final position, *reaching* inhibits itself and all the other behaviors become enabled. *hovering* and *depth* are colored blue to indicate that they meet the conditions to send commands. If there is no contact with an object *hovering* and *depth* remain active. There is coordination between these two behaviors (not indicated in the figure) to execute one after another. 4. When the index finger makes contact, the *hovering* behavior inhibits itself. The same condition makes the *pushing* behavior active (colored blue) which moves the arm forward. The *depth* behavior (colored blue) keeps moving the hand down until it makes contact with the table which is indicated by the infrared sensors. As a consequence of the *pushing* behavior being active the palm touches the object. This actives the *grasping* behavior which runs a preprogrammed sequence.

The *depth behavior* moves the hand along the direction of the optical axis of the camera and adjusts the height of the hand with respect to the object/table. To avoid crashing the hand into the table this behavior is inhibited when the infrared proximity sensor detects an obstacle (usually this happens close to the table). The *hovering behavior* and the *depth behavior* are activated at the beginning of the exploration. The goal of this initial phase is to adjust the position of the hand until the index finger touches the object. This allows adjusting the position of the hand along the directions v_1 and v_3. During the exploration the arm stops when the hand detects the object, to avoid pushing it away or knocking it over; if no contact is detected, the amplitude of the exploration is extended (this increases the probability to touch the object in case the reaching error is large). The exploration terminates when the contact with the object is detected by any of the tactile sensors placed on the index finger. At this point the *hovering behavior* is suspended and the *pushing behavior* activated (figure 5-4.4). The "pushing" movement along v_2 brings the palm in contact with the object while the *depth behavior* takes care of maintaining the correct distance with the table. When the robot detects contact on the palm the exploration stops and the *grasping behavior* is activated. The *grasping behavior* simply closes the fingers to a specific position. The low impedance of the joints allows the fingers to adapt to the different objects being grasped.

Figure 5-5 reports an example of the robot grasping a porcelain cup. The relation between the behaviors and the actions of the robot are presented in figure 5-6. The *grasping* behavior proved to be quite reliable, as several repeated tests show in Section 5.4

5.4 Evaluations

The *grasping* behavior described in Section 5.3 was evaluated by presenting different objects to the robot and by counting the number of successful grasps. We chose objects of different size and shape: a plastic bottle, a plastic rectangular box, a porcelain cup and a plastic cup (see figure 5-7). Some of the objects were partially

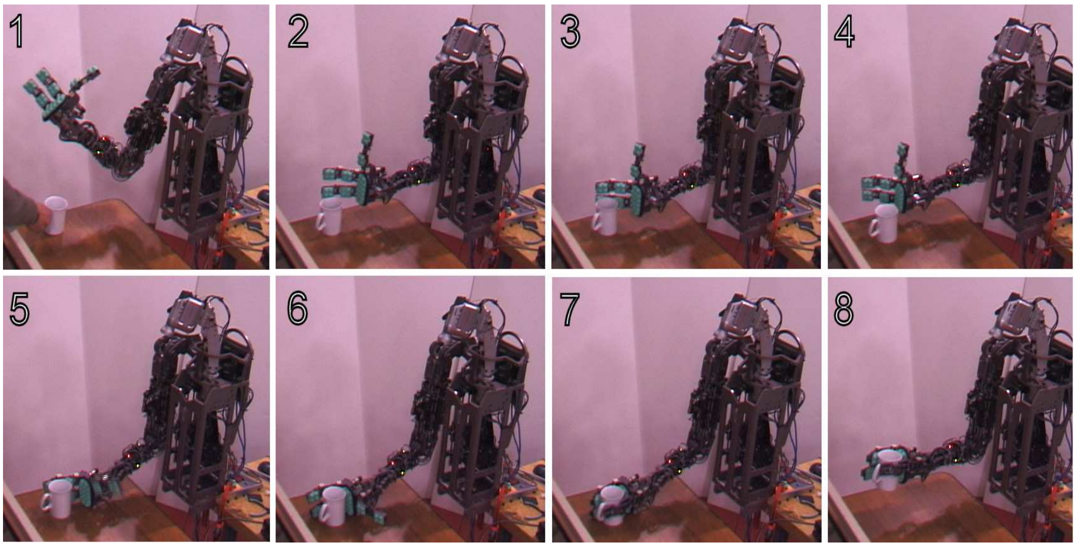

Figure 5-5: Grasping behavior: an example. Sequence of the robot grasping a porcelain cup. Frame 1: the cup is presented to the robot. Frame 2: the robot reaches for the cup. Frames 3 to 6: the robot explores the space and uses tactile feedback to find the object and adjust the position of the hand. Frames 7 and 8: the robot grasps and lifts the cup.

Table 5.1: Objects.

	Description	Weight(Kg)	No.Trials	No.Failures	Contains
1	Plastic bottle	0.265	22	0	Vitamins
2	Porcelain cup	0.255	24	1	Nothing
3	Plastic cup (Starbucks)	0.220	24	4	Bolts
4	Rectangular box (Nesquick)	0.240	24	2	Nesquick powder

filled, so that the weight was roughly uniform among all objects (about 220-250 grams, see Table 5.1). The robot had no prior knowledge about these objects.

Each object was presented to the robot more than 20 times and randomly placed on the table. Overall the number of grasping trials was 94, of which only 7 were not successful. In some of these trials, the robot managed to grasp the object, but was not able to hold it because the grip did not produce enough friction. In a few cases the tactile sensors failed to detect the object and the exploration was aborted before the object was actually grasped (more details are reported in Table 5.1).

As a further validation, we clustered the haptic information originated from the grasping. We collected the hand feedback at the moment the robot lifted the object; the idea is that given the intrinsic compliance of the hand, its configuration and the

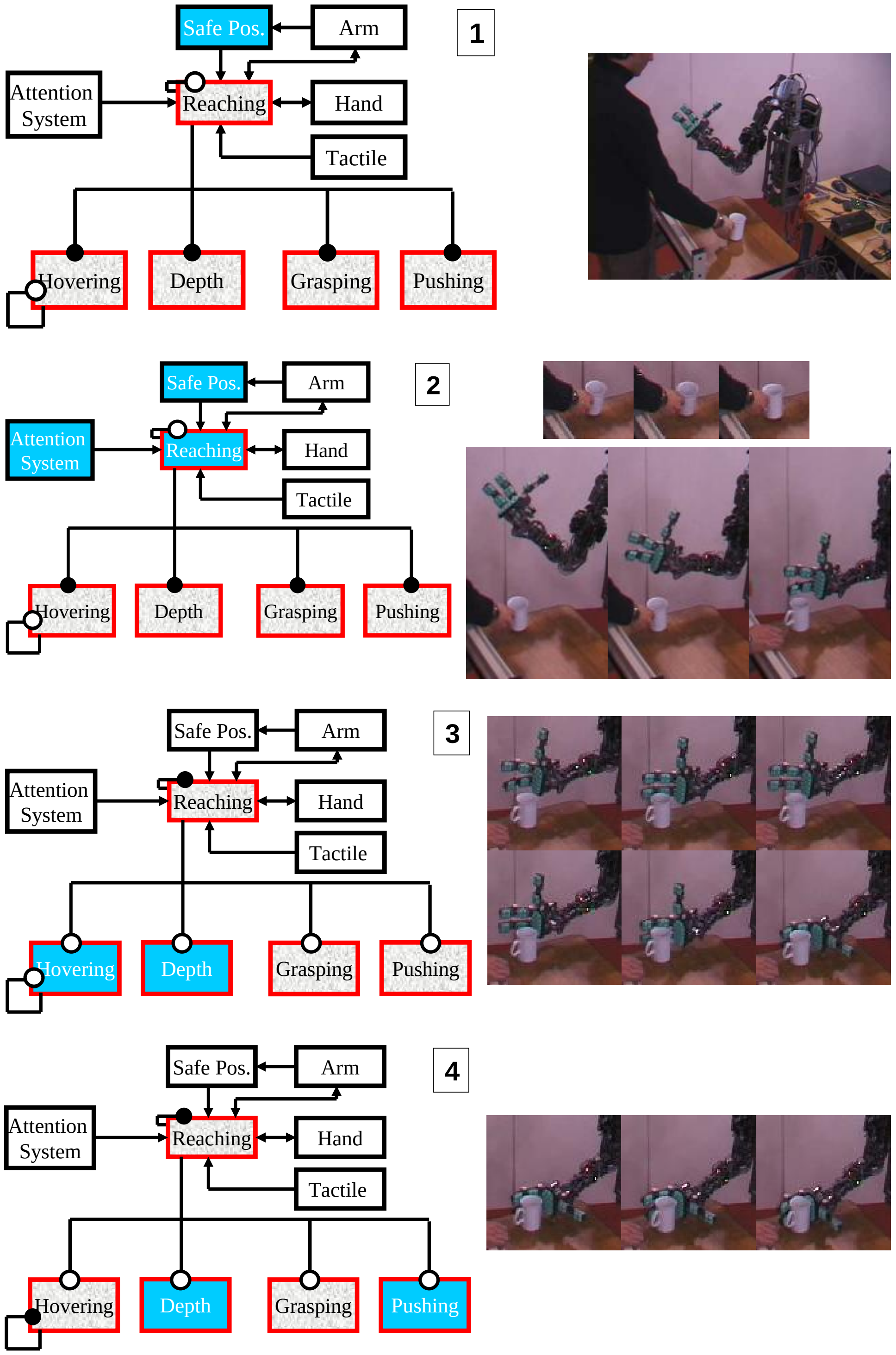

Figure 5-6: From top to bottom, we observe a sequence of the behaviors' activity and the generated robot's actions. The explanation of the behaviors is presented in figure 5-4. 1. The arm is in the safe position. 2. The cup is being shaken and the arm reaches toward the cup. 3. The arm moves side to side and downwards. 4. The arm moves toward the cup.

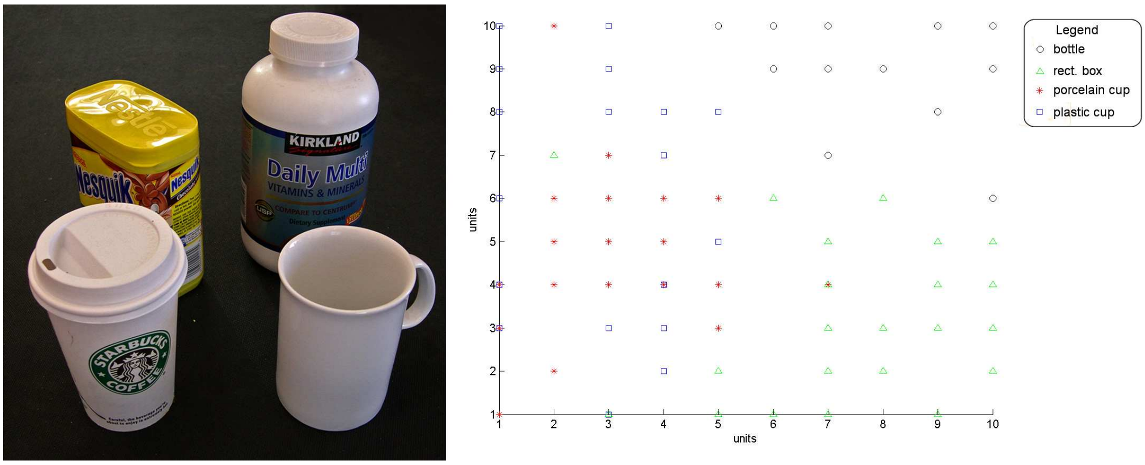

Figure 5-7: Left: the set of objects used in the experiments: a plastic bottle, a porcelain cup, a plastic cup and a rectangular plastic box. Some objects were partially filled to increase the weight (all objects weighed about 220-250g). Right: result of the clustering. Black circles, green triangles, red stars and blue squares represent respectively the bottle, the rectangular box and the porcelain and the plastic cups. The two cups are not clearly separated because they have similar shape in the area where they were grasped.

force exerted by each joint depend on the shape of the object being grasped. The hand feedback was clustered by means of a Self Organizing Map (SOM). The results show that the bottle, the rectangular box and the cups form three clusters. Unfortunately the clusters formed by the two cups are not clearly distinguishable. This is probably due to the fact that the hand grasped the objects from the top, and that in that part the two objects are quite alike (both are circular with similar diameter). In these cases the limited number of fingers (three) made it hard to distinguish between the cylindrical and conic shape of the cups. Together the results prove that the *grasping* behavior of the robot is reliable. The high number of successful trials shows that the haptic feedback manages to drive the robot during the exploration until it finds the object and grasps it. This is further demonstrated by the clustering, which show that the behavior allows extracting meaningful information about the physical properties of the objects (i.e. their shape).

5.5 Analysis of tactile data

In the previous sections, we showed how simple tactile feedback can be efficiently used to grasp unknown objects. In this section, we further analyze the tactile data to obtain useful information.

5.5.1 Sensitivity

The tactile sensors used are capable of providing information useful for grasping and manipulation in general. We further analyze these data in this section. But first we would like to emphasize the difference between compliant sensors and state of the art tactile sensors by presenting a couple of evaluations.

In figure 5-8, we observe an example of the compliant sensors working. In this case, the fingers of the hand close until contact is detected. We see that the sensitivity of the tactile sensors is enough to detect the contact with an egg and avoid the fingers to exert excessive force to break it. It is important to note that the sensors conform to the egg which helps to constrain the egg's motion and distributes the force reducing the pressure on the egg's surface.

In order to contrast our compliant sensor features, we refer to the sensor developed by Maheshwari and Saraf [71]. This sensor is capable of recovering the shape embossed on a US penny, but it is not capable of conforming to the egg. Moreover, given that it is flat and rigid, the force applied concentrates in a small area which causes great stress in that area. As reference, this sensor needs at least $9KPa$ to detect contact. This stress can cause a fracture on the egg surface.

We further evaluate the compliant sensors on a more delicate task. In figure 5-9, we observe the robot gently gripping a paper cylinder while the cylinder is only slightly deformed. There is a weight on the base to keep the cylinder in place. The weight's size is smaller than the circle of the base so it does not affect the cylinder structure. The robot's hand approaches the cylinder by the side. Once the tactile sensors detect the contact, the robot rotates its finger and closes the thumb slowly up until contact is detected. The sensitivity of the compliant tactile sensor is such

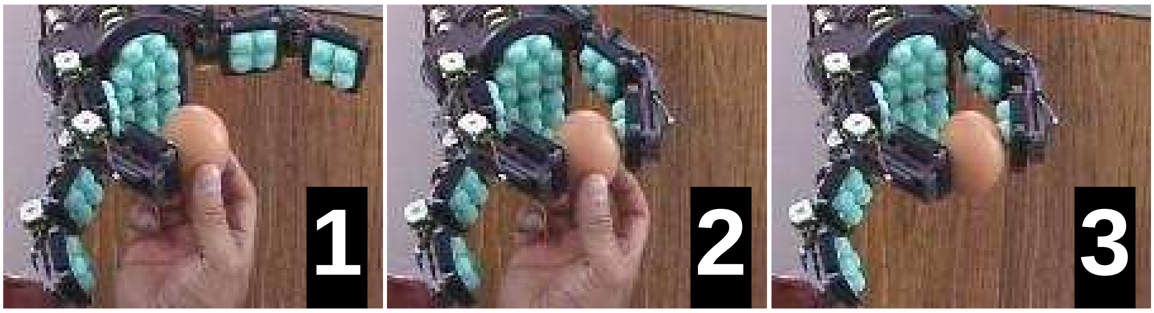

Figure 5-8: In this sequence we observe the hand closing on a egg. This task is easier than the one shown in figure 5-9. If we observe carefully 3. we can notice the properties of the tactile sensors. They detect the contact but also conform to the object increasing friction and making an stable grasp possible.

that the cylinder is only slightly deformed. On the other hand, we can observe in figure 5-10 that the cylinder is deformed when no tactile feedback is used.

The sensitivity of our compliant tactile sensor is in general high because of its structural design. We can think of the overall shape of the sensor as an hemisphere, therefore, when the sensor first comes in contact with a locally flat surface only a point -in theory- is touched. The stress in that point is high causing the structure to deform making detection possible. This feature, allows the sensor to work well with most surfaces.

In contrast, the flat rigid surface of the sensor previously mentioned [71] distributes the force applied. For example, if this sensor comes in contact with a flat surface of the same size as the sensor's surface, a large force will be needed to detect the contact. The experiment with the paper cylinder has little likelihood to work with this sensor, because the rigidity of its surface will deform the cylinder without detection.

The sensitivity of the sensor is within the range of human sensitivity. Human sensitivity is around 0.3 N to 1.0 N as mentioned in [71] and our tactile sensor's minimum detectable normal force is around 0.1 N in the magnetic version and 0.01 N in the optical.

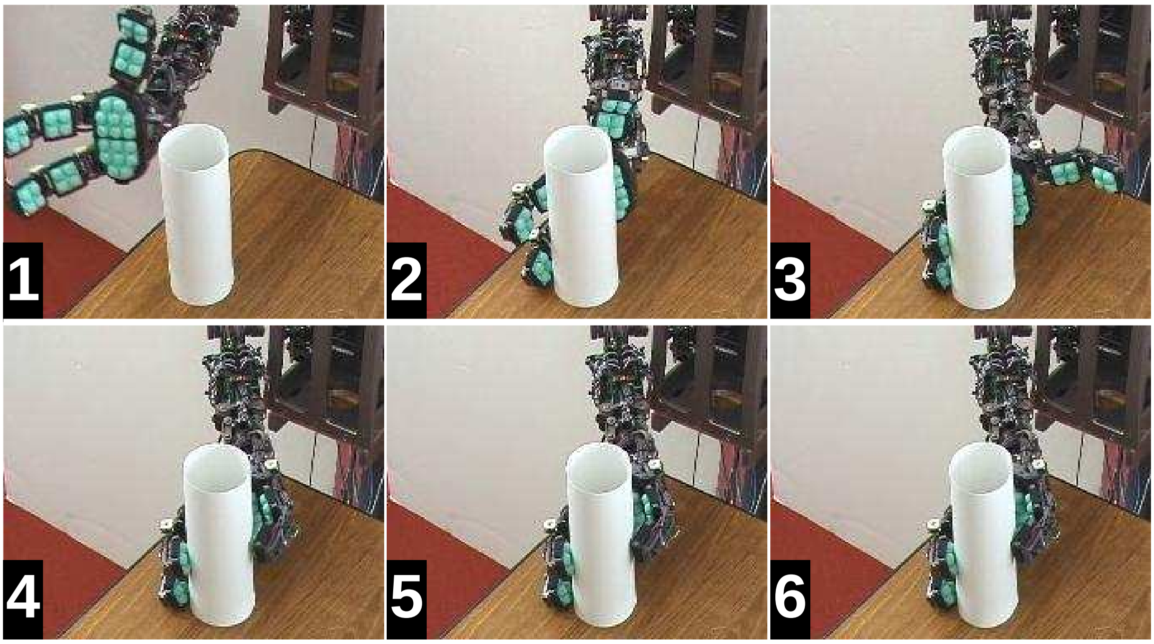

Figure 5-9: In this sequence we observe the robot approaching a cylinder made of paper bond. This cylinder has a weight in its bottom. The robot can detect the first contact, oppose its thumb and move the thumb, gently stopping when detecting the contact. The tactile sensors are sensitive enough to detect the cylinder without crushing it.

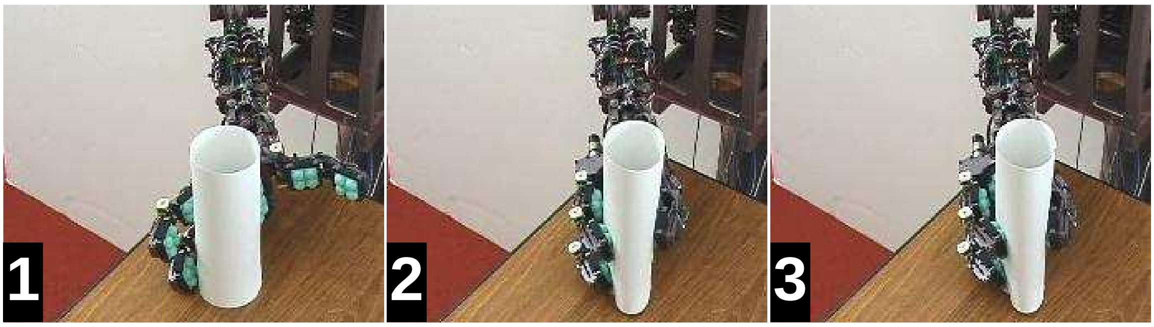

Figure 5-10: In this sequence the robot closes its fingers without using tactile feedback. As a consequence they crush the object.

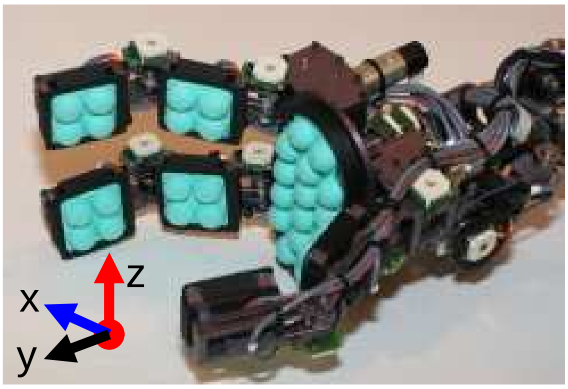

Figure 5-11: Axis used to reference the forces acting on the tactile sensors on the hand.

5.5.2 Interaction

We now turn our attention to the data produced by the compliant tactile sensors when interacting with an object. We can observe a sample of this in figure 5-12. In this figure we observe the trajectory followed by the arm in two different grasping sequences. The trajectory is represented by the angles of the elbow, wrist, and shoulder. The angle of the shoulder shown is the one that moves the arm up.

These differences between the trajectories are reflected in the tactile sensors' readings as shown in the top plot of figure 5-12. In this plot, the magnitude of the tactile vector is the vectorial summation of the forces exerted in each tactile sensor considering the current geometry of the hand.

We can see that the magnitude increases when the fingers are closed around the object. In the first trial, the magnitude remains quite constant while in the second there is a greater change due to the different trajectory of the elbow and the shoulder.

A more detailed plot of the response of the tactile sensors is presented in figure 5-13. In this figure we can observe that the Z and the Y components change as the arm follows a trajectory. The Z component of the second trial (in green)is smaller than the first because the object is in contact with the hand in regions with no tactile sensors. Finally, the X component does not go to zero mainly because the compliant sensors are not calibrated. However, we do not rely on this condition to evaluate the grasp.

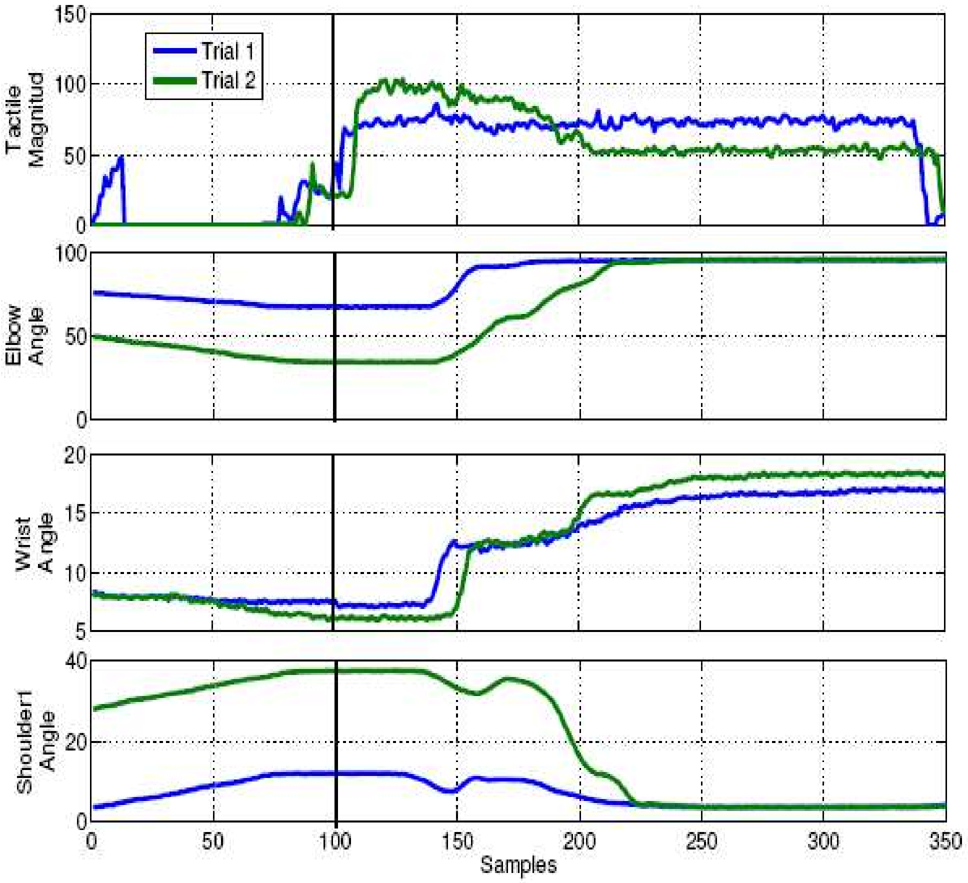

Figure 5-12: The figure presents the trajectories followed by two different grasping sequences of a same object. We can observe that the tactile sensors' response capture the variation in the different trajectories. The magnitude of the tactile vector plotted is the Z component of the vectorial summation of the forces present in each tactile sensor, considering the current geometry of the hand. The reference used is presented in figure 5-11. The vertical line indicates the moment that the fingers close.

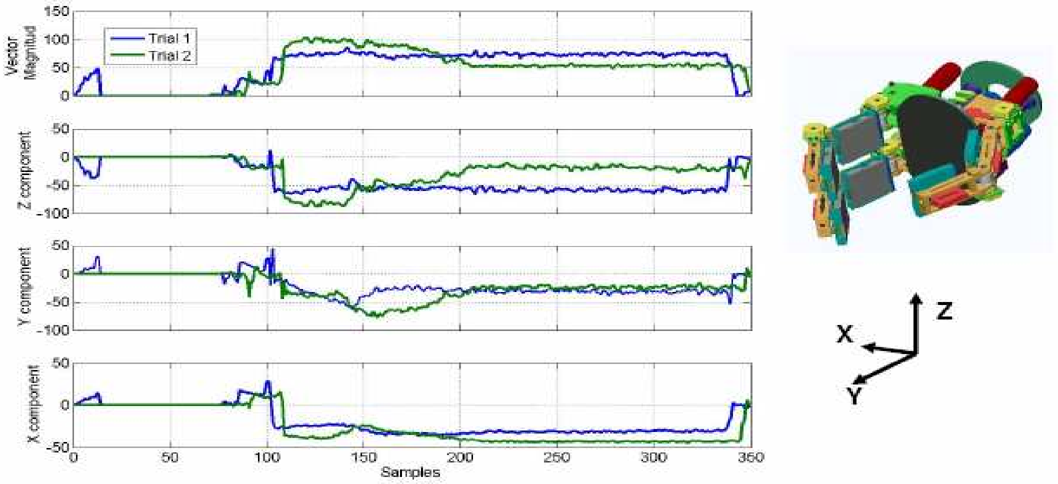

Figure 5-13: Components and summation of the tactile forces. The dimension of the vertical axes are A/D values. Two different grasping sequences are shown.

Further analysis of this information allows us to detect events such as slippage or contact of the object with an external element.

5.5.3 Slippage

In this section we present our implementation of slippage detection using the compliant sensors. Although controlling slippage can be useful in several manipulation tasks, we focus on avoiding slippage when lifting.

Sensing slippage is not an straight-forward task since the phenomena is not quite understood. If we observe humans, our skin has innervations that respond to fast motions on the surface. These innervations are the Meissner's corpuscles. They are specialized on detecting vibrations. Those vibrations are, in principle, related to the catch and release effect caused by an object sliding on the surface of the skin. This has inspired other researchers to build tactile sensors with the capability of detecting vibrations using accelerometers [42].

In our case we do not rely on this principle. Instead, we measure the change of the force applied on the direction of the lifting.

We illustrate this principle on the following experiment. We use the the cylindrical bottle ($mass = 0.179Kg$, $diameter = 92mm$, $height = 216mm$) shown in figure 5-14. The robot approaches the bottle by the side, touches it and subsequently rotates the thumb to grab the object. Since it is the first time it works with the bottle, it closes its thumb and index finger gently until contact with the object is detected. At this point, it closes the fingers more to deflect the sensors. This is necessary in order to obtain reliable readings from forces applied in directions parallel to the base of the sensors. In figure 5-11 we can observe the direction of the forces on the tactile sensors. In this experiment, we use only the tactile sensors on the distal phalanges of the fingers. The force normal to the tactile sensor surface is in the X direction, and the ones parallel are in the plane Y-Z.

When the fingers are closed on the object, the forces in X increase. The forces in Z (on the direction of gravity) and the forces in Y also change due to the deformation of the sensors. This deformation depends, at this point, on the angle of incident between the tactile sensor and the surface of the object and not on the weight.

When the robot starts moving its hand upwards, that is indicated by the increment of elbow angle figure 5-15, the force in Z increases in magnitude because the weight of the object pulls the sensors down. Assuming that the motion is linearly upwards and that the object does not rotate, the force should remain somewhat stable. The exception is at the beginning of the lifting where the dynamics of the motion are noticeable. The event just described is illustrated on the right side of figure 5-15.

On the left side of figure 5-15, we see that the behavior is different. The total force goes back to zero while the fingers are still closed. This is because the object slipped from the hand. A sequence of images of the slippage is shown in figure 5-14. It is important to notice in this figure that the bottle moves downwards but also towards the front. That is, the bottle rotates on the finger tips of the hand.

If we now return to figure 5-15, we can see the dynamics of the sensors/object interaction. After the point B, in the figure, the total force becomes negative because the friction with the object pulls the sensors down. This event is indicated by an increment on the elbow angle. While the total force is negative, there is a slight

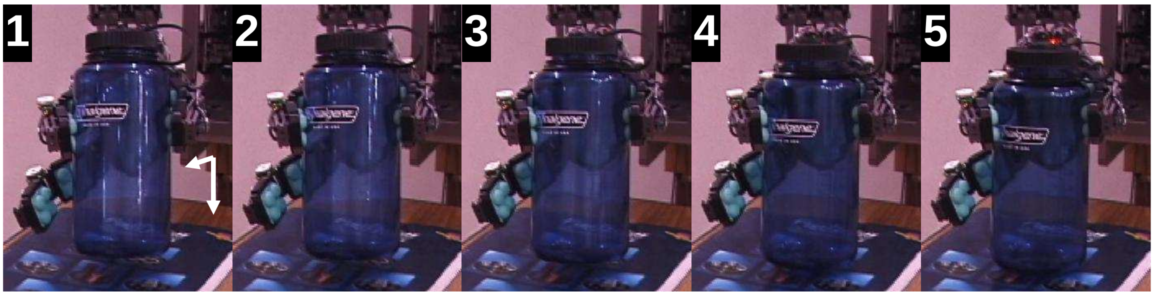

Figure 5-14: From 1 to 5 we observe the bottle slipping. The bottle moves downwards and towards the front. The motion towards the front is not linear but circular.

positive slope on the force that indicates some slippage. But more importantly, the angle between forces Z and Y in the thumb changes smoothly showing the rotational slippage. Subsequently, the object stops pulling down the tactile sensors when it touches the table. At point C in figure 5-15, the fingers are opened.

In order to detect slippage, we observe the total tactile force while the fingers are closed and the arm moves upwards. If we detect a positive slope, we conclude that there is slippage. We can observe this in figure 5-15. From point B to C there is a large positive slope. The slope is presented on the top plot for reference. From points E to F, there is a slight slip that stops and the grasp stabilizes.

The response time of this slip detector is not as fast as it woudl be required to adjust the force instantly. Instead, a new attempt to lift the object is done if slippage has been detected. The current limitation of these sensors is that computation is done off-board. However, the hardware on the sensors have the capability of doing this calculation faster.

It is important to remember that the data comes from a controlled experiment where the lifting is done only with the thumb and the index finger. Neither the palm nor the middle finger are used. In general, this robot is designed to do whole body grasping as opposed to precision grasping. In some cases, an object will come in contact with the fingers on spots where the fingers are not fully covered with the skin. In that case the slippage detection is not possible.

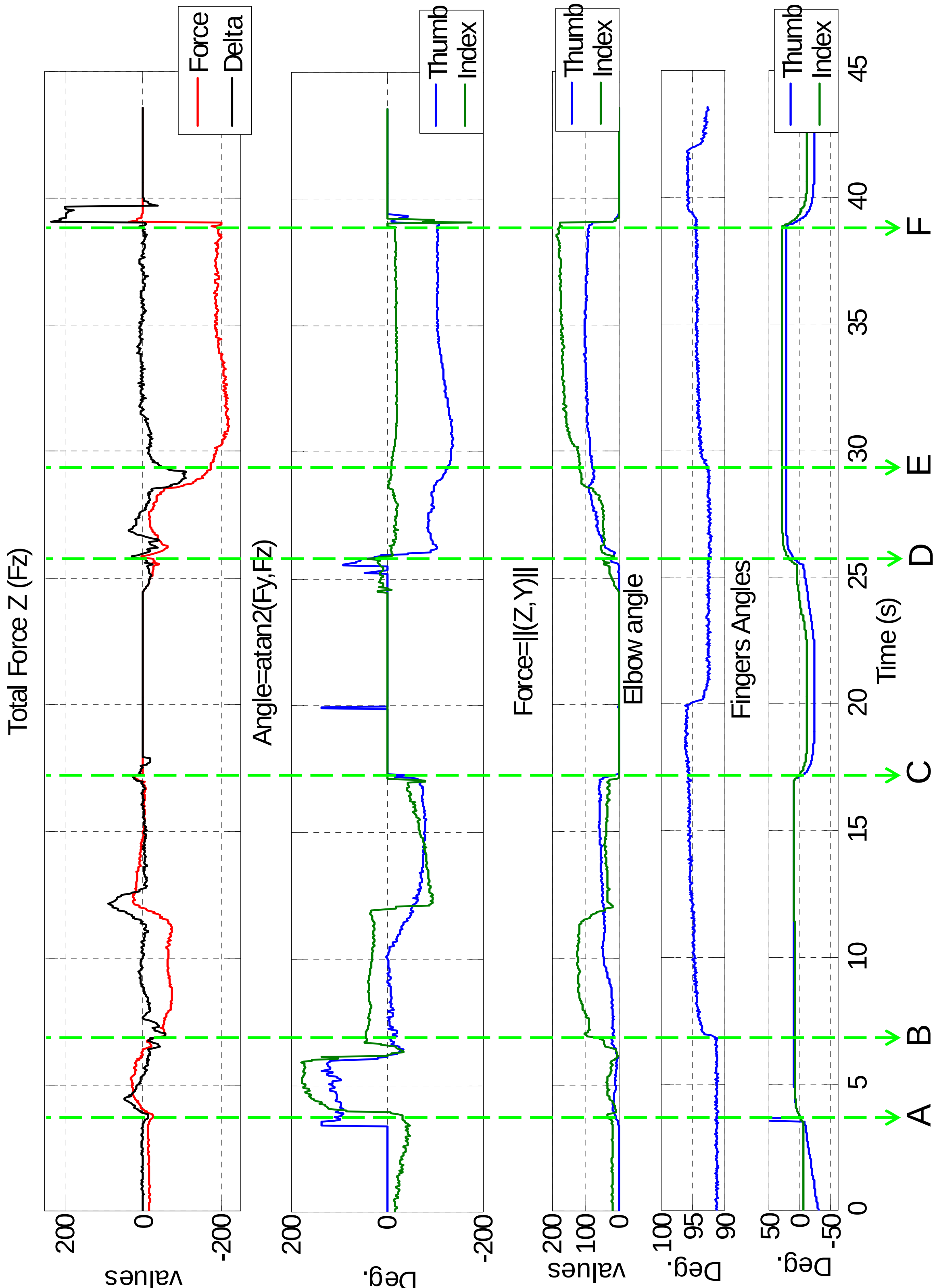

Figure 5-15: Tactile forces in two different lifting events. For details see section 5.5.3

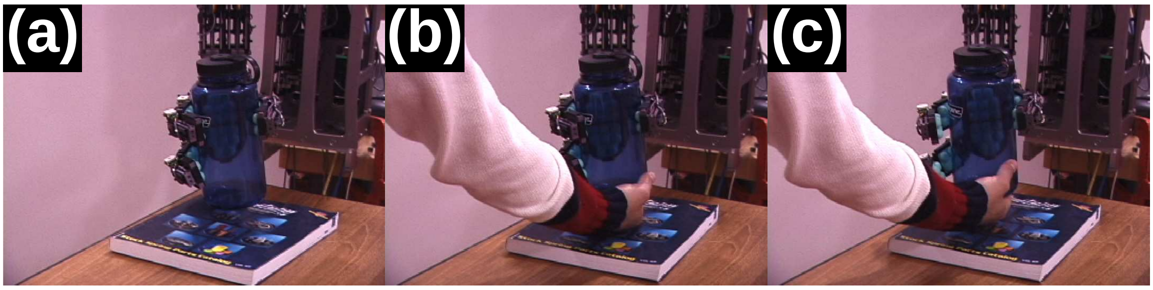

Figure 5-16: This sequence shows the robot detecting forces applied over an object when holding it. This is useful to place objects over surfaces. In (a) the robot is holding a bottle, in (b) a person is touching the bottle from below and the robot detects this force. As a consequence, the robot releases the object as shown in (c)

5.5.4 Applying forces to a held object: placing an object on a surface

Once the robot has an object in its hand, in order to interact with the environment it is important to detect changes of forces applied to the object. For instance, if the robot is holding an object and wants to place it on a flat surface, like a table, it is important to detect when they come in contact so an action can be taken. Other examples where detecting external contact with a held object include: using a hand tool, moving a cup towards our mouth, etc.

OBRERO's tactile sensors are capable of measuring changes on the forces applied to the object. In order to illustrate the principle we present the following example. The robot grabs the bottle used in section 5.5.3. Subsequently,it lifts the bottle and then moves it back to the surface. The forces applied to the object are monitored using the tactile sensors. When a large change on the forces is detected in direction opposite to the motion, the robot assumes that the object came in contact with a surface and releases its grip. The robot is capable of detecting this change at any time. For example, if we place our hand under the bottle before the robot touches the surface, the robot will able to detect this change on the tactile forces. This example is presented in figure 5-16.

It is arguable that this detection can also be done by force sensors on the joints of the robot. However, this is generally less sensitive because the effect of the change on forces needs to be reflected on the joint. More importantly, a detailed model of the forces acting on the joints needs to be implemented in order to filter out the changes produced by other factors. Tactile sensors are closer to the hand/object interaction making them more sensitive and less complex at the moment of extracting the information.

Figure 5-17 shows the response of the tactile sensors to external forces on the object held. In this case, the robot is holding an object and a person pushes the object from the bottom. In order to see the response, we program the robot so that it does not release the object. The plot on the top of figure 5-17 shows the changes on the forces in the direction of gravity. We do not work with the absolute force but with the differences because we are interested in the changes on the force. The values showed have been filtered using a moving average filter of size 8 and the sampling period is 100ms. The bottom plot on figure 5-17 is the integration of the data on the top plot, and it gives an idea of the time evolution of the force applied. We can clearly observe that the force increased because the object was pushed twice and the forces applied had different intensity.

In order to detect the event of pushing or contacting the table setting a threshold is enough, as it is illustrated on the top plot of figure 5-17.

5.6 Feedback sampling frequency

In table 5.2, we observe that the feedback information is available at different frequencies. The low level motor control receives the information from the encoders and potentiometers at 500Hz. This frequency is adequate for force control in the series elastic actuator and for the position control of the limbs.

The tactile feedback is sent to a daemon in a linux node. The information is updated at about 55 Hz. However, the tactile sensors can detect events at higher frequencies using its local microcontroller. Force and/or position feedback from the hand arm and head are sent to the linux nodes from their respective low level controllers. However, they are only available at 12.5 Hz to the linux nodes because of

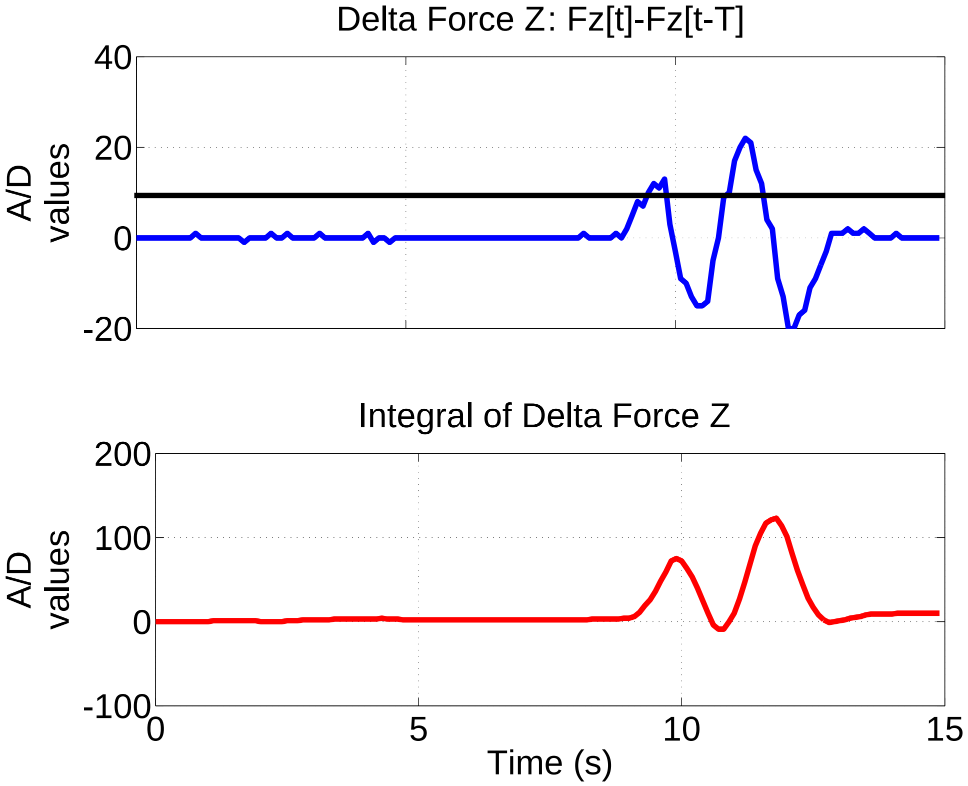

Figure 5-17: Pushing the bottle upwards twice. Top: Temporal difference of the total force in direction on the gravity when the hand is holding an object. See figure 5-11 for reference. Bottom: Integration of the data on the top plot. It shows the changes on force when the object was pushed.

communication delays.

This information shows that the behaviors implemented are constrained in speed by the feedback frequency. Therefore, the control architecture works at different speeds that is slow at the higher levels controllers and faster at the low levels. This structure is considered analog to the control loops in humans. A consequence of this structure is that the right sensing has to be used for a given task. For example, if we want to detect and avoid slippage, the sensing and the reaction have to be executed at the right level to work.

Table 5.2: Feedback sampling frequency

Name	Period (s)	Frequency (Hz)
Low level motor control	0.002	500.00
High level tactile feedback	0.018	55.55
High level position feedback	0.080	12.50
High level force feedback	0.080	12.50

5.7 Limitations of the implementation

The current implementation works well with objects whose height is greater than 0.120 m. This limitation can be removed by rotating the hand so that the index and middle finger end up positioned in top of the object. This movement also covers the case when an object is knocked over during the interaction. However, no all the objects can be handled with the current size of the hand.

The size of the fingers is a current limitation imposed by the original design which used larger tactile sensors. The size of the fingers and the coupling of their phalanges limits the size of the objects that can be manipulated. This limitation can be removed by building a smaller version. The current configuration of the hand is not adequate for all the cases. For instance, it would be useful to be capable of reorienting the palm.

The implementation also fails when the object comes in contact with the hand or arm in regions where there are not tactile sensors. This occurs even when the objects are large.

The spatial resolution of the tactile sensors also limits current implementation because some features of the objects are not differentiable. That is the case of the handle of a coffee cup. This could be solved with smaller version of the tactile sensors.

5.8 Revisiting the observations on humans

In chapter 4, we observed some of the capabilities of blindfolded subjects and used them as inspiration to program OBRERO. In this section, we will briefly compare the capabilities of the current implementation with the ones of the blindfolded subjects.

OBRERO is capable of exploring its environment and detecting contact with objects. It can reorient its hand to grab and lift an object. This is performed without a model of the object. Currently, it cannot deal with more complex cases such as an objects that are knocked over upon contact. It cannot reorient objects that have been grasped either. In other words, the robot has a grabbing capability that is basic compared to the ones exhibited by the subjects in the experiment.

It should also be noted that the robot cannot recognize the objects using tactile information. However, this does not affect the robot's capability of manipulate these objects. The same applies to the subjects in the experiment. That is, the capability of manipulating a given object does not depend on the capability of identifying it.

In this current implementation, the robot does not exploit the proprioceptive memory that is extractable from the position feedback of the arm and hand.

In short, we have given the first step toward a more flexible robotic manipulation using as inspiration human manipulation. There are certainly other aspects of these observations from which robotics could benefit.

5.9 Summary

We have described the design of a behavior that allows a humanoid robot to grasp objects without prior knowledge about their shape and location. We summarize here our approach and the lessons we learned:

- Give up precision, explore instead. Sometimes in robotics we struggle to have robots as precise as possible in performing the tasks for which we program them. We found that exploration can be more effective in dealing with uncertainties.

- Be soft. Exploration must be gentle if we want to avoid catastrophic effects on either the robot or the objects/environment. The mechanical design of the robot proved helpful in this respect.

- Sense and exploit the environment. If inquired, the world can provide useful feedback; however the robot must be able to ask the right questions (interact) and interpret the answers (have appropriate sensors).

We endowed the robot with the minimum capabilities required to explore the environment. These include a simple ability to detect visual motion, a way to control the arm to roughly reach for objects and a set of explorative primitives. Haptic feedback drives the exploration and allows the robot to successfully grasp objects on a table.

We further analyzed the tactile data obtained from the interaction with the objects and implement methods to detect slippage and external forces acting on an object being held. We also showed that the information generated can be potentially used to learn physical properties of objects like shape. This was possible in part because of the properties of the tactile sensors, whose advantages were further evaluated in this chapter.

In sensitive manipulation, we are interested in studying methods to improve the perceptual abilities of robots by exploiting the physical interaction with the environment. In this thesis, we have shown how haptic feedback can significantly improve this interaction thereby enhancing the robot's ability to learn about the environment.

Finally, it is worth saying that, to better illustrate our point, we deliberately took a somewhat extreme approach with respect to relying primarily upon tactile sensory information. We certainly believe that future robots will have to take advantage of the integration of all sensory modalities.

Chapter 6

Comparison

In this chapter, we review work related to robotic grasping and manipulation, and we contrast representative approaches with ours.

6.1 Robotic Grasping and manipulation

Extensive work has been done in the area of grasping objects using robotic hands. A detailed summary of the field can be found in [7] and [6]. An extensive modelling and analysis of different aspects of the mechanics of grasping has been developed. Those aspects include kinematics of the hands, modelling of contact, stability of grasping, robustness of grasp, and dynamics of the hand-object system.

Salisbury [89] has presented an analysis of the kinematics and forces in a hand. He starts by classifying the contact points between fingers and objects. The contact points can be modelled as: frictionless, frictional, or soft. A finger with a frictionless contact only exerts force on the direction of the normal to the object. If the contact is frictional, the finger exerts normal and tangential forces. In the case of a soft contact, the finger exerts a torsional torque in addition to the normal and tangential forces. Once the contact points are modelled the conditions of a stable grasping are defined. This equilibrium is known as force closure. A more constrained definition of stable grasping is known as form closure. Under form closure, the object grasped can resist external disturbances. A more extensive analysis of force and form closure can be found in [8].

Further modelling of contact points includes the analysis of kinematics and compliance. The kinematics of a contact point as two bodies move has been derived by [17] for planar bodies and by [76] for rigid spatial bodies with extensions by [90]. Compliance of the contacts has also been modelled in robotics; representative work is presented in [92], [24], [4]. However, the modelling of compliance has proven to be a more difficult task than the modelling of kinematics (See [7]).

Another aspect that has been considered in manipulation is the robustness of a grasp, which is defined as the ability to reject small disturbances from external forces and/or torques. The assumption made in this measure is that the fingers can be positioned accurately. Although this gives some idea of the performance of grasping, a better theory is needed.

In addition, the dynamics of the hand-object system has been analyzed. These analyses include the control laws of the hand/fingers and are mainly done in simulation. A major problem found is the inconsistency obtained when dynamics of the rigid bodies and the dynamics of the contacts are used together. Such is the case of "peg-in-the-hole" analysis by [108] and [107]. They showed cases where either no or two solutions for the accelerations were found. The inconsistency is attributed to the contact model used, i.e., Coulomb model. Therefore, efforts have been made to use a compliant contact model instead [64]. However, the analysis in simulation is still hard because of the great difference in time scale between the dynamics of contacts and that of rigid bodies.

All the work presented up to this point relies on the existence of models for both the object to manipulate and the fingers. Recently, there have been new approaches to deal with objects of unknown geometry. These approaches rely more on the information provided by the sensors than on the pre-existent model.

For example, in [84, 85] a generalized model of an object is used to grasp objects of unknown geometry. This model lets define potential functions with a unique minimum. The controller that guide the grasping are constructed to minimize these potential functions. The approach, known as control-based, combines controllers to generate tasks such as grasping. The controllers are combined in a hierarchical manner, using a nullspace projection to determine the region (in space state) in which the controllers do not interfere with each other. In the case of grasping, this approach assumes that the fingers are in contact with the object at the beginning of the task [56].

An aspect of manipulation that has been mostly neglected is exploration. That is using the manipulator to learn from the world as opposed to blindly acting on it. This has been mainly because of the way manipulators have been built and the sensing technology available. A starting point to this approach is [31] where a force controlled arm was used to explore its environment. The information obtained from this exploration was used to improve objet segmentation but for manipulation.

There is also a great amount of work in task and trajectory planning for manipulation, we will not review these topics because our approach is a different one. The reader is referred to [11].

6.2 Comparison

In this section, we contrast the approach presented in this thesis with current state of the art to give a better idea of its advantages and disadvantages. We have chosen the most representative approaches.

Much of the early research in manipulation uses a detailed model of the environment. Accurately modelling the environment can be a daunting task, if not impossible, hence many simplifications are done. Sometimes, these simplifications make the research miss the goal of the original problem. For instance, a finger is usually modelled as a point (the finger's tip) disregarding the effects of the rest of the finger, that in fact are important to grasp objects. The previous section reviews the analyses performed in this area. Currently, the results of these analyses have been integrated in simulators, such as GraspIt [75], which are used to implement model-based approaches.

As a current example, we can mention the work by Pelossof [83] where the finger's angles for an optimal grasp are determined given an object description. Support vector machines (SVM) are used to learn a function that maps finger's angles and object description to a defined measure of grasp quality. By maximizing this function, the goal finger's angles for a given object are computed. This work uses a model of a Barret [102] hand and it is implemented in a popular simulator known as GraspIt [75].

However, this approach makes assumptions that do not hold in environments outside the simulator. For instance, objects are modelled as superquadratic forms that can be characterized by two parameters. Moreover, the outputs of the maximization depend on those parameters. Outside the simulator objects do not have superquadrics forms only, unless we consider a particular environment. Besides, extracting the parameters of a superquadric form using the perceptual system of an actual robot will generally be a difficult task.

Further, the simulator uses punctual contacts to simulate friction which does not hold in most of the physical robots since their hands are padded with flexible material to increase friction.

Finally, the outputs of this method are the hand angles that maximize the grasp quality measure. But there is no information on how to get to that position except for the procedure described to acquire training data. This procedure will most likely not work in real robots because of the assumptions made. For example, the object does not move when it comes in contact with the hand, the hand approaches the object in a trajectory normal to the surface of the object, and the object is reachable from any direction.

Contrasting this approach with ours, we can point out the following differences. We do not assume a model for the objects and the position of the fingers are not determined by the output of a learned function. Instead, once they make contact, the fingers of the robot move around the object. This procedure does not depend on parameters from the model of the object and can adapt to the shape of different objects. There are no explicit models of friction to evaluate the grasp. Instead, parameters such as slippage are measured to determine a successful grasp. These measures also allow the robot to modify the forces exerted to improve its grip.

Finally, we consider the phase of approaching an object and coming in contact with it. In real robots perception is not perfect as it is the case in simulation. Therefore, our approach places the hand close to the object and then explores the space around it, in order to come in contact with it. However, coming in contact with an object and detecting this contact are not simple tasks to perform with physical robots as explained in this thesis.

Another approach is presented in [44],where a robot learns to select different grasps. They use a teleoperated and simulated humanoid robot to grasp objects. The simulation is run in a physical engine but the robot is moved slowly to minimize the dynamic effects. Grasp sequences are recorded and indexed by the object's shape. When a new object is presented, its shape is compared with the stored information using both a defined metric and geometric transformations. The grasp sequence whose index is closest to the current object's shape is selected. Trajectories are computed between the points of the sequence.

This approach also models the objects and has been tested in simulations, therefore, it has the disadvantages described previously. The type of objects used are also limited, the object's model is built by combining three basic primitives (boxes, cylinders and spheres) with their symmetry axes aligned.

As in many simulated approaches, we believe that the sensory input used is not plausible. For instance, a geometrical model of the objects to manipulate is available. However, its representation is successful at generalizing the grasps to different objects. It is also interesting to point out that the simulation provides them with many points of contact between the robot's body and the object. Out of those only one is used for learning. What is basically learned is the location of the hand with respect to the object.

Contrasting this approach with the one presented in this thesis, we can see that the differences are similar to the ones described earlier because this approach is also implemented in simulation, and relies on the model of an object. Also, this is a top-down approach that assumes perfect sensing and perfect actuation. Therefore, providing the location of the hand relative to the object is enough information for the low-level controllers to successfully grasp the object. Finally, this approach has not been tested in an actual robot. In contrast our approach provides the robot with basic skills to act in the real world. These skills are used to build more complex behaviors. Also, no perfect perception or actuation are assumed since this is an embodied system.

A different approach is presented in [84, 85]. This approach, known as control-based, combines controllers either sequentially or concurrently to generate the robot's behavior. An interesting aspect of this approach is the grasping controller developed [22, 56] to deal with objects of unknown geometry. This controller moves the robot's fingers around the object to minimize a grasp error function. This error is computed using feedback from load sensors on the distal phalanges of the fingers. The contacts are assumed to be non-frictional contact points. It is also assumed that initially the fingers make light contact with the object.

The error is defined as the summation of the magnitude of each force and moment applied to the object. Then two artificial potential functions with unique minima are defined. The first potential function is for the forces which considers the object to be a sphere. The second is for the torques which considers the object to be a plane. Two independent controllers are then used to minimize each of the potential functions. The controllers are combined using null space composition to avoid interfering with each other. Additional controllers can be combined to improve the grasp. For example controllers to enforce kinematic constrains [56].

The grasp controller has been combined with different controllers and with learning to perform tasks such as bagging objects. These tasks have been implemented in the robot Dexter.

Control-based approach uses a generalized model of objects that is useful in real objects. This model makes it possible to move the fingers around the object to grasp it. However, it only considers the finger as punctual entities. It is not clear how well it will perform when more than one point of the finger makes contact.

It should also be noted that a light initial contact is expected for the controller to run. Therefore, it is assumed that the approach to the object is done precisely enough to guarantee this condition. This implies that the robot needs to have some knowledge of the object a priori to locate the arm in the appropriate place.

In contrast, in our approach we explore the object in an attempt to increase the number of contacts on the hand. The contact locations are not punctual and frictional. We consider that approaching the object cannot always be done precisely. We compensate for this by exploring around the estimated position of the object until the robot makes contact. This can be done because of the low mechanical impedance of our robot.

We also like to mention the work presented in [26]. We consider this approach somewhat complementary to ours. The emphasis is in performing tasks once the object has been grasped. Therefore, hand-eye coordination has been developed and combined with the compliance of the robot's arms. Also, some bimanual coordination has been developed to transfer objects between hands or to hold boxes with both hands.

In contrast, in this thesis, the emphasis is on using tactile sensing combined with the compliance and force control in the hand. Consequently, OBRERO is capable of performing even delicate tasks such us picking up and exploring low mass objects. Vision is also used but as a complement.

In short, we believe that our approach considers the problem of grasping with a broader perspective. That is more phases of the problem are considered: reaching, grabbing, lifting, and landing. The object is not modelled and active perception is used to act. This allows us to generalize this approach to a large number of objects.

Chapter 7

Alternative methods

Some alternative methods developed to extract useful information from the environment are presented in this chapter. These include the use of sound and shadows. These work is also presented in [99] and [36] respectively.

In section 7.1, we describe an experiment where OBRERO find objects without prior knowledge of their presence or location, and then tap those objects with a finger. This behavior lets the robot generate and collect samples of the contact sound produced by impact with that object. We demonstrate the feasibility of recognizing objects by their sound.

In section 7.2, we show how using shadows or reflections of a robot's arm, time-to-contact to a surface and depth information can be estimated. These estimations work even when there is low texture in the images acquired by the camera. This is an important feature because it will be complementary to stereo vision which most likely will fail with low texture images.

7.1 Tapping into Touch

7.1.1 Introduction

Grasping and touch offer intimate access to objects and their properties. In previous work we have shown how object contact can aid in the development of haptic and visual perception [74, 78]. We now turn our attention to audition: developing perception of contact sounds. Hearing is complementary both to vision and touch during contact events. Unlike vision, hearing doesn't require line of sight – it won't be blocked by the arm, hand, or the object itself. And unlike touch, hearing doesn't require the robot to be the one causing the contact event. We are motivated by an experiment we report in chapter 2, where human subjects successfully grasped objects while blindfolded, using coarse tactile information and sound. The presentation of this work is organized as follows. Section 7.1.2 briefly reviews evidence for the importance of augmenting vision with other sensory input for manipulation in human infants and adults, and introduces the notion of *exploratory procedures* in humans and robots. Section 4.2 describes an experiment we carried out with human subjects with their senses interfered to try to simulate our robot. The experiment helps us to understand how humans would solve the kinds of problems with which our robot will be confronted. In section 7.1.3, we review our general developmental approach to robot perception, and then apply it to the problem of contact sounds. This motivates us to develop a robot behavior (described in Section 7.1.4) which gives the robot a way to actively probe the sound of objects in its environment in a robust way, by tapping them. Section 7.1.5 describes how the experience generated by the robot's behavior is exploited for learning. Section 7.1.6 quantifies the accuracy of object recognition enabled by this procedure. Finally, Section 7.1.7 discusses the results and places them on a broader perspective.

7.1.2 Background

The experiments reported in section 4.1 has inspired work on robotics. For instance, an analog of human sensitivity to thermal diffusivity was developed by [18], allowing a robot to distinguish metal (fast diffusion) from wood (slow diffusion). A robotic apparatus for tapping objects was developed by [87] to characterize sounds so as to generate more convincing contact in haptic interfaces. In [29], a special-purpose robot listens to sounds of the surface it "walks" on.

We use a *tapping* exploratory procedure, applied to natural objects by a general purpose, compliant hand (rather than a rigid, special purpose tapping device). The hand used is the one in OBRERO. Repetitive contact between the fingers and the object (the tapping behavior) allows the robot to collect information about the object itself (the sound produced by the collision of the fingers and the object surface) which is used for object recognition.

7.1.3 Overall developmental approach

We wish to give our robots many ways to learn about objects through action [35]. This contributes to perceptual development, where the robot's experience of the world is filtered by prior experience. This process can be broken down into four steps:

- Identification of an opportunity to reliably extract some object features
- Exploitation of that opportunity to extract those features.
- Use careful generalization to transform the robot's perception of its environment.
- Transformation of the robot's activity, enabled by its extended perceptual abilities.

In previous work, we have demonstrated this process. In [3], we showed that poking an object gives us the opportunity to reliably extract visual features of its appearance. By carefully choosing features that generalize, the robot's perception of its environment is transformed, and new activities are enabled [32]. Other opportunities we have explored include the use of grasping [80] and the integration of multi-modal cues across sound, vision, and proprioception [2, 33]. Having established this process, we are now seeking to broaden the range of opportunities that can be identified and exploited (steps 1 and 2 above). In the current work, we identify (and in fact create) an opportunity to reliably extract examples of contact sounds involving an object (by tapping that object). We build the appropriate robot behavior and data collection infrastructure to gather those features.

7.1.4 The robot's behavior

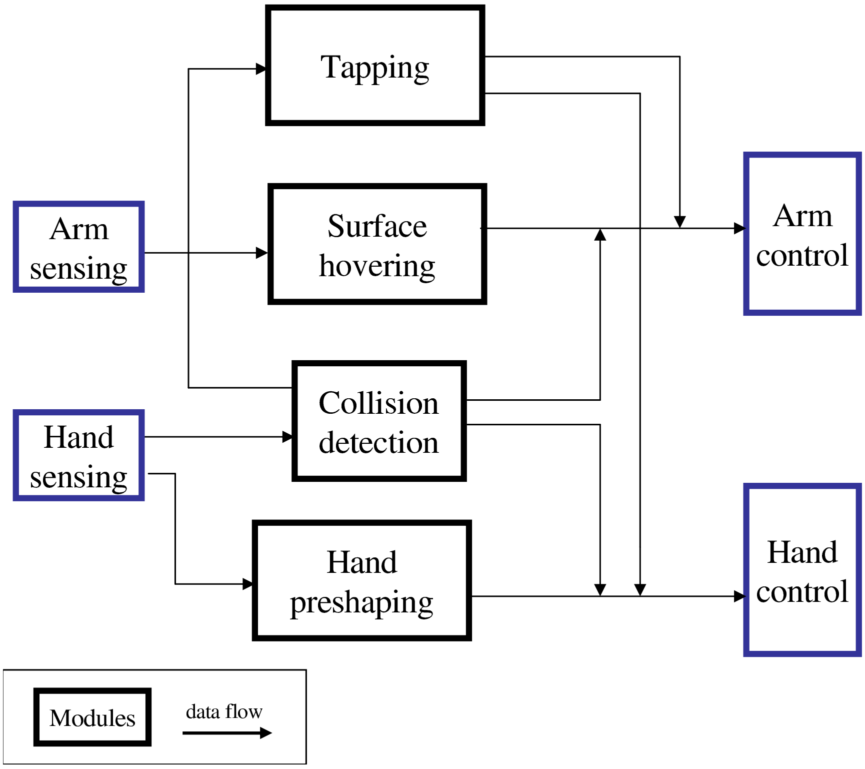

Figure 7-1: The component elements of the robot's behavior. The modules Arm control, Arm sensing, Hand control and Hand sensing represent the connection with the hardware of the robot.

The behavior of the robot is as follows. It sweeps its hand back and forth over a table, and stops to tap any object (or, indeed, any obstacle) it comes in contact with. This overall behavior is the product of the combined operation of a number of sub-behaviors, shown in Figure 7-1. Before we describe how they interact, here is a summary of these component behaviors:

- *Hand preshaping.* This module places the middle and index fingers together and perpendicular to the palm. The thumb is held up, perpendicular to the other two fingers. For preshaping, the fingers are controlled based on position rather than force.

- *Collision detection.* This module uses the outputs from the force and tactile sensors in each finger to determine whether a collision has occurred. This is possible because the hand has very low mechanical impedance and consequently the fingers slightly bend upon contact with an object. This bending is detected by the force sensor, often before the force exerted by the finger has greatly affected the object.

- *Surface hovering.* This behavior hovers the arm and hand over a surface using a predetermined fixed action pattern. The motion can be interrupted at any time.

- *Tapping.* This behavior moves the fingers back and forward for a given time, in another fixed action pattern.

- *Arm control.* This module deals directly with the low level motor control of the arm. The arm, for the work described in this paper, uses position control for each of the joints. To produce motion, a smooth trajectory is interpolated between setpoints.

- *Hand control.* This module provides a connection with the low level controller of the hand. It allows control of parameters such as the gain and the type of controllers, i.e. position and force control.

- *Arm sensing.* This modules reads the force and position measurements from the low level controller for the arm.

- *Hand sensing.* This module reads the force, position and tactile measurements from the low level controller for the hand.

The interaction of these parts is as follows. The *hand preshaping* and *surface hovering* modules make the arm and hand sweep over the surface with the middle and index finger extended forward and the thumb up. This is done by sending commands to the *arm control* and *hand control* modules.

When the fingers of the robot come in contact with an object, the *collision detection* module overrides the messages coming from *hand preshaping* and *surface* *hovering* to the *arm control* and *hand control* modules, commanding the arm to an immediate stop.

At the same time the behavior *tapping* sends commands to the *hand control* module to periodically touch the object and to the *arm control* module to keep the arm in position. The tapping lasts a few seconds, after which the *tapping* module relinquishes the control and stop sending commands. At this point the *surface hovering* and *preshaping hand* modules can get their message across to the motor control modules. Consequently, the arm is repositioned and the sweeping behavior reactivated.

These modules run on different machines on the network of computers that control Obrero. The interconnection between modules was done using `YARP` [34].

During the experiment we recorded vision and sound from the head along with the force feedback from both the arm and hand. The visual feedback was *not used* in the robot's behavior; it was simply recorded to aid analysis and presentation of results. All other sensory information were considered candidates for detecting contact. The force feedback from the hand proved the simplest to work with. Peaks in the hand force feedback were successfully employed to detect the impact of the fingers with the object during both the exploration and tapping behaviors. Force and sound were aligned as shown in Figure 7-2. Once the duration of a tapping episode was determined, a spectrogram for the sounds during that period was generated as shown in Figure 7-3. The overall contact sound was represented directly as the relative distribution of frequencies at three discrete time intervals after each tap, to capture both characteristic resonances, and decay rates. The distributions were pooled across all the taps in a single episode, and averaged. Recognition is performed by transforming these distributions into significance measures (how far frequency levels differ from the mean across all tapping episodes) and then using histogram comparison.

7.1.5 Data Collection for learning

The robot's behaviors are designed to create opportunities for learning, by finding and tapping objects. The modules that exploit these opportunities for learning are entirely from the modules that control the behavior of the robot. The occurrence of

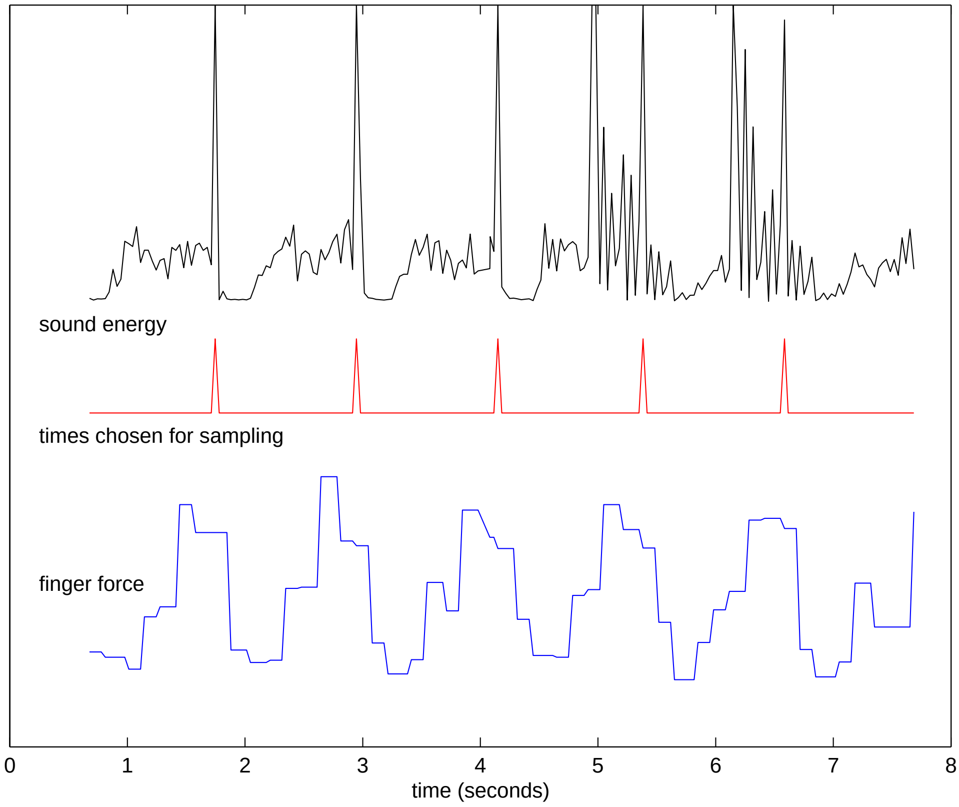

Figure 7-2: Force readings from the fingers (bottom) reveal when tapping may occur. Swings in the force are compared against sound intensity (top), looking for synchronized sounds. Peaks within one fifth of a period from a force swing are accepted. This process lets the robot filter out environmental sounds that occur when the arm is not moving, and even during tapping. In this example, the first three peaks of sound are clean, but the last two are corrupted by a phone ringing (see Figure 7-3).

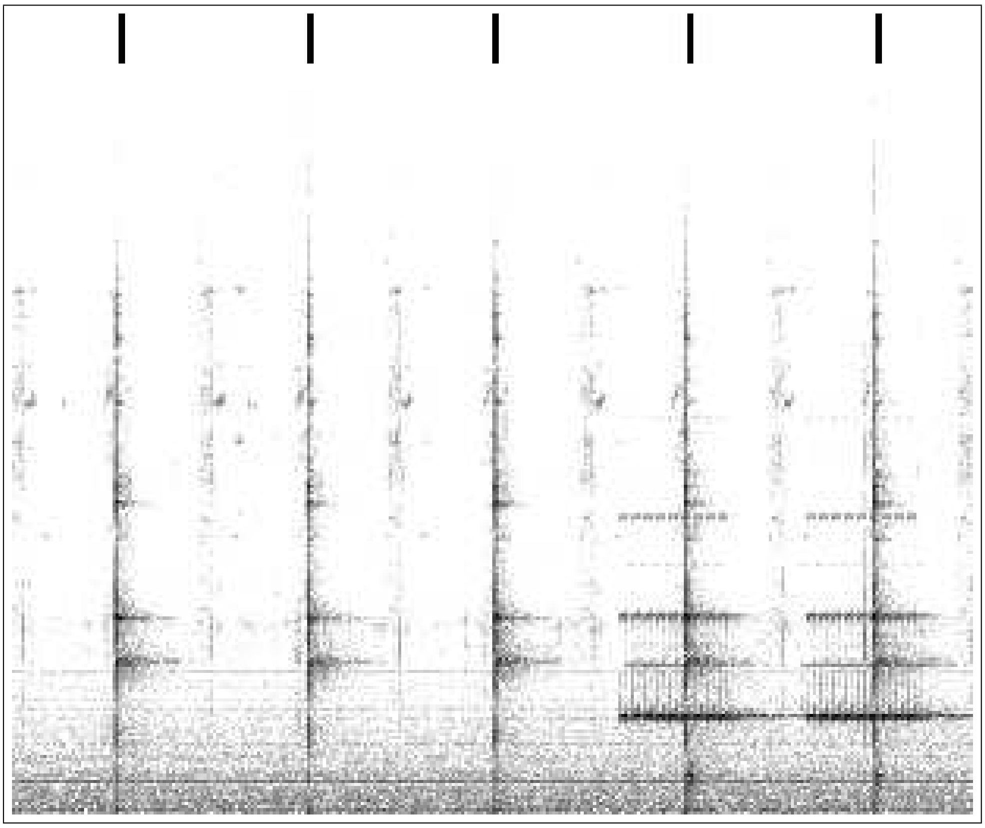

Figure 7-3: This is the spectrogram of the sounds in Figure 7-2 (time on the x-axis, increasing frequency on the y-axis, dark color corresponds to activity). The top of the spectrogram is marked to show the five sample times selected automatically. Between these times, there are patches of sound corresponding to the sound of springs in the fingers. The last two samples have the sound of a phone superimposed on them.

tapping is detected based on sensor data, rather than commanded motion. The only interaction that takes place between these modules is via actions in the world [12]. This improves robustness. We do not have to deal with explicit expectations or their possible failure modes. For example, sometimes the robot fails to hit an object when tapping, so it is good to pay more attention to actual contact rather than commanded motions.

The force measurements from the fingers is summed into a single signal, then classified into "rising", "falling", and "neutral" phases. Classification transitions to "rising" if the signal increases over 10% of the previous range covered by the signal from its highest to lowest point during a rising and falling period. Similarly, the classification transitions to "falling" if the signal falls by over 10% of this range. Since the range is constantly updated, the classification is robust to slow-changing offsets, and the actual gross magnitude of swings. The classifications are scanned for rhythmic rising and falling with a period lying between 0.2 and 2 seconds. Then the force signal in these regions is compared with the sound, to find if peaks in the sound line up well (within 20% of a period) of either peaks or troughs in the force signal (the sign depends on the orientation of the fingers during tapping). All going well, a spectrogram of the sound is performed in the appropriate range. Only the spectrogram around the peaks (presumably from tapping) is significant. Three samples are made in quick succession after each peak, to capture not just characteristic resonance but decay properties.

The robot's learning is performed on-line, but not in real-time. Performing data collection and learning in real-time on a robot can lead to research time wasted optimizing code and iterating designs that are otherwise adequate. But simply switching to off-line performance is undesirable, since it offers too many subtle ways for human input to enter the process. Hence we divided the robot's on-line system into two parts, the *real-time* subsystem that controls behavior, and the *near-time* subsystem that continually processes the robot's experience. This follows the design of the robot Cog's object recognition system [31].

Figure 7-4 shows the time course of an experiment. The key property being

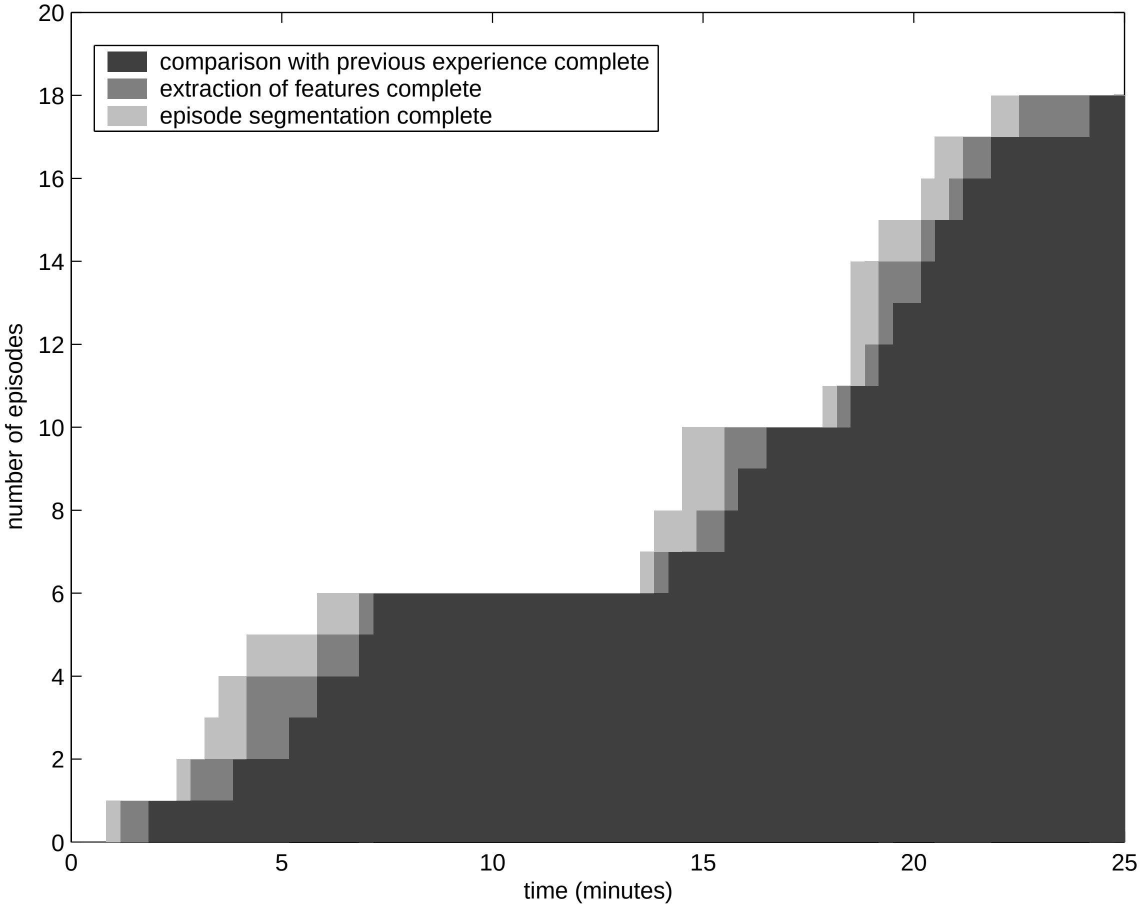

Figure 7-4: Time course of an experiment, showing the aggregation of experience by the robot. Over this 25 minute interval, 18 tapping episodes are detected. The episodes are first segmented (light color), then analyzed to extract characteristic features of the sound (darker color), then finally compared with previous episodes (darkest color). This process is online but unhurried – each episode can take on the order of minutes to be completely processed. In the meantime, the robot can continue with its normal behavior unimpeded.

illustrated is that the processing of the robot's experience happens at a relatively leisurely pace. This is workable as long as the processing can keep ahead of incoming data. For our robot, a complete rotating log of the robot's sensory input is made that covers about 30 minutes. Technically, this is achieved using a modified version of the open-source tool `dvgrab` for recording from a camcorder, and simple text files for other (much lower bandwidth) proprioceptive and summary data. The logs are maintained on a separate computer from the one controlling the robot's bahavior. These logs are processed using the open-source MATLAB-clone `octave`.

7.1.6 Results

We evaluated our work by performing an object recognition experiment. We exposed the robot one evening to a set of seven objects, and then in the morning tested its ability to recognize another set, which had an overlap of four objects with the training set.

Three of these objects were chosen (Figure 7-6) to represent three different materials, plastic, glass and steel (metal). The idea is that the sound produced by each object depends on its size, shape and the material with which it is made; accordingly we expected the tapping to produce three different distinct sounds. A fourth object (a plastic toy) was relatively silent.

For each run, we placed randomly selected objects on the table in front of the robot, and it was responsible for finding and tapping them. Overall the robot tapped 53 times; of these episodes 39 were successful, meaning that the sound produced by the tapping was significantly loud; in the other 14 cases the tapping did not provoke useful events either because the initial impact caused the object to fall, or the object remained too close to the hand. The high number of successful trials shows that given the mechanical design of the hand, haptic feedback was sufficient to control the interaction between the robot and the environment.

We evaluated the performance of our spectrum comparison method by ranking the strength of matches between episodes on the second day and episodes on the first day. Figure 7-5 shows what detection accuracy is possible as the acceptable false positive

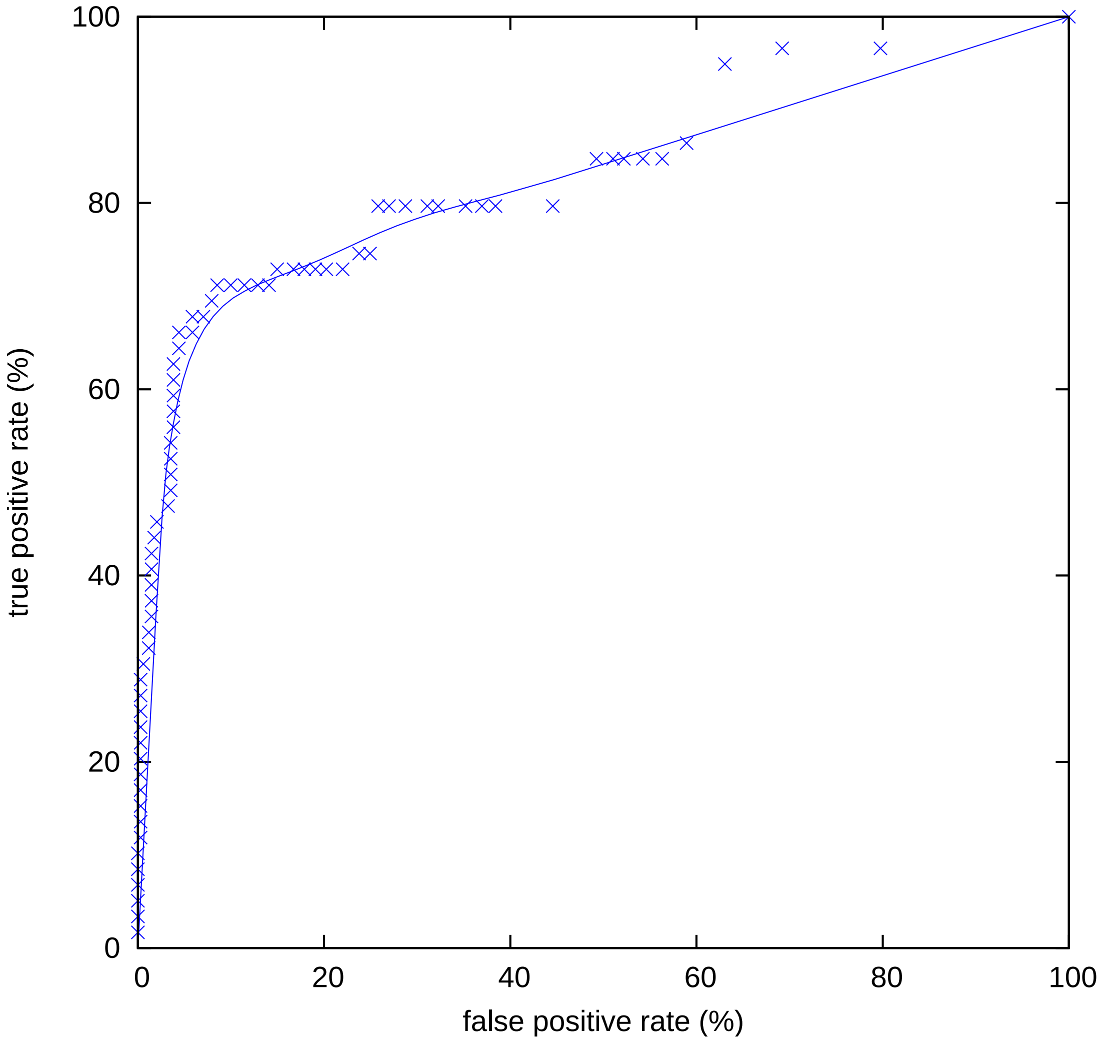

Figure 7-5: Receiver-operator characteristic curve. Tapping episodes from one day are matched against episodes from a previous day. Matches are ranked, then truncated based on a quality threshold. This plot shows the effect of that threshold on the trade-off between false matches and missed matches.

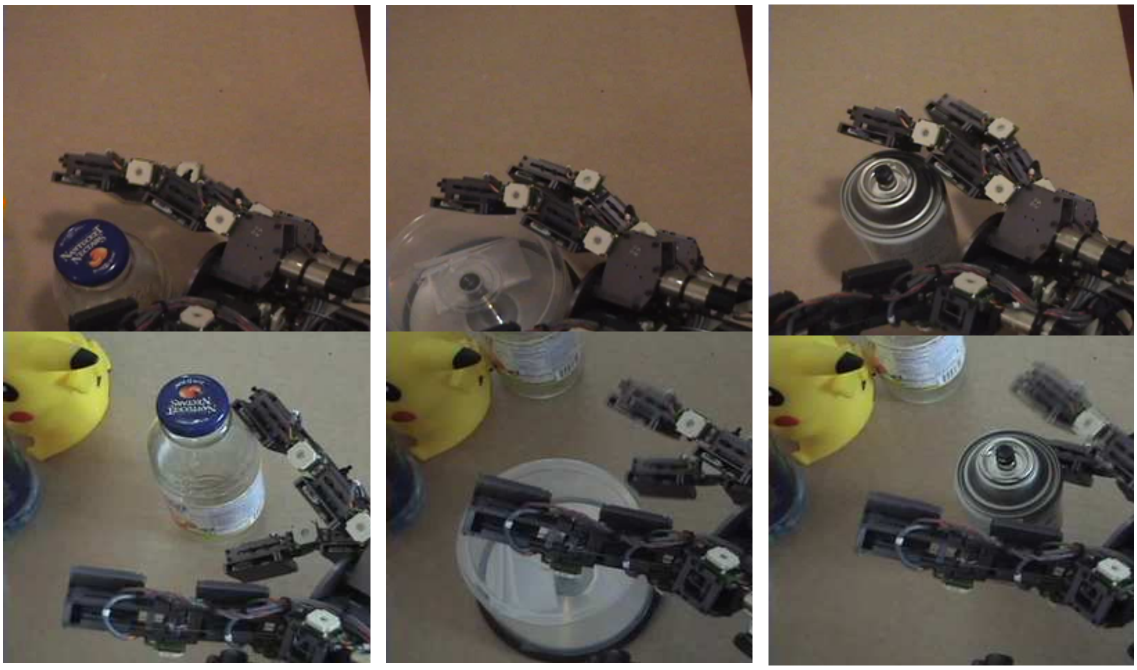

Figure 7-6: There were four objects in common between the training and test run. Three of them were matched perfectly (using a best-match rather than threshold-based strategy) for every episode: a bottle (left), a CD case (middle), and a spray-can (right). Images on the bottom are from the test run, images on the top are from the best matching episode in the training run. These objects have quite distinctive sounds. A plastic toy (left corner of each lower image) failed to be recognized – it was quiet, and made just a dull thud.

rate is varied. This predicts that we can on average correctly match an episode with 50% of previous episodes involving the same object if we are willing to accept 5% false matches.

7.1.7 Summary

We have demonstrated a compliant robot hand capable of safely coming into contact with a variety of objects without any prior knowledge of their presence or location – the safety is built into the mechanics and the low level control, rather than into careful trajectory planning and monitoring. We have shown that, once in contact with these objects, the robot can perform a useful exploratory procedure: tapping. The repetitive, redundant, cross-modal nature of tapping gives the robot an opportunity to reliably identify when the sound of contact with the object occurs, and to collect samples of that sound. We demonstrated the utility of this exploratory procedure for a simple object recognition scenario.

This work fits in with a broad theme of learning about objects through action that has motivated the authors' previous work [35]. We wish to build robots whose ability to perceive and act in the world is created through experience, and hence robust to environmental perturbation. The innate abilities we give our robots are not designed to accomplish the specific, practical, useful tasks which we (and our funders) would indeed like to see, since direct implementations of such behaviors are invariably very brittle; instead we concentrate on creating behaviors that give the robot robust opportunities for adapting and learning about its environment. Our gamble is that in the long run, we will be able to build a more stable house by building the ground floor first, rather than starting at the top.

7.2 The power of the dark side

7.2.1 Introduction

Shadows have long been a bane of computer vision, creating illusory objects and obscuring true object boundaries. Much energy has been devoted to developing the means to detect shadows automatically, not because they are considered valuable in themselves, but because they need to be distinguished from true objects [86]. On the other hand, researchers in computer graphics and virtual reality have struggled to render shadows in order to increase realism and aid depth perception [47]. Robotics draws on all these fields, so should we be fans or foes of shadows?

In this paper we observe that in visually-guided reaching, perception of the shadow(s) cast by a robot arm is potentially very useful. As the arm approaches a surface, the shadow that it casts rushes to meet it. This gives an indication of the arm's proximity to the surface. If we track this approach, we can estimate a "time-to-contact" for touching. This is analogous to the time-to-contact quantity derived from optic flow for navigation [38,69]. We demonstrate directing an arm to touch a target using information from a single camera. This is done by driving the arm to simultaneously reduce the visual error between the endpoint and the target and the visual error between the endpoint's shadow and the target. When both these errors are zero, the endpoint is at the target in 3D space. We use a tactile sensor in the arm's endpoint to verify contact.

Section 7.2.2 introduces Coco, the gorilla-like robot upon which this work is implemented (see Figure 7-7). Section 7.2.3 reviews in brief the literature on shadows and their role in human depth perception, making the case that they are important to the perception of contact and are an very strong cue, particularly when combined with motion. Section 7.2.4 discusses the particular method proposed for detecting cast shadows of the arm, using the fact that the robot is in control of the arm and can move it to its advantage. Section 7.2.5 uses the detected shadow as the basis for a controller, which drives the arm from point to point on a surface, rising away from and falling towards the surface at the beginning and end of the trajectory. Section 7.2.6

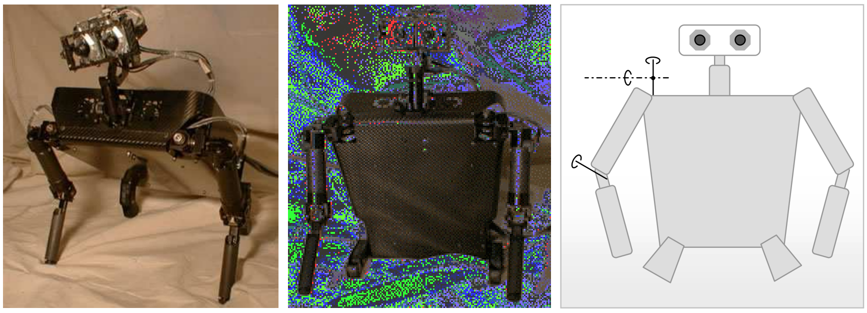

Figure 7-7: The gorilla-like robot Coco [77]. In this work, Coco was sitting up and using its right arm to touch surfaces. The arm has one 'elbow' hinge joint and two differential 'shoulder' joints (see Section 7.2.5). For this work, a single higher-resolution camera (720 × 480 pixels) was used rather than the pair of lower resolution cameras already installed.

looks more closely at estimating the time-to-contact from shadow information. We close with discussion and proposals for future work.

7.2.2 The robot

The platform used is the robot Coco designed by Morse [77] (see Figure 7-7). Coco is a gorilla-like robot, with two legs and two arms. Each leg has two degrees of freedom (DOF), one in the knee and one in the hip. Each arm has three DOF, one in the elbow and two in the shoulder. The neck has two DOF. This robot has a mass of 10 kg. Each arm is 312 mm long, the legs are 110 mm long and the body is 310 × 345 × 100 mm. The arm cam be used to walk and to interact with the environment. The motor control is performed by five Agile MAX2000 motor control boards. Each board drives 3 motors and is controlled across an RS232 serial port. In the tip of the arm there is a force sensor with a resolution of 120 mV/kg and a limit of 0.5 kg.

The vision system consists of a Sony camcorder with a resolution of 720 × 480 pixels and controllable zoom and focus. The control is done by a custom board that interfaces the LANC port of the camera to a RS232 port. The camera connects to an offboard computer running Linux through its IEEE 1394 (Firewire) interface. This computer runs the vision system described in the remainder of this paper, and communicates via TCP/IP with a second computer charged with motor control and interfacing with the motor control boards.

7.2.3 The role of shadows in human perception of depth

Many cues play a role in the human perception of depth, including occlusions, stereopsis, motion parallax, shadows (attached or cast), interreflections, perspective effects, etc. Over the centuries these cues have become better understood in art; the idea of perspective, for example, first gained currency in the 15^{th} century. In this century, the need to understand depth perception has grown in urgency as researchers in computer graphics and virtual reality strive for increasing realism. Stereopsis has proven to be an important cue; shadows and interreflections have also proven effective, and in some cases match stereopsis in power [45–48]. Moving shadows can profoundly influence the interpretation of a scene, in some cases overruling other depth cues such as apparent size [60, 72]. Apparent shadow motion can give a very strong sense of object motion even in the absence of any other cue [59]. Shadows help in the perception of spatial relationships in computer-generated images [109]. Shadows and interreflections are particularly important for giving a strong impression of object contact in computer graphics [97].

There is speculation that the cast shadow of the hand may be integrated into the body schema, in a manner somewhat analogous to the extension of the schema to include tools [82]. There is some evidence that the processing of shadows in the brain, which can impact on object recognition performance, is at least partially done without conscious awareness [19].

In computer vision, shadows cannot be ignored. Much work has been devoted to detecting them, even if only for the purposes of discounting them [94] (for a review, see Prati et al [86]). Less common is work to utilize shadows. In computer graphics, it was realized that the cast shadow of a known object (a long stick) could be used to recover the shape of an object the shadow passed over, in a manner analogous to (but cheaper than) active illumination methods that scan lines of light across the scene [10]. Shadow has also been used to recover body pose in video of moving people [14]. There is a great deal of work on 'shape from shadow' which is generally

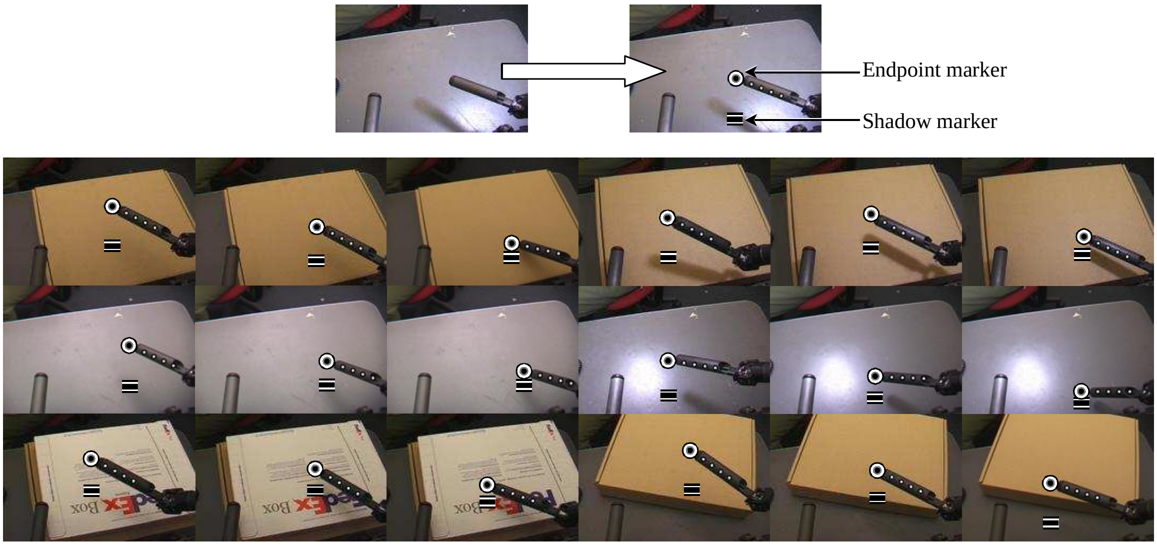

Figure 7-8: Detection of the arm endpoint (shown as a circle) and the arm's shadow (shown as a bar). The surfaces shown are a cardboard box, a metal table, and a text-covered box, all at different distances from the robot. Lighting varies from ambient (where the shadow is barely perceptible to the human eye) to strongly directional, from left or right.

concerned with shading rather than cast shadows.

To summarize, shadows have a demonstrable and significant impact on the ability of humans to perceive depth and adjacency, and to judge whether an object and a surface are in contact. We wish to import that advantage into robotics, rather than continue to rely only on stereo vision for depth perception. Shadows and stereopsis have somewhat complementary properties; stereo is at is best when depth changes are sharp, while shadows are easiest to track when depth changes are relatively smooth. The error in stereo measurements grows with distance from the camera, while the error in shadow measurements grows with distance from the surface. Shadows (and reflections) are detectable even in the absence of texture, or with reflective surfaces, situations that can confuse stereo. We believe that combining stereopsis and shadows could lead to a more robust system for manipulation. In this paper we seek to show the benefits of shadows on their own, but that should not be read as a belief that shadows are better than stereo.

7.2.4 Detecting cast shadows of the arm

What properties of a cast shadow can be used to reveal its presence? The most obvious property of a cast shadow is that the surface it is cast on becomes darker than it otherwise would be. In a single image, this is not very helpful, since we do not know how dark different surfaces should be, and the appearance of surfaces varies for many reasons other than shadows. In a video stream, this property is more helpful, since we can watch for moving patches of darkness, or compare against an image of the background without the arm or its shadow present.

For this work, two methods are used for detecting regions of illumination change, labelled *static* and *dynamic*. The static method is based on building a background model of the workspace while holding the arm to the side, then looking for novel regions of illumination change relative to that model when the arm swings into action. This method can detect very weak shadows, and operates just as well when the arm is stationary as when it is moving. The dynamic method specifically looks for moving regions of illumination change, by comparing the current image from the camera with one from half a second ago. This gives better localization of the shadow of the endpoint (the fastest moving part of the arm) while the robot is in motion, but is useless when the robot is stationary.

The first step in finding the cast shadow is to find the endpoint of the arm itself, as a reference. The endpoint of the arm is tracked at 30Hz. In each frame, a small window (1/25 of the image area) around the last known position of the endpoint is scanned for the dominant local orientation within ±5° of the arm's last known orientation. It is assumed this dominant orientation will be due to the edges of the arm and the highlights running along its length. The tracked position of the arm is then updated to the center of mass of the pixels that are at the dominant orientation, with a small constant offset in the direction of the orientation. When iterated, this procedure drives the tracked position out along the arm to the endpoint, and pulls it back in if it begins to drift. The tracking procedure makes minimal assumptions about the endpoint. The tracker is initialized using an appearance model of the arm

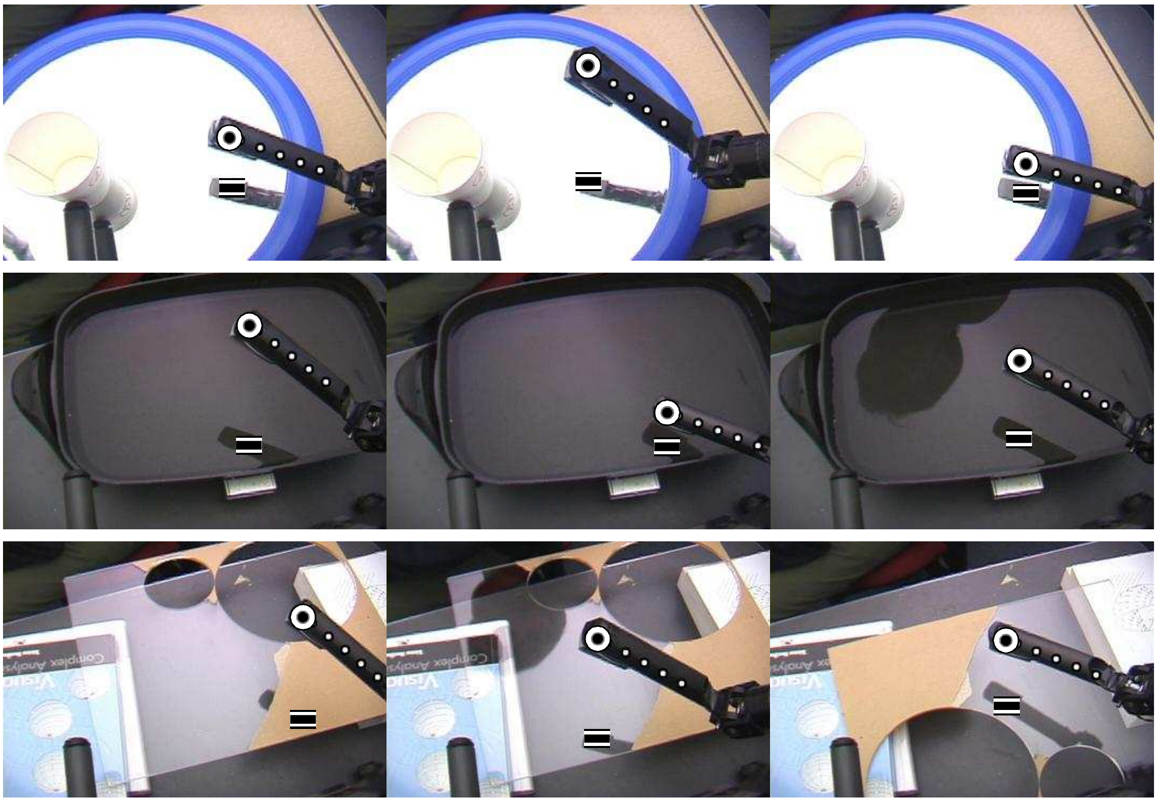

Figure 7-9: Another form of illumination change that can occur is a *reflection*. The method described for shadow detection works without change to detect reflections. The top row shows the reflection of the arm being detected in a small mirror. The middle row shows reflections in a dark bowl of water, and the bottom row shows reflections and shadows on a sheet of acrylic, partially covered.

and an assumption that the arm is initially in an area with good contrast with the background.

Once the position of the endpoint is known, the vision system waits for an opportunity to build a background model. This is provided by simply swinging the endpoint to the side, out of the way. The background model is simply a snapshot of the scene. When the arm swings back into action, static and dynamic shadow cues are combined by simple summation of grayscale pixel differences with the reference images. To speed shadow detection, we make some simplifying assumptions: we assume there will be a shadow below the arm in visual coordinates, and some component (even if very faint, and not the main shadow) will be vertically underneath the arm. Figure 7-8 shows a range of scenarios that meet this assumption. Given that assumption, we

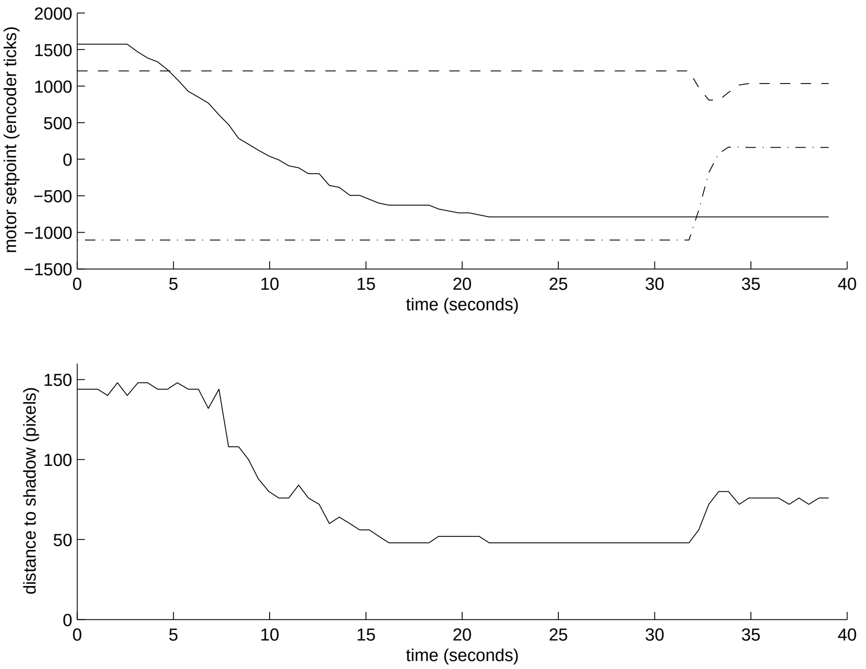

Figure 7-10: Shadow graph.

simply scan down from the endpoint, looking for a peak of illumination change. The shadow (or reflection – see Figure 7-9) we find will in general not be of the forearm. We could trace from this starting point to find the shadow, but choose not to do so, since the distance to this point from the endpoint is as good as any as a measure of shadow distance.

7.2.5 Reaching across a surface

Shadow detection allows us to design a controller for reaching across from one point to another on a surface. A control loop designed to maintain the distance of the cast shadow from the arm served to keep the arm at a constant 'cruising' height. At the beginning of the trajectory the arm rises away from the surface, as judged using shadows, to avoid scraping across the surface. At the end of the trajectory the arm sinks towards the surface (see Figure 7-11).

Which motor combinations move towards the surface? This just needs to be known qualitatively, and doesn't need to be exact – errors can be absorbed into the closed loop visual errors. A very crude kinematic model for the workspace can be

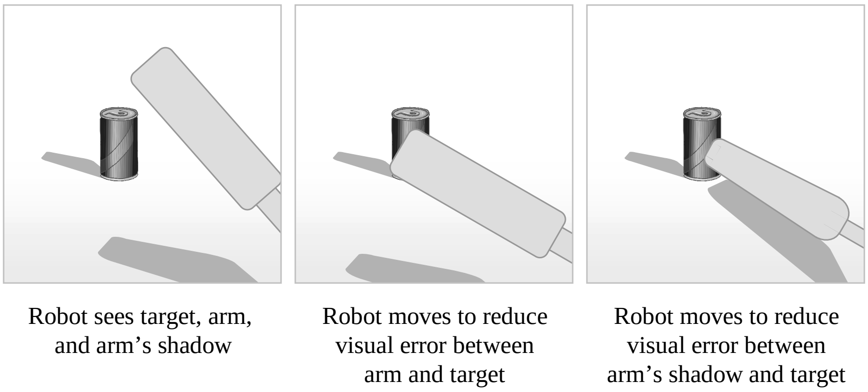
Robot sees target, arm, and arm's shadow
Robot moves to reduce visual error between arm and target
Robot moves to reduce visual error between arm's shadow and target

Figure 7-11: Given a visual target (e.g. the can shown) we can define a 2D endpoint error and a 2D shadow error, based on the visual offset between the target and the endpoint, and the visual offset between the target and the shadow. If both visual errors are driven to zero, the endpoint is at the target. Either error individually is insufficient to drive the arm to a 3D target.

constructed by making small exploratory movements and measuring the amount of change caused in 2D endpoint location and shadow distance by displacements in each of the degrees of freedom. Coco's arm has three degrees of freedom – a hinge joint at the elbow (θ_1) and a differential pair at the shoulder (θ_2 and θ_3). To control the arm, virtual axes are used that correspond very roughly with extension away from the body (m_1) rising towards the head (m_2) and drawing inwards towards the chest (m_3), which are related to the physical axes as follows:

$$\begin{aligned}\theta_1 &= m_1\\ \theta_2 &= m_1 + m_2 + m_3\\ \theta_3 &= m_1 - m_2 + m_3\end{aligned}$$

Drawing inwards towards the chest (m_3) is achieved by driving the differential shoulder motors θ_2 and θ_3 in lock-step. Rising towards the head (m_2) is similar, but now the differential motors are driven in opposition. Extension of the arm (m_1) is achieved by moving the elbow joint, then compensating at the shoulder by the same angle to keep the endpoint moving out along a ray from the shoulder.

We wish to be able to move the endpoint of the arm, detected in visual coordinates, towards a target also defined in visual coordinates, while controlling the height of the arm above a surface (estimated from the behavior of the arm's shadow). When the robot's gaze is directed downwards at a workspace, the virtual axes correspond very approximately to variables we would like to control. Changing m_2 (rotation towards the head) is approximately what we need to control depth, whereas m_1 and m_3 will have a big impact on moving the position of the arm across the image while having a lesser impact on its depth.

Since we can also recover the projected angle of the arm in the image, which for the geometry of this robot is an approximate indicator of rotation towards the chest (m_3), desired visual displacements can be expressed in terms of desired visual rotation and extension of the arm. These quantities can be controlled coarsely using m_2 and m_3. We label the three visual control quantities as v_1 (desired visual extension of the arm), v_2 (desired visual distance to shadow), and v_3 (desired visual rotation of the arm). To discover the relationship between the observed variables and the control variables, the robot makes exploratory movements that start small, then grow until it sees significant changes in at least one of the observed variables. The result of this exploration will be somewhat dependent on the geometry of the surface and the location of the arm – here is the result of a typical run:

$$\begin{bmatrix} \Delta v_1 \\ \Delta v_2 \\ \Delta v_3 \end{bmatrix} = \begin{bmatrix} 8.1 & 6.3 & -4.4 \\ 8.4 & 10.8 & -7.1 \\ 3.9 & 1.9 & 11.3 \end{bmatrix} \begin{bmatrix} \Delta m_1 \\ \Delta m_2 \\ \Delta m_3 \end{bmatrix}$$

Note that Δm_1 has its strongest influence on Δv_1, Δm_2 has its strongest influence on Δv_2, and Δm_3 has its strongest influence on Δv_3, which corresponds to the general intuition. This matrix is sometimes called the image Jacobian [110]. For our visual observations, which are of the same rank as the control space, it is particularly easy to invert the matrix and produce a reasonable controller:

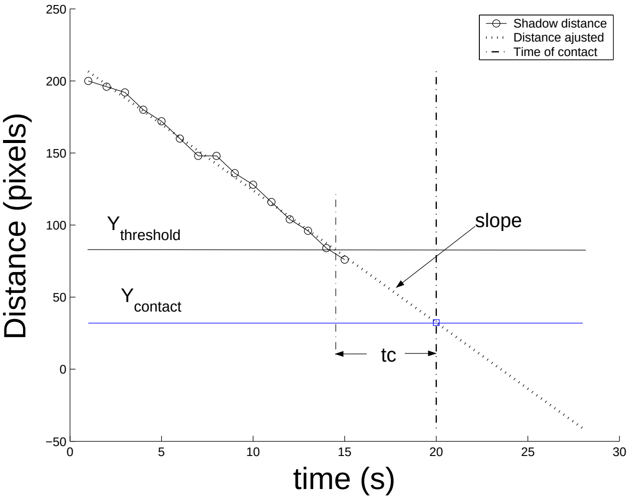

Figure 7-12: This figure depicts the quantities used to estimate the time to contact.

$$\begin{bmatrix} \Delta m_1 \\ \Delta m_2 \\ \Delta m_3 \end{bmatrix} = \begin{bmatrix} 0.31 & -0.18 & 0.01 \\ -0.28 & 0.25 & 0.05 \\ -0.06 & 0.02 & 0.08 \end{bmatrix} \begin{bmatrix} \Delta v_1 \\ \Delta v_2 \\ \Delta v_3 \end{bmatrix}$$

7.2.6 Estimating time to contact

With the current platform, we had the robot in front of a flat surface where the arm can be moved. The vision system detects the orientation of the arm and its end point. The shadow projected by the arm over the surface is also detected (see figure 7-14). The distance between the shadow and the end point in the image captured by the vision system is used to determine how close is the arm from the surface.

In this section we evaluate and quantify the accuracy of shadow detection for predicting the time to contact as the robot arm approaches a surface. The principle used to estimate the time to contact is depicted in figure 7-12. We can estimate the time to contact using the following equation.

$$tc = \frac{(Y_{contact} - Y_{threshold})}{slope} \tag{7.1}$$

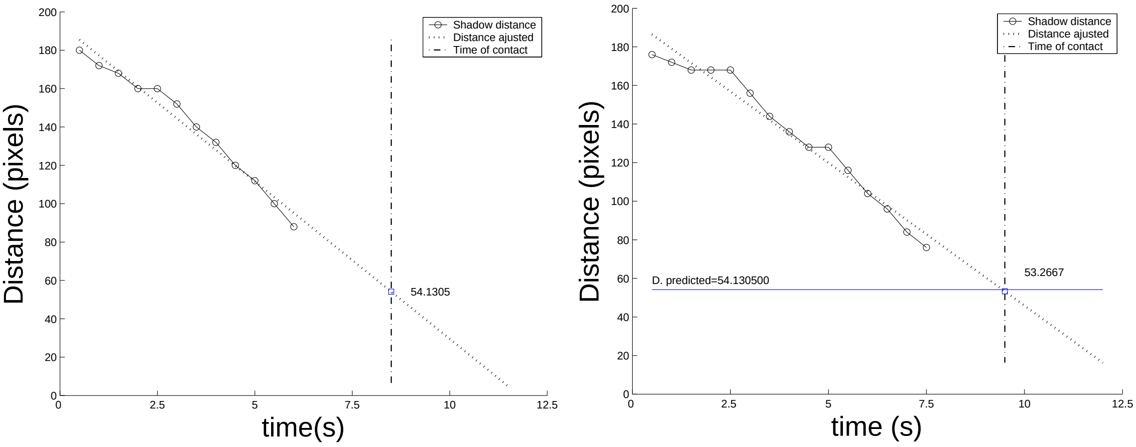

Figure 7-13: On the left is the result of driving the arm towards a surface (a brown box under ambient light). The Y axis represents the distance in pixels between the end point of the arm and the shadow. As the arm drives down, this quantity reduces linearly. Ideally, it should reach zero at the moment of contact (vertical line); in practice, there is an offset. Results for another run are shown on the right, showing that the offset is consistent.

where tc is time to contact after the shadow enters within $Y_{threshold}$ pixels of the endpoint, $slope$ is the rate of change of the shadow distance, and $Y_{contact}$ is the distance at which the arm makes contact with the surface.

This equations holds if the rate of change of the distance between the arm and shadow is roughly constant. We expect this to be true if the illumination is constant and the arm moves at constant velocity. To test these assumptions we ran the following experiment. We direct the robot to drive its arm approximately downwards

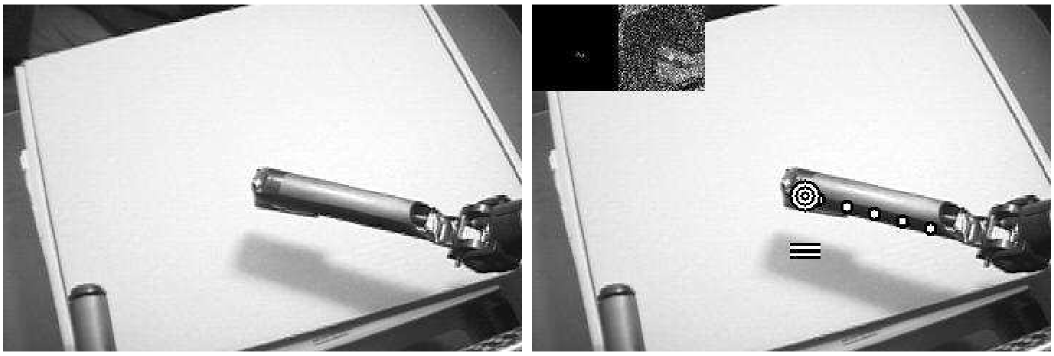

Figure 7-14: View of the flat surface from the robot surface. The illumination is ambient light. On the right we can observe the detection of the end point of the arm and its casted shadow.

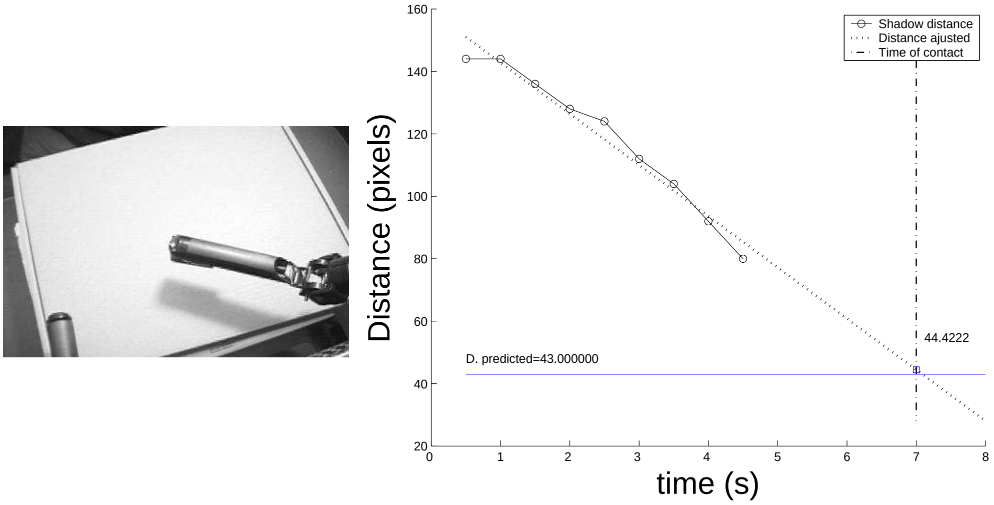

Figure 7-15: In this scenario the level of illumination has been increased and the direction of the light source changed.

using the m_2 virtual axis described in section 7.2.5. The robot detects that it touches the surface using the touch sensor in the endpoint of its arm. Figure 7-13 shows the outcome of one such trajectory. We can observe that as the robot moves its arm towards the surface the distance between the shadow and the arm is reduced. One complication shown in the figure is that when the shadow is too close to the arm (around 70 pixels), the shadow merges with the arm and cannot be reliably localized. At this point cues such as shadow darkness and crispness should be used, but are not yet available. However, time to collision still can be computed with good precision. Under the same light conditions we ran the experiment starting at different positions. The results are shown to the right in Figure 7-13. In this figure we observe that the difference between $Y_{contact}$ from the first experiment (54.1) and the current one (53.3) is very small. This supports the assumption that we can reliably and accurately compute time to contact for a given value of *slope*.

Next, we performed experiments with different surfaces and/or illumination conditions. The results are shown in Figures 7-15 and 7-16. Further experiments were done in a given scenario to address the consistency of the $Y_{contact}$ threshold and its effect in the time to contact estimation. We performed 10 runs with fixed illumination and surface while varying the initial arm position. For each run we fitted a line using

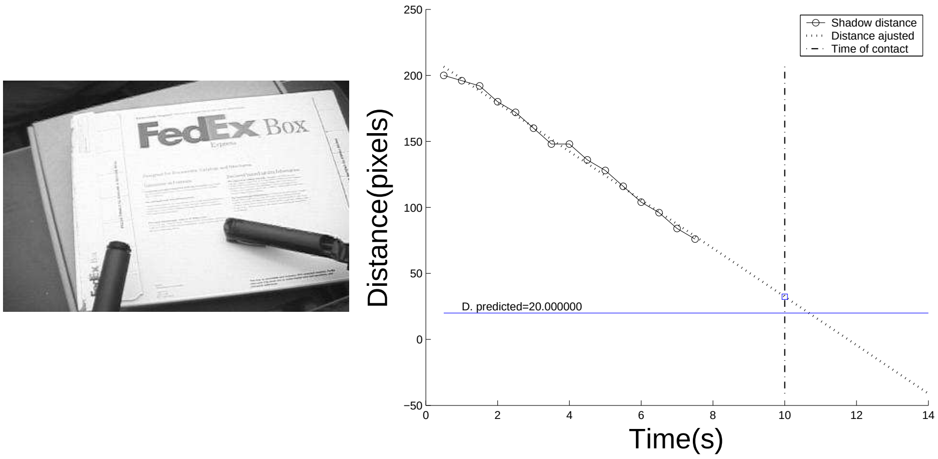

Figure 7-16: The surface used in this experiment has a different texture than in Figures 7-13 and 7-15.

the distance points whose value were above 70 pixels. This value (70 pixels) is the threshold above which the detection of the shadow is reliable. We can observe this threshold in figure 7-12 labelled as $Y_{threshold}$. From the line fitted to the data we obtained its slope. Consequently, the only remaining parameter required to determine the time to contact according to equation 7.1 is $Y_{contact}$.

$Y_{contact}$ is obtained in each one of the experiments when the touch sensor detects a contact. These results are presented in table 7.1. Due to the variability of the behavior of the touch sensor, we can observe that $Y_{contact}$ is quite noisy. Therefore, we computed its average to use in equation 7.1. The time to contact values presented in table 7.1 are calculated using the $Y_{threshold}$ value of each run and the average $Y_{contact}$.

The time to contact obtained for each of the experiments has a very small standard deviation, which shows that the method is applicable in practice, and suggest that the time estimation is more accurate than the touch sensor readings can confirm. The computation can be done very quickly since the slope obtained from a set of data can also be obtained on-line with a few points.

7.2.7 Summary and future directions

This work is part of a larger project to build a robot capable of manipulating unfamiliar objects in an unstructured environment. This is a very challenging task since

Table 7.1: Data obtained from 10 experiments ran under the same surface and illumination conditions.

Number	Slope (pixel/s)	$Y_{contact}$ (pixel)	tc (s)
1	-20.8	-5.8	2.8
2	-19.2	27.5	3.1
3	-19.0	36.8	3.1
4	-15.7	-41.4	3.8
5	-19.3	25.1	3.1
6	-20.3	6.4	2.9
7	-20.8	11.1	2.8
8	-20.2	1.5	2.9
9	-20.8	44.0	2.8
10	-20.1	3.1	2.9
mean	-19.6	10.8	3.0
stdev.	1.5	24.6	0.28

perceptual uncertainty translates directly into clumsy motion. For example, if the robot estimates the location and orientation of a surface incorrectly, then there is a limit to what can be salvaged by clever control algorithms.

Our goal is to give our robot the perceptual abilities it needs to actively resolve ambiguity in its environment, when the passive vision algorithms it uses fail. Previous work in our group showed how manipulation could be used to detect object boundaries experimentally [30]. In that work, shadows were dealt with as a nuisance. Now we want to make them a positive benefit. In this paper, we showed that shadows cast by the arm make a fine depth cue. The shadow detection mechanism could and should be extended to further evaluate the shadow, and not just report its gross position. Over important parameters include size, darkness, and the crispness of its outline. For example, as the arm approaches the surface, darkness and crispness increase dramatically. This should permit an independent measure of time to collision which can take over when gross position begins to give little information.

We are also making an effort to exploit object shadows (see Figure 7-17). Since our camera has controllable zoom and focus, we can look very closely at the boundary of objects, and confirm the existence of shadows that indicate a point at which we

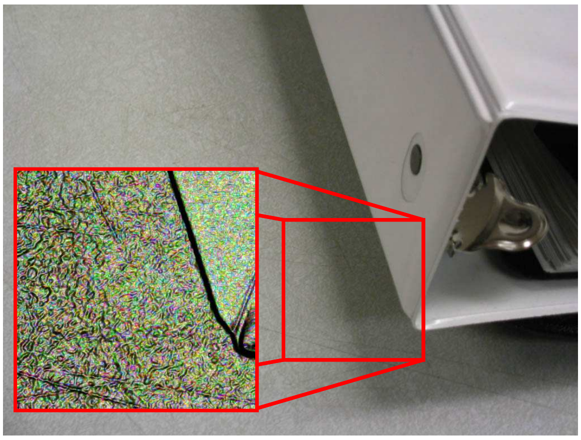

Figure 7-17: Shadows tend to be low-frequency illumination changes. With sufficient zoom and good focus, higher-frequency texture can be found and used to show the shadow is not a physical edge. Comparing cast edges with true edges can suggest the presence of physical gaps – very useful for the purposes of manipulation.

might be able to lift the object.

Chapter 8

Conclusions

The main contribution of this thesis is an alternative approach to manipulation that we call *sensitive manipulation.*

In this approach we abandoned the paradigm of stiff and highly precise arms guided by a model to manipulate objects. Instead, we provide the robots with the capability of exploring their environment. This needs hardware and software support. The hardware needs to be capable of safely coming in contact with objects and directly sensing the interaction with them. The software exploit the hardware capabilities by exploring the environment to learn from it and compensate for the lack of precision.

Sensitive manipulation has been demonstrated by implementing the first humanoid robot that integrates tactile information for guiding manipulation. This implementation has been enabled by developing important technological elements. These are: compliant tactile sensor, scalable force actuator, force actuated hand, and the integrated platform OBRERO.

8.1 Lessons learned

- Using tactile information makes manipulation conceptually easier. Ideas such as sliding a finger on an object or moving the finger until there is light contact with the object become possible to implement. However, the sensors used have to be capable of providing this information.

- Tactile sensors increase their capability of extracting information when they are mounted in a structure that moves and can safely come in contact with the world. An structure with these feature is a low mechanical impedance limb. These combination, sensors and low impedance limbs, makes exploration safe and meaningful.

- Precisely position an arm respect to an object to interact with it, it not always the best strategy. Giving up the precision in exchange of the capability of exploration, it is also a powerful strategy especially in dynamic environments.

- Tactile sensors should cover all of the hand surfaces. The current robot's hand has many places that are not covered by the skin. More often than expected, the hand comes in contact with an object in surfaces not covered by skin. Consequently, these contacts are not detected and the interaction with the object is difficult to control. This is not a problem in humans and animals that are completely covered by sensor-rich skin.

- Tactile sensors should cover the arms and other parts of the robot body. The arm, as well as the hand, come in contact with the environment. Detecting contact with the arm will help to explore the environment and to control the arm's orientation with respect to the surface where we want the robot to operate. In our case having an infrared sensor that act as tactile sensor was useful to detect contact with the table. Moreover, detecting contact with the robot's own body will make the robot safer since it will not damage itself. This might imply having tactile sensors in the robot's body. These sensors will be useful in performing whole-body grasps, for example lifting a box using its chest as support or grabbing a basketball using the forearm and the side of the body. The sensors might not need to be as sophisticated as the ones on the hand but their presence will be of great help.

- It is important to consider the behavior of a group of sensors. For example, the sensors in the palm of OBRERO have the same sensitivity as the ones on the fingers. However, because of their position in the palm, more than one sensor comes in contact with a large object's surface causing the force to be distributed. Consequently, the pressure applied to the individual sensors was small making it difficult to detect. This effect needs to be considered. Nevertheless, the sensors on the palm worked well when they came in contact with edges since the force was distributed among less sensors.

- The current fingers on OBRERO's hand have two coupled phalanges driven by only one motor which can decouple when the proximal phalange is locked. However, for tasks such as exploring or grasping, we thought that a different finger design is needed. We consider that having three phalanges (proximal, middle, distal) where the middle and the distal are coupled and can close forming a hook would be useful. The proximal phalange should be driven independently from the middle/distal coupling. From our experience, we found that it is possible to move the hand to have the proximal phalange make contact without greatly modifying an object's position. Ideally, the next step is to move the distal phalange around the object to grasp it. However, in our current design, the proximal and distal phalanges are coupled. Consequently, the proximal phalange needs to be locked to have the distal phalange move independently. In order to lock the proximal phalange, force is exerted on the object making it move. The suggested design would deal with this problem.

8.2 Future Work

We are interested in pushing the idea of dense tactile sensing to make robots more adaptive to dynamic environments. We have observed that several times (more than expected) the robot comes in contact with the environment in regions that are not covered with tactile sensors. That does not only apply to the hand but to the whole body. In these cases, it is difficult to know what action to execute to react to the environments because the contact has not been detected. In order to solve this problem, it is certainly possible to create a model. However, that solution leads very quickly to the limitations of a model-based approach. Therefore, we plan to cover the robot's body with tactile sensors. This plan involves to work in parallel to develop further the tactile sensing technology using the principles learned. The development includes: reducing the size of the sensor, increasing their mechanical durability, and making local computations for fast events as slippage.

Another important part is to tightly integrate vision with tactile feedback to improve the perception and action. For instance, in this thesis, we execute the task with no knowledge about the object and some idea of where the table is. Having a hand and arm fully covered with tactile sensors will allow the robot to explore more freely the environment and detect the table. The information about the table can be correlated with the focus information from the camera to calibrate the focus distance.

More importantly, while a robot is interacting with an object a model of the latter can be extracted. This model would combine visual and tactile sensing. Proposing to build a model might come as a surprise at this point in this thesis. This is because all along we have emphasized that we do not used one. However, in this case, the model obtained is not built a priori instead it is grounded on sensory-motor processes. This contrasts with the traditional models that used representations of objects imposed by the designer. A visual and tactile model of an object will help, for example, to improve the prediction of the position of an object when it is visually detected, to correctly preshape the hand before interacting with the object, and to preset the lifting force.

Finally, the hand was carefully designed to be used in insertion tasks. An important feature in insertion tasks is the capability of compensating for misalignments between the parts. A method to provide this capability is the remote center of compliance (RCC). In this method, the part to be inserted is mounted in a mechanism with passive degrees of freedom in the directions of the possible misalignments. In OBRERO's hand, the compliance of the fingers provides a similar mechanism to RCC. This is an area that we would like to explore further because many task involve some sort of part matching.

Bibliography

[1] Ronald C. Arkin. *Behavior-based Robotics.* MIT Press, Cambridge, MA, June 1998.

[2] Artur Arsenio and Paul Fitzpatrick. Exploiting amodal cues for robot perception. *International Journal of Humanoid Robotics*, 2(2):1–19, 2005. in press.

[3] Artur Arsenio, Paul Fitzpatrick, Charles C. Kemp, and Giorgio Metta. The whole world in your hand: Active and interactive segmentation. In *Third International Workshop on Epigenetic Robotics*, 2003.

[4] H. Asada. *Studies on Prehension and Handling by Robot Hands with Elastic Fingers.* PhD thesis, Kyoto University, 1979.

[5] D.H. Ashmead, M.E. McCarty, L.S. Lucas, and M.C. Belvedere. Visual guidance in infants' reaching toward suddenly displaced targets. *Child Dev.*, 64(4):1111–27, 1993.

[6] A. Bicchi. Hands for dexterous manipulation and robust grasping: a difficultroad toward simplicity. *Robotics and Automation, IEEE Transactions on*, 16(6), 2000.

[7] A. Bicchi and V. Kumar. Robotic grasping and contact: A review. *In Proceedings of the IEEE International Conference on Robotics and Automation*, April 2000.

[8] A. Bicchi and V. Kumar. *Ramsette: Articulated and mobile robots for services and Technology*, volume 270, chapter Robotic Grasping and Manipulation, pages 55–74. Springer-Verlag, Berlin Heidelberg, Germany, 2001.

[9] A. Bicchi, J.K. Salisbury, and D.L. Brock. Contact sensing from force measurements. *International Journal of Robotics Research*, 12(3):249–262, June 1993.

[10] J. Bouguet and P. Perona. 3D photography using shadows in dual-space geometry. *International Journal of Computer Vision*, 35(2):129–149, Nov 1999.

[11] Michael Brady, John M. Hollerbach, Timothy L. Jhonson, Tomás Lozano-Pérez, and Matthew T. Mason. *Robot Motion: Planning and Control.* MIT Press, Cambridge, MA, 1982.

[12] Rodney A. Brooks. Elephants don't play chess. *Robotics and Autonomous Systems*, 6(1&2):3–15, 1990.

[13] Rodney A Brooks. Intelligence without reason. In *Proceedings of 12th International Joint Conference on Artificial Intelligence*, pages 569–595, Sydney, Australia, August 1991. Morgan Kaufmann.

[14] A. M. Bruckstein, R. J. Holt, Y. D. Jean, and A. N. Netravali. On the use of shadows in stance recovery. *International Journal of Imaging Systems and Technology*, 11(5):315–330, 2001.

[15] E.W. Bushnell and J.P. Boudreau. Motor development and the mind: The potential role of motor abilities as a determinant of aspects of perceptual development. *Child Dev.*, 64(4):1005–1021, 1993.

[16] J. Butterfass, M. Grebenstein, H. Liu, and G. Hirzinger. Dlr-hand ii: next generation of a dextrous robot hand. *IEEE International Conference on Robotics and Automation*, 1:109–114, 2001.

[17] Chun Sheng Cai and Bernard Roth. On the planar motion of rigid bodies with point contact. *Mechanism and Machine Theory*, 21(6):453–466, 1986.

[18] M. Campos, R. Bajcsy, and V. Kumar. Exploratory procedures for material properties: the temperature perception. In *5th Int. Conf. on Advanced Robotics*, volume 1, pages 205–210, June 1991.

[19] U. Castiello, D. Lusher, C. Burton, and P. Disler. Shadows in the brain. *Journal of Cognitive Neuroscience*, 15(6):862–872, August 2003.

[20] R.K. Clifton and M.G. Clarkson D.W. Muir, D.H. Ashmead. Is visually guided reaching in early infancy a myth? *Child Dev.*, 64(4):1099–110, 1993.

[21] R.K. Clifton, P. Rochat, D.J. Robin, and N. E. Berthier. Multimodal perception in the control of infant reaching. *J Exp Psychol Hum Perform.*, 20(4):876–86, 1994.

[22] J. Coelho and R. Grupen. A control basis for learning multifingered grasps. *Journal of Robotic Systems*, 14(7):545–557, 1997.

[23] Shadow Robot Company. Design of a dextrous hand for advanced clawar applications. *Proceedings of CLAWAR 2003*, 2003.

[24] M. R. Cutkosky. *Robotc Grasping and Fine Manipulation.* Kluwer, Boston, MA, 1985.

[25] Roger W. Brockett Dimitris Hristuy, Nicola Ferrier. The performance of a deformable membrane tactile sensor basic results on geometrically defined tasks. *Int'l Conf. on Robotics and Automation*, 2000.

[26] A. Edsinger and C. Kemp. Manipulation in human environments. In *Proceedings of the IEEE/RSJ International Conference on Humanoid Robotics*, 2006.

[27] A. Edsinger and J. Weber. Domo: A force sensing humanoid robot for manipulation research. *Proceedings of the IEEE/RSJ International Conference on Humanoid Robotics*, 2004.

[28] Brooks et. al. Sensing and manipulating built-for-human environments. *International Journal of Humanoid Robots*, 1(1):1–28, 2004.

[29] S. Femmam, N. K. M'Sirdi, and A. Ouahabi. Perception and characterization of materials using signal processing techniques. *Instrumentation and Measurement*, 50(5):1203–1211, October 2001.

[30] P. Fitzpatrick and G. Metta. Grounding vision through experimental manipulation. *Philosophical Transactions of the Royal Society: Mathematical, Physical, and Engineering Sciences*, 361(1811):2165–2185, 2003.

[31] Paul Fitzpatrick. *From First Contact to Close Encounters: A Developmentally Deep Perceptual System for a Humanoid Robot.* PhD thesis, MIT, Cambridge, MA, 2003.

[32] Paul Fitzpatrick. Perception and perspective in robotics. In *Proceedings of the 25th Annual Conference of the Cognitive Science Society*, Boston, 2003.

[33] Paul Fitzpatrick, Artur Arsenio, and Eduardo R. Torres-Jara. Reinforcing robot perception of multi-modal events through repetition and redundancy and repetition and redundancy. *Interaction Studies Journal*, 2005. accepted for publication.

[34] Paul Fitzpatrick and Giorgio Metta. Yarp: Yet another robotic platform. http://yarp0.sourceforge.net/doc-yarp0/doc/manual/manual/manual.html.

[35] Paul Fitzpatrick, Giorgio Metta, Lorenzo Natale, Sajit Rao, and Giulio Sandini. Learning about objects through action - initial steps towards artificial cognition. In *Proc. of the IEEE Internat'l Conf. on Robotics and Automation*, Taipei, Taiwan, May 2003.

[36] Paul Fitzpatrick and Eduardo Torres-Jara. The power of the dark side: using cast shadows for visually-guided reaching. In *Proceedings of the IEEE International Conference on Humanoid Robotics*, Los Angeles, November 2004.

[37] E. J. Gibson. Exploratory behavior in the development of perceiving, acting, and the acquiring of knowledge. *Annual Review of Psychology*, 39:1–41, 1988.

[38] J. J. Gibson. *The Ecological Approach to Visual Perception.* Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1986.

[39] Andrew Russell Greg Hellard. A robust, sensitive and economical tactile sensor for a robotic manipulator. *Australian Conference on Robotics and Automation*, 2002.

[40] L. D. Harmon. Automated tactile sensing. *International Journal of Robotics Research*, 1(2):3–32, 1982.

[41] G. Hirzinger, A. Albu-Schaffer, M. Hahnle, I. Schaefer, and N. Sporer. On a new generation of torque controlled light-weight robots. *IEEE International Conference on Robotics and Automation*, 4:3356 – 3363, 2001.

[42] R.D. Howe and M.R. Cutkosky. Sensing skin acceleration for slip and texture perception. *Robotics and Automation. In Proceedings of IEEE International Conference on*, 1:145–150, May 1989.

[43] Robert D. Howe and Mark R. Cutkosky. *Touch Sensing for Robotic Manipulation and Recognition*, pages 55–112. The Robotics Review 2. O Khatib, J.J. Craig and T. Lozano-Perez. MIT Press, Cambridge, Massachusetts, 1992.

[44] Kaijen Hsiao and Tomás Lozano-Perez. Dynamic sensor-based control of robots with visual feedback. *IEEE /RSJ International Conference on Intelligent Robots and Systems*, October 2006.

[45] Helen H. Hu, Amy A. Gooch, Sarah H. Creem-Regehr, and William B. Thompson. Visual cues for perceiving distances from objects to surfaces. *Presence: Teleoperators and Virtual Environments*, 11(6):652–664, 2002.

[46] Helen H. Hu, Amy A. Gooch, William B. Thompson, and Brian E. Smits. Visual cues for imminent object contact in realistic virtual environments. In

Proceedings of the 11th IEEE Visualization Conference, Salt Lake City, Utah, October 2000.

[47] G. S. Hubona, G. Shirah, and D. Jennings. The effects of cast shadows and stereopsis on performing computer-generated spatial tasks. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 34(4):forthcoming, July 2004.

[48] Geoffrey S. Hubona, Philip N. Wheeler, Gregory W. Shirah, and Matthew Brandt. The relative contributions of stereo, lighting, and background scenes in promoting 3D depth visualization. *ACM Transactions on Computer-Human Interaction*, 6(3):214–242, 1999.

[49] Interlink Electronics, Inc, 546 Flynn Road, Camarillo, CA 93012. *Integration Guide*, 1998. P/N 90-65801 Rev. C.

[50] S.C. Jacobsen, J.E. Wood, D. F. Knutti, and K. B. Biggers. *The UTAH/MIT dextrous hand: Work in progress*, pages 341–389. Robot Grippers. Springer-Verlag, Berlin, 1986.

[51] Helge Ritter Jan Jockusch, Jorg Walter. A tactile sensor system for a three-fingered robot manipulator. *Int. Conf. on Robotics and Automation*, 1997.

[52] R. S. Johansson. How is grasping modified by somatosensory input? In D. R. Humphrey and H J Freund, editors, *Motor Control: Concepts and Issues*, pages 331–355. John Wiley and Sons Ltd, Chichester, 1991.

[53] R. S. Johansson and G. Westling. *Attention and performance XIII*, chapter Tactile afferent signals in control of precision grip. Lawrence Erlbaum Assoc., Hillsdale, N.J., 1990.

[54] R. Platt Jr., A. H. Fagg, and R. A. Grupen. Extending fingertip grasping to whole body grasping. *Proceedings of International Conference on Robotics and Automation*, pages 2677–2682, 2003.

[55] Richard M. Voyles Jr., Gary Fedder, and Pradeelp K. Wlosla. Design of a modular tactile sensor and actuator based on an electrorheological gel. *Int'l Conf. on Robotics and Automation*, 1996.

[56] Robert Platt Jr., Andrew H. Fagg, and Roderic A. Grupen. Nullspace composition of control laws for grasping. *In Proceedings of IROS*, 2002.

[57] Eric R. Kandell, James H. Schwartz, and Thomas M. Jessell. *Principles of Neural Science*. McGraw-Hill, fourth edition, 2000.

[58] Charles Kemp. *A Wearable System that Learns a Kinematic Model and Finds Structure in Everyday Manipulation by using Absolute Orientation Sensors and a Camera*. PhD thesis, MIT-CSAIL, 2005.

[59] D. Kersten, D. C. Knill, P. Mamassian, and I. Bulthoff. Illusory motion from shadows. *Nature*, 379(6560):31, 1996.

[60] D. Kersten, P. Mamassian, and D. C. Knill. Moving cast shadows induce apparent motion in depth. *Perception*, 26(2):171–192, 1997.

[61] R. L. Klatzky, S. J. Lederman, and V. A. Metzger. Identifying objects by touch: An expert system. *Perception and Psychophysics*, 4(37):299–302, 1985.

[62] A. Konno, K. Nagashima, R. Furukawa, K. Nishiwaki, T. Noda, M. Inaba, and H. Inoue. Development of a humanoid robot saika. *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 805–810, 1997.

[63] A. Konno, K. Nishiwaki, R. Furukawa, M. Tada, K. Nagashima, M. Inaba, and H. Inoue. Dexterous manipulations of humanoid robot saika. *Preprints of Fifth Int. Symp. on Experimental Robots*, pages 46–57, 1997.

[64] P. R. Kraus, A Fredriksson, and V. Kumar. Modeling of frictional contacts for dynamics simulation. *Proceedings of Workshops on Dynamic Simulation: Methods and Applications*, pages 33–38, 1997.

[65] Pencilla Lang. Design and prototyping of a fibre optic tactile array. *The Journal of High School Science*, November 2002.

[66] Pencilla Lang. Optical tactile sensors for medical palpation. *Canada-Wide Science Fair*, May 2004.

[67] S. J. Lederman and R. L. Klatzky. Hand movements: A window into haptic object recognition. *Cognitive Psychology*, 19(3):342–368, 1987.

[68] S. J. Lederman and R. L. Klatzky. Haptic identification of common objects: Effects of constraining the manual exploration process. *Perception and Psychophysics*, 66:618–628, 2004.

[69] D. N. Lee. The optic flow field: the foundation of vision. *Philosophical Transactions of the Royal Society of London B*, 290(1038):169–179, 1980.

[70] C.S. Lovchik and M.A. Diftler. The robonaut hand: a dexterous robot hand for space. *IEEE International Conference on Robotics and Automation*, 2:907 – 912, May 1999.

[71] Vivek Maheshwari and Ravi F. Saraf. High-resolution thin-film device to sense texture by touch. *Science*, 312(5779):1501 – 1504, June 9 2006.

[72] P. Mamassian, D. C. Knill, and D. Kersten. The perception of cast shadows. *Trends in Cognitive Sciences*, 2(8):288–295, 1998.

[73] Giorgio Metta. *Babybot: a study into sensorimotor development.* PhD thesis, LIRA-Lab, DIST, September 2000.

[74] Giorgio Metta and Paul Fitzpatrick. Early integration of vision and manipulation. *Adaptive Behavior*, 11(2):109–128, June 2003.

[75] A.T. Miller and P.K. Allen. Graspit!: A versatile simulator for robotic grasping. *Robotics and Automation Magazine, IEEE*, 11(4):110–122, Dec. 2004.

[76] David J. Montana. The kinematics of contact and grasp. *International Journal of Robotics Research*, 7(3):17–32, June 1988.

[77] C. J. Morse. Design of a quadruped walking robot for social interaction. Master's thesis, MIT Department of Electrical Engineering and Computer Science, Feb 2001.

[78] L. Natale, G. Metta, and G. Sandini. Learning haptic representation of objects. In *International Conference on Intelligent Manipulation and Grasping*, Genoa, Italy, July 2004.

[79] L. Natale, F. Orabona, F. Berton, G. Metta, and G. Sandini. From sensorimotor development to object perception. In *IEEE-RAS International Conference on Humanoid Robots*, Tsukuba, Japan, December 5-7 2005.

[80] L. Natale, F. Orabona, G. Metta, and G. Sandini. Exploring the world through grasping: a developmental approach. In *Proceedings of the 6th CIRA Symposium*, Espoo, Finland, June 2005.

[81] Lorenzo Natale. *Linking Action to Perception in a Humanoid Robot.* PhD thesis, LIRA-Lab, University of Genoa, 2004.

[82] F. Pavani and U. Castiello. Binding personal and extrapersonal space through body shadows. *Nature Neuroscience*, 7(1):14–15, Jan 2004.

[83] R. Pelossof, A. Miller, P. Allen, and T. Jebara. An svm learning approach to robotic grasping. In *Proceedings of IEEE International Conference on Robotics and Automation. ICRA*, volume 4, pages 3512–3518, April 2004.

[84] R. Platt, R.A. Grupen, and A.H. Fagg. A control basis for learning multifingered grasps. *Humanoid Robots, 5th IEEE-RAS International Conference on*, pages 141–147, 2005.

[85] Robert Platt. *Learning and generalizing control-based grasping and manipulation skills.* PhD thesis, University of Massachusetts Amherst, September 2006.

[86] A. Prati, I. Mikic, M. M. Trivedi, and R. Cucchiara. Detecting moving shadows: algorithms and evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):918–923, July 2003.

[87] J. L. Richmond and D. K. Pai. Active measurement and modeling of contact sounds. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 2146–2152, San Francisco, April 2000.

[88] D. Robinson, J. Pratt, D. Paluska, and G. Pratt. Series elastic actuator development for a biomimetic robot. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 1999.

[89] J.K. Salisbury. *Kinematic and Force Analysis of Articulated Hands*. PhD thesis, Stanford University, May 1982.

[90] N. Sarkar, V. Kurmar, and X. Yun. Velocity and acceleration equations for three-dimensional contact. *Journal of Applied Mechanics*, 1996.

[91] S. Schaal and C.G. Atkenson. Constructive incremental learning from only local information. *Neural Computation*, (10):2047–2084, 1998.

[92] P. R. Sinha and J. M. Abel. A contact stress model for multifingered grasps of rough objects. *IEEE Transactions in Robotics and Automation*, 8:7–22, 1992.

[93] Takao Someya, Yusaku Kato, Tsuyoshi Sekitani, Shingo Iba, Yoshiaki Noguchi, Yousuke Murase, Hiroshi Kawaguchi, and Takayasu Sakurai. Conformable, flexible, large-area networks of pressure and thermal sensors with organic transistor active matrixes. *Proc. of National Academy of Sciences*, August 2005.

[94] J. Stauder, R. Mech, and J. Ostermann. Detection of moving cast shadows for object segmentation. *IEEE Transactions on Multimedia*, 1(1):65–76, March 1999.

[95] A. Streri. *Seeing, reaching, touching: the relations between vision and touching in infancy.* MIT Press, 1993.

[96] A. Streri and M-G. Pêcheux. Tactual habituation and discrimination of form in infancy: a comparison with vision. *Child Development*, (57):100–104, 1986.

[97] W. B. Thompson, P. Shirley, B. Smits, D. J. Kersten, and C. Madison. Visual glue. Technical Report UUCS-98-007, University of Utah, March 1998.

[98] E. Torres-Jara and J. Banks. A simple and scalable force actuator. *35th International Symposioum on Robotics*, March 2004.

[99] E. Torres-Jara, L. Natale, and P. Fitzpatrick. Tapping into touch. In *Fith International Workshop on Epigenetic Robotics (forthcoming)*, Nara, Japan, July 22-24 2005. Lund University Cognitive Studies.

[100] Eduardo Torres-Jara. Obrero: A platform for sensitive manipulation. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2005.

[101] William T. Townsend. *The Effect of Transmission Design on Force-Controlled Manipulator Performance.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1988.

[102] William T. Townsend. The barretthand grasper - programmably flexible part handling and assembly. *Industrial Robot: An International Journal*, 27(3):181–188, 2000.

[103] A.L. van der Meer, F.R. van der Weel, and D.N. Weel. The functional significance of arm movements in neonates. *Science*, 267:693–5, 1995.

[104] R. Volpe. *Real and Artificial Forces in the Control of Manipulators: Theory and Experiments.* PhD thesis, Carnegie Mellon University, Department of Physics, Sept. 1990.

[105] C. von Hofsten. Eye-hand coordination in the newborn. *Dev. Psychology*, 18(3):450–61, 1982.

[106] C. von Hofsten. An action perspective on motor development. *Trends in cognitive sciences*, 8(6):266–272, 2004.

[107] Y. Wang and V. Kumar. Simulation of mechanical systems with unilateral constraints. *Journal of Mechanical Design*, 116:571–580, 1994.

[108] Y.-T. Wanga, V. Kumar, and J. Abel. Dynamics of rigid bodies undergoing multiple frictional contacts. *IEEE International Conference on Robotics and Automation*, 3:2764–2769, May 1992.

[109] L. R. Wanger, J. A. Ferwerda, and D. P. Greenberg. Perceiving spatial relationships in computer-generated images. *IEEE Computer Graphics and Applications*, 12(3):44–58, May 1992.

[110] L. Weiss, A. Sanderson, and C. Neuman. Dynamic sensor-based control of robots with visual feedback. *IEEE Journal of Robotics and Automation*, RA-3(5):404–417, October 1987.

[111] M. Williamson. Series elastic actuators. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 1995.

[112] M. Williamson. Neural control of rhythmic arm movements. *Neural Networks*, 11(7-8):1379–1394, 1998.

[113] Matthew M. Williamson. Exploiting natural dynamics in robot control. In *Fourteenth European Meeting on Cybernetics and Systems Research (EMCSR '98)*, Vienna, Austria, 1998.

[114] S. Middelhoek Z. Chu, P.M. Sarro. Silicon three-axial tactile sensor. *International Conference on Solid-State Sensors and Actuators*, 1995.