Touch driven dexterous robot arm control

Zhanat Kappassov

A dissertation submitted to

Pierre & Marie Curie University

Sorbonne Universités, Université Pierre & Marie Curie Paris VI

Doctoral School of Mechanics, Acoustic, Electronic and Robotic Science of Paris

École Doctorale de Sciences Mécanique, Acoustique, Électronique et Robotique de Paris

in partial fulfillment of the requirements for the Ph.D. degree

in

Robotics

by

Zhanat KAPPASSOV

Touch Driven Dexterous Robot Arm Control

To be defended in February 2017

Committee

Youcef MEZOUAR	Professor at Sigma-Clermont	Referee
Jean-Pierre GAZEAU	Professor at the University of Poitiers	Referee
Agnès ROBY-BRAMI	Directeur de recherche INSERM	Examiner
Guillaume WALCK	Research engineer at Bielefeld University	Examiner
Kaspar ALTHOEFER	Professor at Queen Mary University of London	Examiner
Véronique PERDEREAU	Professor at the Pierre and Marie Curie University	Adviser
Juan Antonio CORRALES RAMÓN	Associate Professor at Sigma-Clermont	co-Adviser

Abstract

Robots have improved industry processes, most recognizably in conveyor-belt assembly systems, and have the potential to bring even more benefits to our society in transportation, exploration of dangerous zones, deep sea or even other planets, health care and in our everyday life. A major barrier to their escape from fenced industrial areas to environments co-shared with humans is their poor skills in physical interaction tasks, including manipulation of objects. While the dexterity in manipulation is not affected by the blindness in humans, it dramatically decreases in robots. With no visual perception, robot operations are limited to static environments, whereas the real world is a highly variant environment.

In this thesis, we propose a different approach that considers controlling contact between a robot and the environment during physical interactions. However, current physical interaction control approaches are poor in terms of the range of tasks that can be performed. To allow robots to perform more tasks, we derive tactile features representing deformations of the mechanically compliant sensing surface of a tactile sensor and incorporate these features to a robot controller via touch-dependent and task-dependent tactile feature mapping matrices.

As a first contribution, we show how image processing algorithms can be used to discover the underlying three dimensional structure of a contact frame between an object and an array of pressure sensing elements with a mechanically compliant surface attached onto a robot arm's end-effector interacting with this object. These algorithms obtain as outputs the so-called *tactile features*. As a second contribution, we design a tactile servoing controller that combines these tactile features with a position/torque controller of the robot arm. It allows the end-effector of the arm to steer the contact frame in

a desired manner by regulating errors in these features. Finally, as a last contribution, we extend this controller by adding a task description layer to address four common issues in robotics: exploration, manipulation, recognition, and co-manipulation of objects.

Throughout this thesis, we make emphasis on developing algorithms that work not only with simulated robots but also with real ones. Thus, all these contributions have been evaluated in experiments conducted with at least one real robot. In general, this work aims to provide the robotics community with a unified framework to that will allow robot arms to be more dexterous and autonomous. Preliminary works are proposed for extending this framework to perform tasks that involve multicontact control with multifingered robot hands.

Keywords: Robot Arms/Hands Control, Physical Interaction, Tactile Sensing Arrays, Tactile Servoing, Manipulation, Exploration, Co-manipulation

RÉSUMÉ

Les robots ont amélioré les projet industriels, notamment les systèmes d'assemblage basé sur des tapis roulants et ont le potentiel l'apporter plus de bénéfices: les transports, l'exploration de zones dangereuses, de fonds sous - marins et même d'autres planètes, la santé et dans la vie courante. Une barrière majeure pour leur sortie des environnements industriels en enceintes protegée vers des environnements partagés avec les humains, c'est leur capacité réduite dans les tâches d'interaction physique, inclure la manipulation d'objets. Tandis que la dextérité de la manipulation n'est pas affectée par la cécité chez les humains, elle décroit énormément chez les robots: ils sont limités à des environnements statiques, alors que le monde réel est très changeant.

Dans cette thèse, nous proposons une approche différente qui considère le contrôle du contact pendant les interactions physiques entre un robot et l'environnement. Néanmoins, les approches actuelles pour l'interaction physique sont pauvres par rapport au nombre de tâches qu'elles peuvent exécuter. Pour permettre aux robots d'exécuter plus de tâches, nous obtanions des caractéristiques tactiles représentant les déformations de la surface souple d'un capteur tactile et nous incorporons ces caractéristiques dans le contrôleur d'un robot à travers des matrices de mapping tactile basées sur les informations tactiles et sur les tâches à développer.

Dans une première contribution, nous montrons comment les algorithmes de traitement d'images peuvent être utilisés pour découvrir la structure tridimensionnelle subjacente du repère de contact entre un objet et une matrice de capteurs de pression avec une surface souple attachée à l'effecteur d'un bras robotique qui interagit avec cet objet. Ces algorithmes obtiennent comme sorties les dites caractéristiques tactiles. Dans une deuxième contribution, nous avons conçu un contrôleur qui combine ces caractéristiques tactiles avec un contrôleur position-couple du bras robotique. Il permet à l'effecteur du bras de déplacer le repère du contact d'une manière désirée à travers la régulation d'erreurs de ces caractéristiques. Finalement, dans une dernière contribution, nous avons étendu ce contrôleur avec l'addition d'une couche de description de tâches pour adresser quatre problèmes communs de la robotique: l'exploration, la manipulation, la reconnaissance et la co-manipulation d'objets.

Tout au long de cette thèse, nous avons mis l'accent sur le développement d'algorithmes qui ne marchent pas simplement avec des robots simulés mais aussi avec des robots réels. De cette manière, toutes les contributions ont été évaluées avec des expériences faites avec au moins un robot réel.

En général, ce travail a comme objectif de fournir à la communauté robotique un cadre unifié qui permette aux bras robotiques d'être plus dextres et autonomes. Des travaux préliminaires ont été proposés pour étendre ce cadre au développement de tâches qui impliquent un contrôle multi-contact avec des mains robotiques multi-doigts.

Mots clés: contrôle de bras/mains robotiques, interaction physique, matrices de capteurs tactiles, asservissement tactile, manipulation, exploration, co-manipulation.

Contents

List of Figures

Chapter 1
Introduction

THE AUTOMATIZATION of manufacturing processes using machines and robots has been a rapidly evolving field within the last century. Engineers' inspiration came from the intention to pursue an increase in industrial productivity, to investigate explosive and toxic spaces, as well as to decrease human labor in household chores, thus increasing overall life quality.

Rapid industrialization initiated the development of more and more sophisticated conveyor systems in early XX^{th}. Later, the advent of electronic computers allowed programming complex systems' behaviors. It led to the first demonstration of a robot integrated into a production line in 1961, which is credited to General Motors.

In order to use robots in specific tasks we need to program efficient and robust algorithms managing their interactions with the environment. The robots obtain the information on surrounding objects by means of various types of sensing modalities. In partial analogy to human perception receptors, the robots receive signals from visual, audio, and tactile sensors. In this work, we focus on the latter. Unlike the other ones, the tactile modality integrated with dexterous manipulators opens the way to autonomous operation with complete exclusion of human intervention. At this stage, a lot of work is still to be done in order to achieve this ambitious goal.

With this goal in hand, this dissertation is an attempt to improve robotic arms/hands performance by introducing a control framework for touch driven robots. The efficacy of the approach is demonstrated by a number of experiments conducted with two robot platforms: a motorized light-weight arm and a mechanically compliant pneumatic arm with a robot hand attached onto its end-effector. The framework can be implemented

with any industrial robotic manipulators and service robots.

1.1 Motivation

The robot performance fells below human performance in regards to physical interaction with the environment. One of the reasons for this is that artificial sensors are not as capable as human sensors. In most cases related to scene recognition and robot control in variant environments, problems are ill-conditioned: a small error in images or interaction forces can have fatal consequences. In facial emotion recognition, as an example, an error in a digital image would lead to a wrong guess about the emotion imprinted on the image. Human perception, in other hand, is very robust, able to deal with noise, distortions, and changes in the environment. Another reason is that robots are controlled by preprogrammed algorithms, and thus, cannot anticipate and deal with such unforeseen situations that may be encountered by humans.

In studies of human perception, it has been observed that the visual system better recognizes the front view of objects, whereas the hand better recognizes objects from the back [Newell 01]. This complementary visual and tactile knowledge enhances the perception of three-dimensional objects. Humans learn the shapes of objects since their childhood by grasping, manipulating, and exploring surfaces of the objects. Drawing on the idea of visio-haptic perception in humans, robotic systems can also benefit from tactile perception.

In industrial applications, the concept of co-manipulation has emerged in recent years and attracted a growing interest. A number of manufacturing processes involve a combination of automated assembly tasks with manual tasks performed by human operators, since there are still rather complex manipulation tasks that cannot be autonomously performed by a robotic system. Therefore, there is a need for robot hand/arm assistants that perform complex manufacturing tasks in cooperation with a human operator. As it was mentioned above, tactile perception plays an essential role in the tasks that involve interactions with the environment; co-manipulation is not an exception.

In service robotics, the ability to manipulate objects is of utmost importance. The

number of contact points between the robot and the objects during manipulation tasks may vary a lot and be more than one. Hence, tactile sensors based on conventional force and torque sensors may result in task execution failures as they cannot detect multiple contacts. Arrays of pressure sensors can be used to cope with this problem; this fact leads to a need for more advanced control algorithms using the tactile data stream of the pressure sensing arrays.

In summary, many robot arm/hand applications (e.g. car assembly, food sorting, human-robot co-manipulation to name a few) require tactile perception to control the robot arm/hand. Therefore, the ability to control physical interactions is necessary for robots to operate in a real environment.

1.2 Goals and questions

Similar to the human skin, nowadays tactile sensors are compliant: the sensing surface deforms under external forces applied by the environment. The flexibility of the surface enhances the grasping forces and compensates for imprecisions in robot positioning caused by non-ideal actuators and the inaccuracy of their encoders.

In general, the goal of the research conducted within this thesis is to make robots more versatile and adaptable, able to interact with highly variant environments through their artificial tactile skin. In order to achieve it, this thesis develops a general control framework for robots to handle the contact frames that occur during interactions between the environment and a pressure sensing array with a compliant surface.
The following questions are considered in this work to accomplish this primary research objective:

- What are the measures that best represent the contact between the environment and a pressure sensing array?
- How can the contact measures be applied within a robot's control loop to steer the contact in a desired manner?
- How can tactile data be used in autonomous robot hands/arms to accomplish tasks

like exploration, manipulation, recognition, and co-manipulation?

- How can tactile data from multiple sensors installed on an autonomous multifingered robot hand can be used to reconstruct and localize objects within the hand while manipulating them?

1.3 Contributions of the thesis

The research field of robotics is becoming more and more interdisciplinary in nature. Being previously comprised of computer science and, mechanical and electrical engineering, nowadays robotics includes neuroscience, biology, and physiology studies on sensory-motor learning and coordination. The research reported in this work follows a similar direction. It represents the integration in a novel way of a number of ideas with real robot platforms for autonomous interaction with the environment. The progress that is reported in the thesis lies in the exploitation of rapidly growing tactile sensing technologies into a reconfigurable system that can manipulate, explore, recognize objects and perform co-manipulation tasks. The main contributions of this research work are: the identification of a key problem to be solved for making robots more dexterous and reactive, and the solution to this problem in the particular task of the interaction with the environment.

Although there are ongoing theoretical investigations of problems in robotic sensory-motor control, development and experimental validation of techniques in real, unconstrained, and variant environments is also needed. There is at present a large imbalance between robots' functionalities in simulated environments and those in the actual world. A rather big part of this work was therefore dedicated to bridging the gap between theoretical robotics and working systems that really interact with the environment.

In general, this research extends the present capabilities of robotic systems in manipulation, exploration, and co-manipulation tasks. More precisely, the tactile based control of a robotic system is studied and a novel **tactile servoing framework** (Fig. 1.1) is proposed.

The framework consists of seven major parts: a robot arm with a tactile sensor, a kinematic model of the robot, a position controller of the robot's end-effector, a tactile

data processing part, a tactile controller, a task descriptor, and an application of the touch-driven robot. The tactile data processing part extracts the information from points of contact, and sends it to the tactile controller and to the task descriptor. Depending on the application, the task descriptor defines the way how the tactile controller drives the end-effector that is physically interacting with the environment.

The contributions of this thesis are as follows:

New tactile features. When a robot with a compliant sensing array interacts with the edge of an object, the contact forces cause deformations of the compliant surface of the tactile sensing array. These deformations are estimated by newly proposed tactile features that represent moments of force around two axes lying in the sensing plane. This in turn allows to control the robot in the cases when the edge is not fully covering the sensing surface or when the force applied by the edge is not distributed uniformly.

Touch-dependent inverse tactile Jacobian/task-dependent projection matrices. In order to drive the robot using the set of the proposed tactile features, two tactile matrices are proposed that relate the feature errors to the Cartesian pose error of an end-effector. Using these matrices the motion twist of the robot's end-effector can be related to any of the features' errors, which allows the robot to manipulate, explore, align with objects, and perform co-manipulation based on the tactile errors.

Unifying robotic tasks. Different robotic tasks are unified under a single tactile servoing framework. This opens a door for extending the proposed tactile servoing framework to more advanced tasks that involve touch-driven motions.

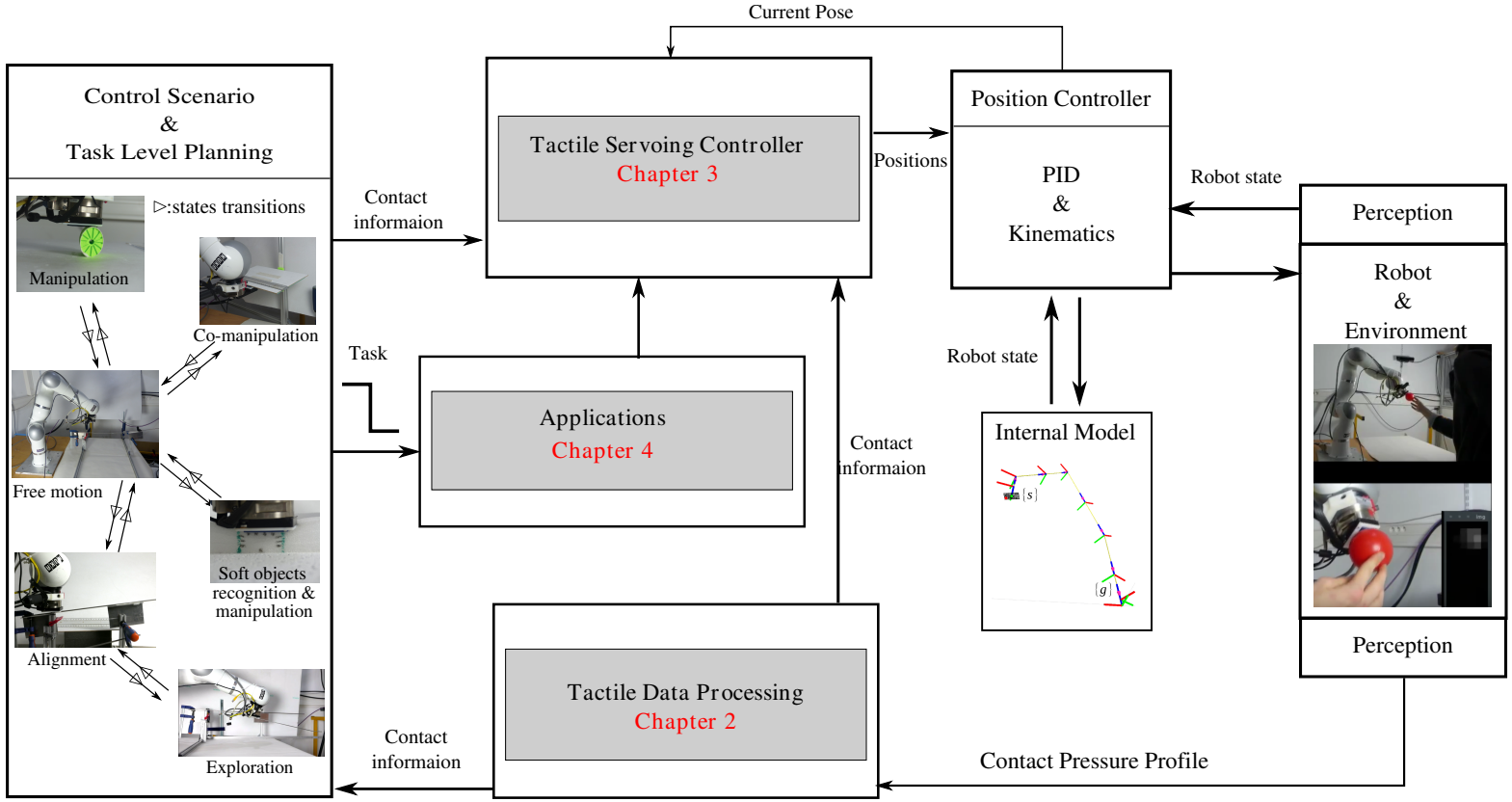


Figure 1.1: Overview of the tactile servoing framework developed for controlling interactions with the environment. Chapter 2 presents the contribution to tactile feature extraction and describes how a contact frame is defined by tactile features. Chapter 3 describes the developed tactile servoing controller that handles tracking and control of the contact frame. Chapter 4 shows how the tactile features and controller are used to perform manipulation, exploration and co-manipulation tasks.

1.4 Structure of the thesis

This thesis contains six chapters to present the contributions towards the identified research questions. The remainder of the thesis is organized as follows:

- Chapter 2 describes the signal processing techniques applied to process signals of planar sensing arrays in order to estimate contact frames during interactions with the environment. The chapter overviews the state-of-the-art, which is followed by explanations on the feature extraction algorithms and some results in simulation and reality.

- Chapter 3 proposes the Tactile Servoing Controller – the core of the developed framework. The controller is dedicated to control interactions with the environment. Tactile-based matrices are introduced and explained after reviewing related works in tactile servoing. Benchmarking the tactile controller is performed with controller responses on real manipulation platforms.

- Chapter 4 describes the task description layer of the proposed tactile servoing framework and illustrates its applications. These applications include exploration of an unknown environment, object manipulation by rolling, and human-robot co-manipulation. The chapter also shows results of manipulation of deformable and rigid objects, rolling objects with cylindrical and elliptical shapes, reconstruction of rigid objects, and estimation of stiffness of a non-static object.

- Chapter 5 presents the future direction – Tactile Localization and Mapping (TaLaM) – and describes the preliminary works on the integration of custom-made capacitive tactile sensors developed by the CEA (Center for Atomic Energy in France) onto a multifingered hand. The state-of-the-art of tactile sensing in dexterous robot hands is reviewed. The extension of the tactile servoing control framework by TaLaM is introduced in order to control multifingered robot hands. A dexterous hand makes contact with the environment using its finger surfaces and thus, contacts occur at multiple locations. Therefore, tactile feature errors from multiple contacts are to be considered as the input parameters for the TaLaM.

- Chapter 6 concludes the thesis.

1.5 Related publications

The review and novel developments introduced in this work have been peer-reviewed and validated with the publications listed hereafter:

Journal articles

- Z. Kappassov, J. A. Corrales, V. Perdereau, "Tactile sensing in Dexterous Robot Hands – Review", *Robotics and Autonomous Systems* , vol. 74, issue 1, pp. 195-220, 2015 (referenced by Journal Citation Reports).

Conference publications

- Z. Kappassov, J. A. Corrales, V. Perdereau, "ZMP features for Touch Driven Robot Control via Tactile Servo", *International Conference on Experimental Robotics (ISER)*, presented, Tokyo, Japan, October 3 - 6 , 2016. Recipient of the TOYOTA Student Participation Award.

- A. Vasquez, Z. Kappassov, V. Perdereau, "In-hand Object Shape Identification Using Invariant Proprioceptive Signatures", *International Conference on Intelligent Robots and Systems (IROS)* , Daejeon, Korea, October 9 - 14, 2016.

- Z Kappassov, Y Khassanov, A Saudabayev, A Shintemirov, HA Varol, "Semi - anthropomorphic 3D printed multigrasp hand for industrial and service robots", *International Conference on Mechatronics and Automation (ICMA)*, Takamatsu, Japan, August 4 - 7, 1697-1702, 2013.

Workshop publications

- Z. Kappassov, G. Maldonado, J. A. Corrales, V. Perdereau, "Tactile features for Touch Driven Robot Control via Tactile Servo" *IROS*, Hamburg, Germany September 28, 2015.

Chapter 2
Contact frame and tactile features

This chapter presents the tactile information that can be derived from signals of pressure sensing arrays during physical interactions. In many robotic applications, these sensors are modeled as rigid arrays of adjacent pressure sensing elements so that the geometric deformations of their surfaces are neglected. This thesis proposes to define these deformations with rotational and translational compliance variables that are later used as new tactile features for tactile servoing tasks. Section 2.2 reviews related works and recent techniques to solve the problem of tactile data processing. As a result of these tactile data processing techniques, tactile features are obtained and they define a contact frame. Section 2.3 describes the feature extraction algorithms. It also introduces definitions used in the following work. Two contact types are considered: point- and edge- contacts that are described in Section 2.3.1 and 2.3.2, respectively. Section 2.3.2 introduces the Zero Moment Point features. In order to evaluate the tactile feature extraction algorithms, a simulated model of a pressure sensing array is created in Section 2.4.1. Finally, the tactile features are evaluated using the simulation model and the real sensor in 2.4.2.

2.1 Introduction

A lot of everyday and manufacturing tasks, which include object manipulation, grasping, and surface exploration tasks, always involve physical interactions. They therefore ask nowadays autonomous robots and manipulators for more dexterous physical interactions with the environment and the humans [Gates 07]. These physical interactions need tactile feedback (referred to as "the artificial sense of touch") in order to control the properties of the contact between robots and objects [Lee 99].

The sense of touch can be given to robots by artificial tactile sensors [Cutkosky 14]. They can be proprioceptive (intrinsic), such as actuator torque sensors, or exteroceptive, such as force/pressure sensors [Dahiya 10]. Even though intrinsic sensors can give

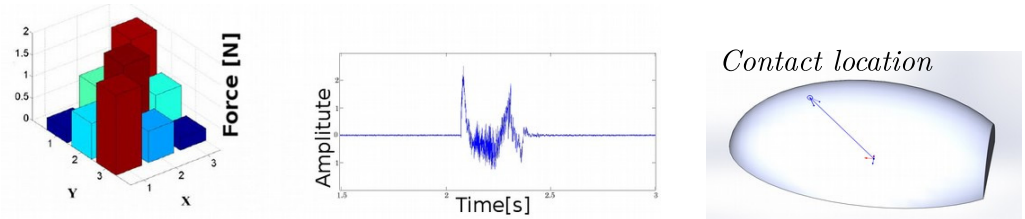
approximate information about interaction forces as shown in [Righetti 14], the exteroceptive sensors give much more precise information about interaction properties [Wettels 14]. Thus, exteroceptive sensors (tactile sensors) are considered in the following.

There are different types of artificial tactile sensors (see appendix B for a detailed review of tactile sensing technologies): force-torque sensors, dynamic tactile sensors, and pressure profile sensing arrays. They provide different types of data stream as illustrated in Fig. 2.1. In general the data stream can be dynamic or static, and single-point or multi-point contact according to the time response and spatial resolution of the tactile sensor type. Different tactile sensor types measure different properties, including force vectors, vibrations, and contact patterns:

- Pressure profile sensing arrays can provide the information about the interaction forces and the locations of the multiple contact points as shown in Fig. 2.1a.
- Tactile sensors with fast response (vibro-tactile sensors), such as accelerometers (microphones) [Kyberd 98] and barometers with fluid media [Fishel 08] can provide the information about vibrations at the contact point (Fig. 2.1b).
- Force-torque sensors located inside a solid body with known shape can provide the information about interaction forces, as well as the location of the contact point [Liu 12b] as illustrated in Fig. 2.1c. However, they can provide information about one-point contact only.

The force-torque and vibro-tactile sensors can detect contact with an object, but they do not provide any information regarding the locations of several points of contact. Therefore, the pressure profile sensing arrays are superior in terms of providing the spatial information among these types of sensors.

The tactile sensing array is constructed as an array of pressure sensing elements, referred to as *tactel* or *sensing cell*. Each tactel measures the pressure caused by an object pressed against the tactel. And the whole array of tactels provides a spatial distribution of the pressures (herein referred to as contact pattern for the sake of readability). During physical interactions, the contact pattern serves as a source to estimate a *contact frame*. The contact frame lies at the contact point in the three-dimensional (3D) Cartesian space



(a) (b) (c)

Figure 2.1: Tactile sensing signal types: (a) a two-dimensional pressure distribution of a tactile sensing array, where the sensing tactels are located on xy plane and force/pressure is measured along z-axis [Xie 14]; (b) dynamic tactile signal from a single tactel or from an ensemble of tactels, which can be acquired during a slippage [Göger 09]; (c) 6 DoF force/torque sensor measurements in an ellipsoid-shape fingertip [Liu 12b].

(the axes and the origin of the frame are shown in Fig. 2.2). The contact frame is of utmost importance during physical interactions, because most of the tasks, involving the interactions between robots and environment, are naturally defined in the 3D Cartesian space, where objects lie.

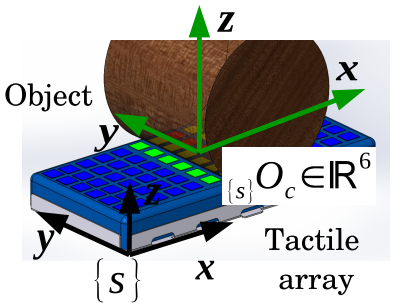


Figure 2.2: A tactile sensing array in contact with an object and the corresponding contact frame $_{\{s\}}O_c$. Green system of coordinates illustrates the contact frame with respect to the sensing frame $\{s\}$.

The structure of a tactile sensing array can be rigid or flexible. Rigid sensors cannot cover an arbitrary shape unless they were specifically manufactured for a given part of a robot, as for example the fingertip sensors developed at the University of Bielefeld [Koiva 13]. In contrast, flexible sensors can cover and wrap around parts of a robot, as for example the iCub tactile skin [Schmitz 11]. Regardless the structural flexibility, the surface of a tactile sensing array can be hard or elastic. An elastic layer increases the range of the measurable force but decreases the sensitivity of the sensor. The measurable

force range is proportional to the maximum deformation of the sensing surface, itself proportional to the thickness of the elastic layer. However, the elastic surface acts as a low-pass filter and therefore decreases the sensitivity.

In robotics literature, elastic layers of tactile sensing arrays were added in order to compensate for the lack of precision of robot position controllers but these arrays were modeled as rigid sensors and, thus, they could not provide the information about the 3D contact frame. However, the compliance of the elastic layer can be used for governing the contact frame that occurs when an object is pressed against the elastic layer. As will be shown in Section 2.3.1, when the deformation of the elastic surface is taken into account, more information about the contact can be retrieved.

In order to govern the contact frame, a set of measures is needed. In computer vision theory, the measures that define an object frame (the frame that defines the pose of the object) are known as features. The purpose of the features is to transform images into a more compact and descriptive space, while still capturing the object frame. In physical interaction theory, they are known as *tactile features*. These tactile features are computed from a contact pattern. The computation process is referred to as feature extraction in accordance with the terminology used in computer vision. As mentioned in the previous paragraph, more data can be extracted when the elastic surface is considered. These data are the new measures that are related to the rotational compliance variables of the elastic surface. And these measures are represented by the *Zero-Moment-Point* (ZMP) tactile features proposed in this thesis.

Thus, this chapter will explain how the new tactile features are computed from the deformations of the elastic surface of a pressure sensing array. In the following, the state-of-the-art in tactile feature extraction is given first. Then a contact frame between the sensing array and an object is defined. The pressure sensing array is modeled as a rigid image sensor combined with a compliant force sensor. The contact frame and the deformations of the sensing surface are estimated as a set of tactile features. These features are then described and validated in the last section of the chapter. The tactile features will be the primary inputs in the developed tactile servoing framework (Fig. 3.1).

2.2 State of the art in tactile feature extraction

Research in pressure sensing arrays has attracted increasing attention in recent decades. Technological advances in these sensing devices [Sekitani 10, Mittendorfer 11] consequently stimulate the interest in tactile data processing. In regard to different applications (including robotic manipulation [Nicosia 01, Howe 94], grasping [Romano 11], object exploration and recognition [Pezzementi 11], texture classification [Xu 13], slip detection [Göger 09]), different signal processing techniques have been applied to tactile signals in order to extract *tactile features*. These feature extraction techniques can be classified in two main categories: analysis in the frequency domain and spatial analysis.

The tactile feature extraction techniques remind the operation of the humans' system of mechanoreception. There are a lot of mechanoreceptors distributed in the fingers and the palm of the human hand. According to Johansson [Johansson 09] the mechanoreceptors can be classified into fast adapting (FA) and slowly adapting (SA) mechanoreceptors. The FA and SA mechanoreceptors are suitable for measuring skin vibrations and to detect static stimuli, respectively. The nature has always been driving researchers in many different fields, especially, robotics. Romano et al. [Romano 11] tried to mimic the human system of mechanoreceptors by extracting two tactile features from the tactile sensing arrays attached to the grippers of the PR2 robot. The first tactile feature is an estimate of the total normal force applied by the gripper to a grasped object. It is the sum of the forces measured by all tactels of a sensor. The second tactile feature is an estimate of the force disturbances that appear at frequencies higher than 5 Hz. The force measured in each tactel of the sensor was subjected to a discrete-time first order Butterworth high-pass filter designed for the 24.4 Hz sampling rate of the force signals. This second feature was extracted to allow the robot to avoid the slip of the grasped object. When the amplitude of the high-passed force signals exceeded some level, the slip event was triggered. The contact frame between the grasped object and the grippers was not of importance as the authors used it in an autonomous pick-and-place operation.

The second tactile feature in the above approach was derived by filtering the measured signals in the time domain. The tactile signals can also be analyzed in the frequency domain. Cutkosky et al. [Cutkosky 14] proposed a tactile feature that allows

a robot grasping an object to distinguish between two types of sliding: sliding of the grasped object over a planar sensing surface and sliding of the grasped object over a surface of the environment. The tactile feature is a ratio between spectral powers of the pressure oscillations at a single tactel and at a set of tactels. A spectral power of tactile responses was also adopted as a tactile feature by Loeb et al. [Fishel 12] in order to recognize surface textures. These last two tactile features were used for classification purposes. However, the Fourier Transform provides no information on the time when frequencies occur. Therefore, in order to prevent sliding motions or slip, Göger et al. [Göger 09] and Teshigawara et al. [Teshigawara 11] used the short time Fourier transform and the discrete wavelet transform, respectively, in order to experimentally observe the feature values at which slip occurs.

The latter features were derived by analyzing signals from the pressure arrays in the frequency (time-frequency in [Göger 09]) domain. In order to obtain reliable features in the frequency domain, approaches require electronics (DAQ - data acquisition boards) with a rather high sampling rate, for example a DAQ with 10 kHZ was used in [Teshigawara 11]. Back in time when electronics were less efficient, Kyberd et al. [Kyberd 92] detected the slip of a grasped object by considering changes in a tactile array in the following way: an increase (decrease) of force was represented by 1 (-1) and no change by zero. By differentiation of the matrix containing the corresponding elements (-1, 0, 1), the author obtained gradients showing the direction of the slippage. Thus, this tactile feature can also detect a slippage in the spatial domaing. The pressure sensing array used by the authors was constructed from Force Sensing Resistors (a polymer thick film) arranged in a 4×4 array. By substituting the changes in each tactel by the three integer numbers, the authors artificially transformed the array into a dynamic *image* similar to the nowadays Dynamic Vision Sensors [iniLabs 15]. Therefore, they tracked changes in the contact frame.

2.2.1 Contact frame features

The tactile features discussed so far are suitable for event-driven manipulation to detect contact with the environment (light touch), or slip, or to distinguish between sliding

and rolling motions of a manipulated object as shown in [Howe 94]. However, tactile features that represent the contact frame are needed to control a robot interacting with the environment. For that, signals from pressure sensing arrays can be analyzed in the spatial domain. And the spatial distribution of the pressures/forces sensed by tactels can be two or three dimensional because the arrays can have different shapes, including spherical [Koiva 13], cylindrical [Briot 79] and flat (planar) [Berger 91].

Non-planar stiff sensors cannot provide measures of edge contact from a single snapshot. Even though researchers in human tactile perception and haptic applications [Platkiewicz 16] argue that edges are detected through shear strain of the human skin, the state-of-the-art tactile sensors do not measure shear strains of the sensors' contact surface. Moreover, nowadays planar pressure sensing arrays can measure normal forces only. But this is enough to provide the information about a contact frame for the following types of contact that can happen between two bodies: edge- (line) and vertex- (point) on plane. Distinguishing these two types of contact, i.e. whether the finger is touching a corner or an edge, is of utmost importance in closed-force grasping approaches as mentioned by Fearing et al. [Fearing 85]. This thesis is therefore focused on the edge and point types of contact only. The techniques to extract tactile features for these two types of contact are discussed in the following.

Since the time when the technology in tactile sensing became advanced enough so that the spatial resolution of the planar pressure sensing arrays was sufficient to get tactile images [Overton 81], studies on extraction of tactile features representing the contact frame have gained attention thanks to robot applications involving grasping/ manipulation and advances in robot gripper/hand design. Early works using pressure arrays were presented by Montana [Montana 88] and Fearing [Fearing 85]. Whereas Montana relied on contact kinematics and suggested finding the surface curvature by rolling a planar sensor about a contact, Fearing applied a linear elastic model to predict strains beneath a compliant surface (skin) for several cases, including finger touching a sharp edge. In both works the extracted tactile features were represented by the surface curvatures of an object under external forces.

Other researchers have taken a different approach. Pressure sensing arrays can pro-

vide a "tactile image" or contact profile of an object. In other words, the usage for such sensors is to record the static image produced by an object. Therefore, previous works used vision like algorithms to extract tactile features identifying the location, orientation, and shape of the object. The first approaches were heading to detect an edge in a tactile image by extending some of the existing detection methods in vision systems to tactile images. Muthukrishnan et al [Muthukrishnan 87] applied a median filter. The filter replaces a center taxel in a 3×3 kernel by the median of the taxels in the kernel. This filter is usually applied to eliminate noise. A thresholding operation and a contour tracking algorithm were applied. Since the computational power of computers was much more limited at those times than nowadays, the processing time of such a small array of data was of concern. The computational costs were important, especially when the processed signals were used in real-time applications. Thus, Berger et al. [Berger 91] proposed a faster algorithm by applying an automatic threshold operation, edge detection and Hough Transformation to fit a line in the sensor image and finally yield the orientation of this line with respect to the sensor's coordinate frame. Since the tactile image was thresholded, a total force was represented by the area of the sensor covered by the object. The relationship between the estimated and real forces was not discussed.

With emphasis on the drawbacks of the thresholding operation, which could lead to large errors when the threshold is wrong, Chen and Zhang [Chen 95] proposed a more direct approach based on the analysis of the subsurface strain and stress. Under a number of assumptions, the authors derived the following tactile features: the normal force, relative contact location, and contact orientation. They presented tactile feature extraction algorithms for the point and edge contact types. They analyzed a 2-D stress function $\sigma(x, y, d)$, where x, y are the coordinates of a tactel and d is the depth to the tactel as shown in Fig. 2.3 a, and its moments in the spatial domain $M_{x^p,y^q} = \int\int x^p y^q \sigma(x, y, d)\, dx\, dy$, where $(p + q)$ defines the order of a moment. The stress function for each type of contact is different and omitted here. The important results of the proposed feature extraction were the following:

- for the case of the point contact (Fig. 2.3b), a set of tactile features can be either given by $[a, b, P]$ for the frictionless contact or $[Q_x, Q_y, P]$ for the contact with friction,

where P is a normal force, (a, b) is tactile contact point location, Q_x, Q_y are the tangential forces: $Q_x = \frac{\sqrt{(dM_0)^2+M_x^2-M_0M_{xx}}}{d}$, $Q_y = \frac{\sqrt{(dM_0)^2+M_y^2-M_0M_{yy}}}{d}$

- for the case of the edge contact (Fig. 2.3c, d), a set of tactile features $[a, \alpha, P]$, where $(a = \frac{M_x}{M_0}, 0, 0)$ is the assumed intersection point of the edge with the x axis, $P = -\frac{1}{2L}M_0 \cos\alpha$ is the normal force, and $\alpha = \frac{1}{2}tan\left(\frac{-M_{xy}}{M_{xx}} - M_{yy} - 2aM_x - a^2M_0\right)$ is the angle of the edge [Zhang 00].

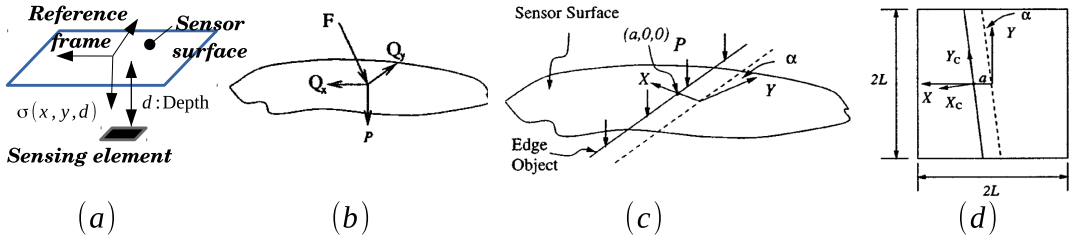


Figure 2.3: (a) Reference frame for defining a stress distribution function. (b) Point-contact. An edge in contact with a planar tactile sensor in (c) 3D and (d) 2D. Images are modified from [Zhang 00].

The edge was assumed to be infinitely long and have intersects with two opposite corners of the tactile sensor. Thus, there were many constraints to cope with in this approach. This approach modeled the relation of the contact frame and expected image moments; this allowed the next researchers to extract tactile features using image processing techniques.

Computations of strain and stress distributions are still complex and a number of physical constraints must be satisfied. Therefore, image processing techniques, including image moments, can be applied to extract tactile features. Following this logic, Ho et al [Ho 12] derived quantitative information about the location and orientation (for the edge-contact type) of a contact frame by exploiting image moments defined as:

$$M_{i,j} = \sum_{x=1}^{N_x}\sum_{y=1}^{N_y} x^i y^j I(x,y) \tag{2.1}$$

where $M_{i,j}$ is the i, j-th order image moment; N_x and N_y are the number of pixels along x and y axes, respectively; $I(x, y)$ is the intensity of the (x, y) pixel, x, y are the coordinates of each tactel in the image. When $i + j = 0$, the order of the image moment is zero and

the moment is the zeroth moment. When the sum is equal to 1, the moment is the first order and so on. The features were the following:

- the centroid of the contact area:

$$CoP_x = \frac{M_{1,0}}{M_{0,0}} \tag{2.2}$$

where $M_{1,0}$ is the first moment with respect to the x axis, $M_{0,0}$ is the zero moment, CoP_x is the center of pressure in the x direction. The second tactile feature is along y axis:

$$CoP_y = \frac{M_{0,1}}{M_{0,0}} \tag{2.3}$$

where $M_{0,1}$ is the first moment with respect to the y axis, CoP_y is the center of the pressure in y direction. And M_{00} is proportional to the total force:

$$f \propto M_{0,0} \tag{2.4}$$

- the orientation of the contact area:

$$\alpha = \frac{1}{2}\arctan(\frac{2\mu_{11}}{\mu_{20}}\mu 02) \tag{2.5}$$

is computed through central moments:

$$\mu_{pq} = \sum_x \sum_y (x - x_0)^p (y - y_0)^q I(x, y) \tag{2.6}$$

The sensing array had a rather high resolution of 16 by 16 tactels. However, when the spatial resolution of sensing arrays is not high enough, a tactile image can be resized and augmented by various interpolation methods. For example, in [Liu 12a], a tactile image was resized from a 5 $\times$ 9 to a 12 $\times$20; then tactile features were extracted with a higher accuracy. Li et al [Li 13a] showed that the computation of the centers of contact by the image moments provides a sub-tactel (tactel's size divided by two) resolution due to the averaging effect from multiple tactels composing a contact region. Therefore, the artificial augmentation by a factor two is still reasonable. The authors derived the same features

for the point contact type: the total force was given by the zeroth moment M_{00} and the centers x_0 and y_0 of contact along x and y axes were given by eq. (2.2) and eq. (2.3), respectively. But in the case of the edge contact, the authors applied the Principal Component Analysis (PCA), so that the main eigenvector of the PCA represented the orientation of the edge with respect to the coordinate axes of the sensing frame. The sensor did not have compliant surface as in the latter approach.

Thus far, the following tactile features were derived:

- point contact: $[x_0, y_0, f, -, -, -]^T$. Fig. 2.4a illustrates a spherical object pressed against a pressure sensing array and Fig. 2.4b the corresponding sensor surface (up) with pressure profile (bottom) on the left-hand side and gray scale image of the pressure profile on the right-hand side.

- edge contact: $[x_0, y_0, f, -, -, \alpha]^T$ Fig. 2.4c illustrates a cylindrical object pressed against a pressure sensing array and Fig. 2.4d the corresponding sensor surface (up) with pressure profile (bottom) on the left-hand side and gray scale image of the pressure profile on the right-hand side.

As mentioned before, any object's pose can be defined with three position (x, y, z) and three orientation (w_x, w_y, w_z)[1] variables $[x, y, z, w_x, w_y, w_z]^T$ with respect to a chosen frame in the world, for example, the sensor frame. A point contact can be localized by the three coordinates $[x, y, z]$ and the three tactile features representing these coordinates derived. In contrast, an edge contact can be localized by all six coordinates $[x, y, z, w_x, w_y, w_z]^T$ since the edge can also apply a moment of force around axes lying in the planar surface and, therefore, can cause rotational deformations of the sensing surface when the surface is elastic. However, only four tactile features have been derived until now to describe the contact frame in the case of the edge contact. This is due to the following reasons:

- In [Muthukrishnan 87] and [Berger 91] the tactile image was thresholded before applying feature extraction algorithms, so the deformations of the sensing surface due to an external moment of force applied by an edge could not be estimated. Moreover, an elastic layer was added to only compensate for the imperfections of

[1] minimal representation using Euler angles

the motion controller of the robot equipped with the sensor. This layer caused undesirable cross-talk which was mitigated by the thresholding.

- In [Zhang 00] due to the complexity of calculations, it was assumed that the applied forces along an edge were distributed equally.

- In [Li 13a] and [Ho 12] the tactile pressure sensing array was rigid and did not incorporate an elastic surface.

In summary, it was always assumed that the force was distributed equally along the edge (Fig. 2.4e). However, in the presence of an elastic surface on the array the edge can be inclined due to the compliance of the elastic layer (Fig. 2.4f). In order to estimate this inclination a tactile feature is needed, but had not been presented yet. It is undoubtedly the main weakness of the approaches developed so far.

In order to overcome this drawback, a new algorithm to extract a new tactile feature is proposed in this chapter. It will be shown in Chapter 4 (devoted to the applications of a touch driven robot) that this algorithm is robust and allows a robot to perform different complex tasks continuously with the same controller. The proposed algorithm considers the compliance of the elastic surface in order to extract the new tactile feature. It is based on the calculation of the distance between the Center of Pressure and the Center of Contact as it will be explained in the following. This distance represents a moment of force, which in turn is related to rotational displacements through the rotational compliance constants of the pressure sensing array. In accordance with the terminology conventionally used in the legged locomotion theory, the proposed tactile feature is referred to as **Zero-Moment-Point** (ZMP) feature. The ZMP represents the point where a whole body does not apply a moment of force on the walking surface.

2.3 Extraction of tactile contact features

The final target of this chapter is to derive tactile features that can be further used by a robot to interact with the environment. In general, interaction tasks involve dealing with position and force constraints in the robot's link that is in contact. This robot's link

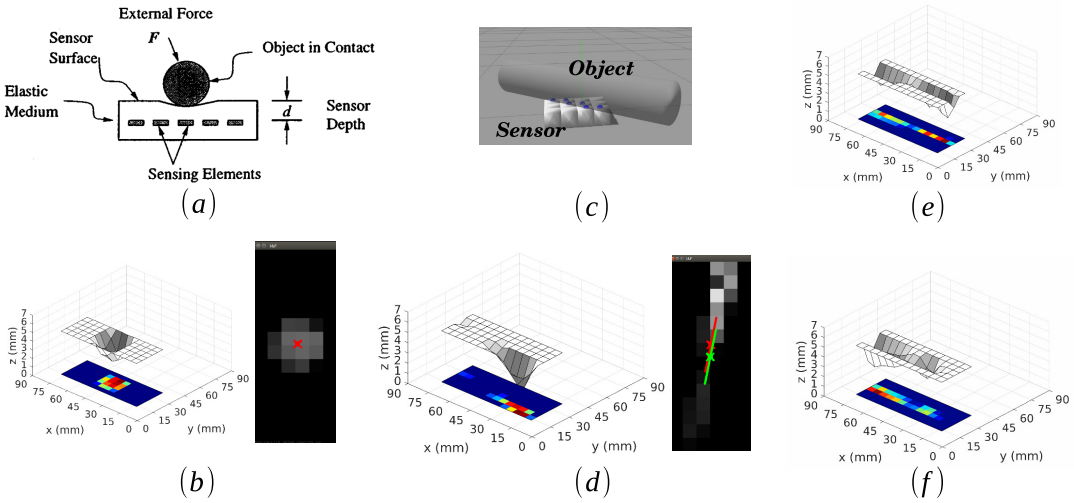


Figure 2.4: (a) Point contact [Zhang 00] and (b) the corresponding deformation of the sensor surface (up) with the pressure profile (bottom) on the left-hand side and gray scale image of the pressure profile on the right-hand side. (c) Edge contact and (d) the corresponding deformation of the sensor surface (up) with the pressure profile (bottom) on the left-hand side and gray scale image of the pressure profile on the right-hand side. (e) The force applied by an object is distributed equally along the edge and (f) not equally.

has a pressure sensing array that will produce a two-dimensional pressure profile $I(x, y)$. The constraints are defined by a contact frame between this link and the environment. Therefore, the exact definition of the contact frame and its parameters is crucial. These parameters include the local contact information such as the magnitude and direction of the contact force, and the contact location and orientation with respect to the pressure sensor. Moreover, an on-line tactile image processing is necessary to calculate this local contact information. Therefore, this section will describe the contact frame parameters and the feature extraction algorithm for the point-contact type first and then for the edge-contact type.

For each types of contact, the impact of an elastic layer will be shown. Therefore, contacts with and without the elastic layer will be considered. This elastic layer is generally added on top of sensing elements of a pressure sensing array. Besides the elastic layer, the sensing elements can also deform under the normal force. Even though the sensing elements can deform, the lengths of deformations of the elements are so small that they can be neglected. The deformations are small because these sensing elements measure a strain that is in the order of micro-meters. Therefore, when the elastic layer is not added,

the sensing surface will be considered as rigid. When the layer is added, the elasticity is assumed to be given by the elastic layer only. In the presence of the mechanically compliant layer[2], the sensor can be modeled as a general spring. A vector of compliance constants will be shown in the following. Deformations of the sensing surface will be estimated through tactile image moments, which also will appear in the following.

Finally, the goal of this section is to construct vectors of tactile features that represent the contact frames in the Cartesian space for the two types of contact discussed so far: edge and point. These vectors will be used in the tactile servoing controller (next Chapter) to move a robot.

2.3.1 Point-contact

There are many cases in which the contact between the sensing surface and an object is of point-contact type, e.g., sharp objects, spherical objects or corners of rectangular shape objects. The contact frame is described for the contact without an elastic layer, then with an elastic layer. It was assumed that shear forces can be neglected.

Tactile sensor as an image sensor

Fig. 2.5a and b show from the side and top views, respectively, a stiff spherical object pressing a sensing array without a compliant surface. The origin of the coordinate axes of the sensing frame $\{s\}$ lies at the geometrical center of the sensor on the sensing surface. The stiffness of the sensing surface is high enough so that interaction forces during manipulation do not cause deformations. When the stiff object applies a normal force on the sensing surface, two components of the location of the contact frame $\{c\}$ are captured by the pressure array: x_c and y_c are the x- and y-components of the location. The z-component z_c of the location coincides with the sensing surface, and is therefore equal to zero. The normal force does not cause any deformation of the sensing surface. Since

[2]This may reduce the sensitivity of the sensors. However, the recent results in tactile skin development conducted at the Bielefeld robotics laboratory, Germany [Büscher 15] have shown that by placing the sensing layer on top of the skin, one can achieve a higher sensitivity compared to the arrangement when the sensing layer is at the bottom.

the object touches the sensing surface at a single point only, the measurable force range is low; it is limited to the dynamic range of the tactel in contact.

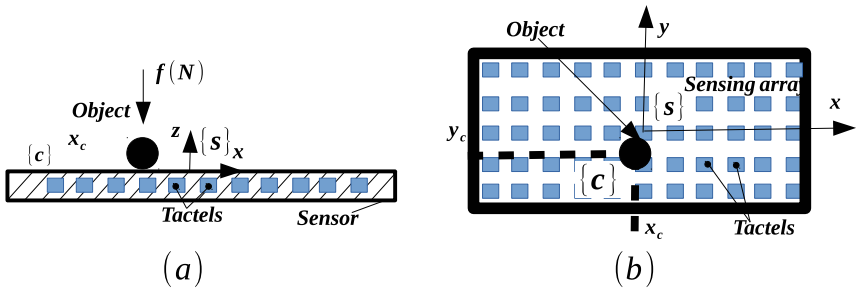


Figure 2.5: A point contact with a pressure array without an elastic surface. The tactile sensing array works as an image sensor. An object under external forces on the sensing surface is shown from (a) the side view and (b) the top view. Coordinates x_c and y_c of a contact frame $\{c\}$ are defined with respect to the tactile sensing frame $\{s\}$.

Tactile sensor as a compliant force sensor

Consider the case when the same stiff spherical object applies a normal force onthe same sensing array, but at this time the sensor has an elastic layer (Fig. 2.6a). The origin of the sensing frame is still located at the geometric center of the sensing surface, but now lies in the elastic layer along z axis as shown in Fig. 2.6a. The layer has the compliance

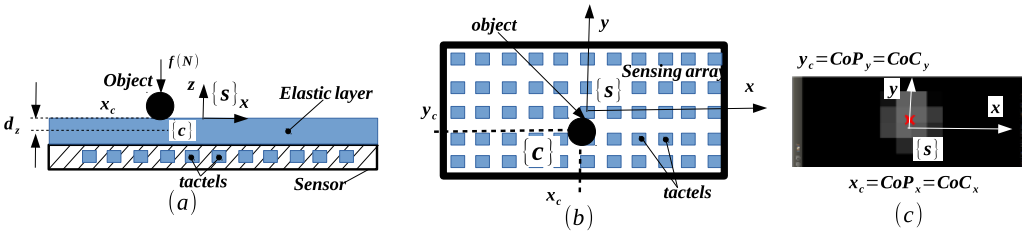


Figure 2.6: A point-contact with a pressure array with elastic surface. An object under external force deforms the sensing surface, which is shown from (a) the side view and (b) the top view. (c) Gray scale representation of the pressure distribution. Coordinates x_c, y_c and d_z of the contact frame $\{c\}$ are defined as the center of the area of contact with respect to the tactile sensing frame $\{s\}$.

constant C_z along z axis. Then the deflection d_z (see Fig. 2.6a) of the contact frame along this axis is given by:

$$d_z = C_z \cdot f \tag{2.7}$$

where f is the total force applied by the object. Given a pressure P and the surface area of one tactel A, the force is given by:

$$f = P \cdot A \tag{2.8}$$

The amount of the applied force also affects the area of contact as shown in Fig. 2.7. The coordinates (x_c, y_c) of the contact frame are now represented by the center of this contact area. How to find this center of the contact area and the force will be explained next.

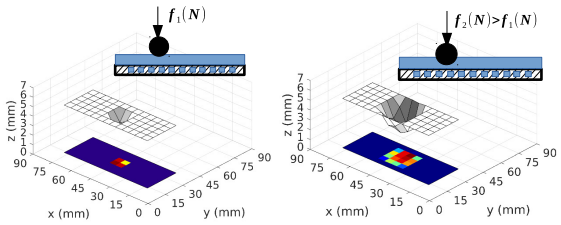


Figure 2.7: The area of contact increases with the increase of the applied force by an object.

Tactile feature extraction

The gray scale representation is one possible form for the display of the pressure distribution. Pressures/forces produced at all tactels are proportional to the deflection of the sensing surface given in eq. (2.7). It is then reasonable to plot the data as an image where the intensity of a pixel corresponding to an individual tactel is proportional to the pressure/force at that location (Fig. 2.6c). From the resulting gray scale image, we need to extract the parameters of the contact frame: the contact position (center of contact), and the contact force.

To this end, we need image processing methods providing the required tactile features. As mentioned in Section 2.2, the image moment analysis was applied to compute tactile features from the image moments as shown in eq. (2.1). The first tactile feature represents the center of the contact frame along x axis (CoP_x) and is given by the relation of the first and zeroth image moments as shown in eq. (2.2). The second tactile feature is along y axis (CoP_y) as given in eq. (2.3). Thereby, the CoP is defined as the force-weighted center of pressure of the contact. Red dot in Fig. 2.6c illustrates these two

tactile features that represent two coordinates (x_c, y_c) of the corresponding contact frame shown in Fig. 2.6b. The third coordinate $z_c = d_z$ is proportional to the total normal force and shows how deep the contact frame is deformed from a resting state. This force can be estimated by the zeroth moment as given in eq. (2.4) or by the area of contact between the object and the sensing surface (Fig. 2.7). The area is a reasonable estimate since the number of points of contact increases with the total force. Therefore, the origin of the contact frame for the point contact type is given by the center of pressure and deflection of the surface from the resting state:

$$_{\{s\}}O_c = [CoP_x, CoP_y, d_z]^T \tag{2.9}$$

which is in fact the 3×1 vector defining the contact frame in the Cartesian space.

2.3.2 Edge-contact

The next step is to extract tactile features representing the contact frame for the edge type of contact. We will see that the elasticity of the sensing surface becomes very useful. Similar to the point contact type, the edge contact type occurs in a lot of cases. For example, during the blind exploration of objects with rectangular and cylindrical shapes.

Tactile sensor as an image sensor

Fig. 2.8a and b show from the side and top views, respectively, a stiff cylindrical object pushing a sensing array without compliant surface. The origin of the coordinate axes of the sensing frame $\{s\}$ lies at the geometrical center of the sensor on the sensing surface. When the stiff cylinder applies a normal force on the sensing surface, there are multiple tactels in contact with it. They create a contact region, whose center represents the origin of the contact frame $\{c\}$.

There is one more parameter that defines the contact frame: the orientation of the edge about z-axis. In Fig. 2.8b, $\vec{O_cO'_c}$ is the unit vector representing the axis of the edge. The angle α_z between this axis and the x-axis of the sensing frame represents the orientation of the edge about the z-axis of the sensing frame.

The normal force does not cause any deformation of the sensing surface. Therefore, the z-component z_c of the contact frame coincides with the sensing surface and is equal to zero. A tiny roughness of the surfaces in contact may make the pressure array erroneously detect a single point contact instead of an edge. Therefore, mechanical compliance of the sensing surface was necessary as reported in previous works, e.g. in [Berger 91].

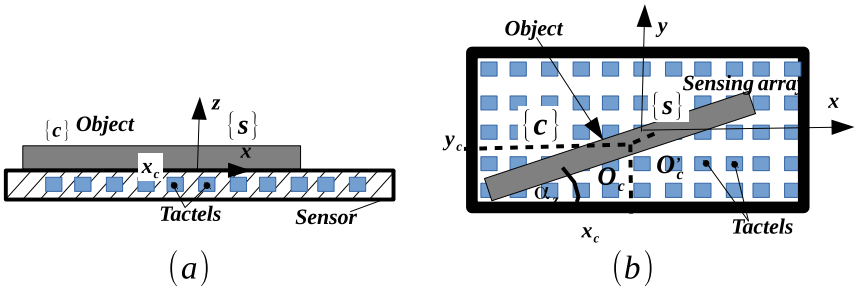


Figure 2.8: An edge-contact with a pressure array without elastic surface. The tactile sensing array works as an image sensor. An object under external forces on the sensing surface is shown from the (a) side view and (b) the top view. Coordinates x_c and y_c of the contact frame $\{c\}$ are defined with respect to the tactile sensing frame $\{s\}$.

Tactile sensor as a compliant force sensor

As in the case of the point-contact, we are now considering the impact of an elastic layer on the sensing surface. Fig. 2.9a and b show the side and top views, respectively, of the contact between a stiff cylinder and an elastic surface. The origin of the sensing frame is defined so that it lies on the contact surface when the latter does not deform. The only component of the contact that changes in the present situation is the position z_c along the z axis of the sensing frame. The rest three components x_c, y_c, α_z remain the same as for the rigid sensor. The position along z-axis is given by the deflection from the resting state of the layer (eq. (2.7)).

We assumed that the cylindrical object was under a uniformly distributed force vector f as shown in Fig. 2.9a. The solution for the non-uniform case will be presented in Section 2.3.3. However, we should, first, find the tactile features that represent the contact frame of the edge under a uniform external force distribution.

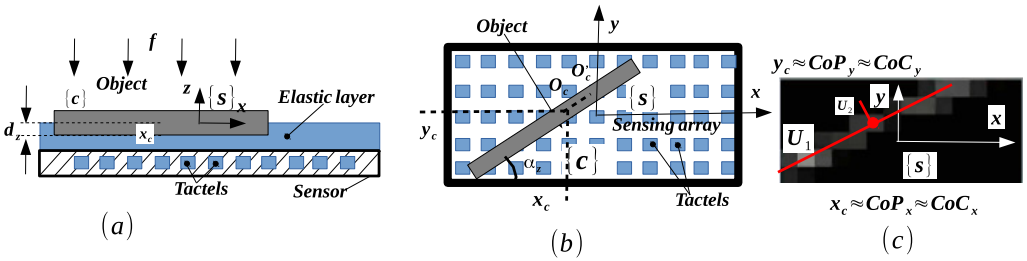


Figure 2.9: An edge-contact with a pressure array with elastic surface. An object under external force deforms the sensing surface, which is shown from (a) the side view and (b) the top view. (c) Gray scale representation of the pressure distribution. Coordinates x_c, y_c and d_z of the contact frame $\{c\}$ are the center of the area of contact with respect to the tactile sensing frame defined as $\{s\}$.

Tactile feature extraction

The edge is a very informative characteristic as many objects have edges or cylindrical shapes. In artificial vision, researchers have spent a lot of efforts to detect edges and identify their parameters in camera images. Let us assume that we know in advance that there is an edge in contact with our pressure array. Our task is to extract the contact force, and the location and orientation of the line of contact from the tactile image. To simplify the problem, we already assumed that the force applied by the edge has a uniform distribution. Again, as for the point-contact, the tactile image moments are derived to define the contact frame. Then two components of the location of the contact frame x_c and y_c are given by the equations eq. (2.2) and eq. (2.3), respectively. The contact force is represented by the zeroth moment $M_{0,0}$ (2.1). Due to the translational deformation of the contact frame, z_c is given by eq. (2.7).

The next tactile feature to calculate is the orientation of the edge relative to the sensor array about z-axis. Let us use the Principal Component Analysis (PCA, appendix A) for extracting the orientation of the edge. First, the acquired contact image $I(x, y)$ is thresholded with an appropriate threshold (i.e. the sensor value that is higher than the noise of the sensor). The orientation of the edge is given by the principle component representing the direction in which the contact points are spread the most (Fig. 2.10). This principal component U_1 the eigenvector having the highest eigenvalue λ_1.

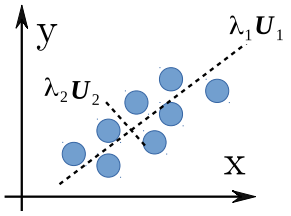


Figure 2.10: Principle Component Analysis. Two orthogonal eigen vectors U_1 and U_2 and their corresponding eigen values λ_1 and λ_2 with $\lambda_1 > \lambda_2$ are obtained.

Then the orientation of the edge is given by:

$$\alpha_z = \tan^{-1}(\frac{U_1}{\mathbf{e}_x}) \tag{2.10}$$

where $\mathbf{e_x}$ is the basis vector defining the x component of the sensing frame. Fig. 2.9c illustrates the four tactile features derived so far for the edge-contact type. Thus, the origin of the contact frame for the edge-contact type is given by the center of pressure, the deflection of the surface from the resting state, and the orientation of the edge about the z axis of the sensing frame:

$$_{\{s\}}O_c = [CoP_x, CoP_y, d_z, \alpha_z]^T \tag{2.11}$$

which is in fact a 4×1 vector defining the contact frame in Cartesian space.

2.3.3 Edge-contact with non-uniform load

So far, it was assumed that the pressure sensing array was in contact with the cylindrical object that was under an external force with a uniform distribution along the edge (Fig. 2.9a). Let us consider now the case when the line force applied onto the edge is non-uniform as shown in Fig. 2.11.

An important question arises: can the **angle of inclination** (d_{wy} or d_{wx}), location (c_x, c_y), total force (d_z), and orientation (α_z) be recovered from the flat pressure sensing array that measures only normal pressures? So, instead of 4 tactile features, there will be 6 features defining the contact frame in the 3D Cartesian space together with rotations.

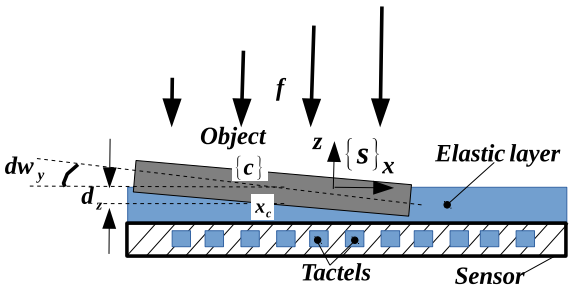


Figure 2.11: An edge-contact with a pressure array with elastic surface. An object under an external line load with a non-uniform distribution. The line load force deforms the sensing surface in such a way that it causes inclinations of the contact frame with respect to the tactile sensing frame $\{s\}$.

3D spring as a tool to determine the moment of force (force angle) for a line load

Eq. (2.7) shows that the deformation d_z of the sensing frame is proportional to the normal force and the compliance constant C_z along the z axis of the sensing frame (Fig. 2.9a and Fig. 2.6a). Thus, the z coordinate of the contact frame depends on the translational compliance constant of the mechanically compliant sensing surface. In the case of a non-uniform line load, the contact frame with the object is inclined with respect to the sensing surface (as shown in Fig. 2.11), i.e. there is a moment of force. This fact suggests taking into account rotational compliance constants about the y and x axes. Fig. 2.11 illustrates the inclination about the y axis. However, the line load can be non-uniform about the x axis too. We, therefore, take into account the translational compliance constant along z axis and the two rotational compliance constants, and we model the elastic surface as a general 3D spring with the following vector of compliance constants:

$$\boldsymbol{C} = [0, 0, C_z, C_{w_x}, C_{w_y}, 0]^T \tag{2.12}$$

where C is the vector of compliance constants and C_z, C_{wx} and C_{wy} are the translation compliance constant in z direction, rotational compliance constants about x and y axes, respectively. It is assumed that friction forces are relatively small (negligible). So, the deformations along z-axis and about the x and y axes, are defined by the compliance

constants:

$$ {}_{\{s\}}\boldsymbol{dO_c} = \begin{bmatrix} 0 \\ 0 \\ dz_c \\ dwx_c \\ dwy_c \\ 0 \end{bmatrix} = \boldsymbol{C} \cdot \boldsymbol{f}^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & C_z & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{wx} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{wy} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ f_z \\ \mu_x \\ \mu_y \\ 0 \end{bmatrix}^T \tag{2.13} $$

where ${}_{\{s\}}\boldsymbol{dO_c}$ represents for the declination with respect to the sensing frame $\{s\}$ of the contact frame from its resting state due to the normal force f_z, and the moments of force around x axis μ_x and y axis μ_y.

The rotational compliance constants can be experimentally defined from calibration. One question remains: how to calculate the moment of force around x or y axis? Computing this moment of force would involve the analysis of a 3D matrix (tensor) [Fearing 85]. And previous methods required a precise model of the material of the sensor and a lot of time to compute. Therefore, this thesis proposes an optimized algorithm that can be executed on real-time to compute a moment of force with a small loss of precision.

Zero-Moment-Point feature

In the legged robot locomotion theory, there is a point called "Zero Moment Point" or simply ZMP. This ZMP is the point on the ground where the total moment generated by gravity and inertia equals zero [Vukobratovi 90]. In other words, the ZMP represents the point on a planar surface, for example the foot of a robot, where the robot rests at its equilibrium without applying efforts to the joints of the foot. In this thesis we adopt this concept for the tactile features related to robot manipulation and various robot arm applications. First, the CoP_x and CoP_y tactile features are derived according to eq. (2.2) and (2.3), respectively. Then the original tactile image is binarized with the lowest possible thresholding level. The threshold value can be experimentally[3] tuned. After the binarization, the tactile image represents the area of contact: any tactel that is in contact

[3]However, algorithms can be applied to automatically select the threshold value, e.g. a histogram shape-based method.

with the environment gets the maximum grey-scale value. In this binarized image, the first order moments $M_{1,0}$ and $M_{0,1}$ are calculated according to eq. (2.2) and (2.3), respectively. Since the tactile image is binarized (i.e. the pressure value of each tactel in the sensing array is equal to zero (no contact) or one (contact)), these equations represent the Center of Contact (CoC_x along x axis and CoC_y along y axis) not the Center of Pressure of the contact pattern. Before the image was binarized, the image moments represented the coordinates of the Center of Pressure because each tactel kept the actual pressure value (i.e. the resulting image moment is weighted towards areas with higher pressure values). The Euclidean distance between these two different centers (CoP and CoC) represent the rotational displacement of the contact frame due to the applied moment of force:

$$\begin{aligned} dw_x &\propto CoP_y - CoC_y \\ dw_y &\propto CoP_x - CoC_x \end{aligned} \tag{2.14}$$

where dw_x and dw_y are the rotational displacements from the resting state around x and y axes, respectively.

Nevertheless, a tactile sensing array cannot directly measure the moment of force and the rotational compliances of the elastic surface of the sensor are not given explicitly. In fact, the rotational constants C_{wx} and C_{wy} exist due to the translational compliance along the z axis of every tactel in the sensing array. In other words, the rotational displacements are generated when there are different displacements along z axis on different tactels in the sensing array, i.e., one side of the sensor can be more compressed than the other side because of the compliance. This physical relationship between rotational compliance and translational compliance is explained next and the proof of the proposed method for tactile feature extraction provided.

Relation with the physics of the contact frame

In the above section, since the tactile sensing array was modeled as a compliant contact frame measuring device, the rotational deformations at the contact frame were caused by an applied moment of force. This moment of force caused a displacement between

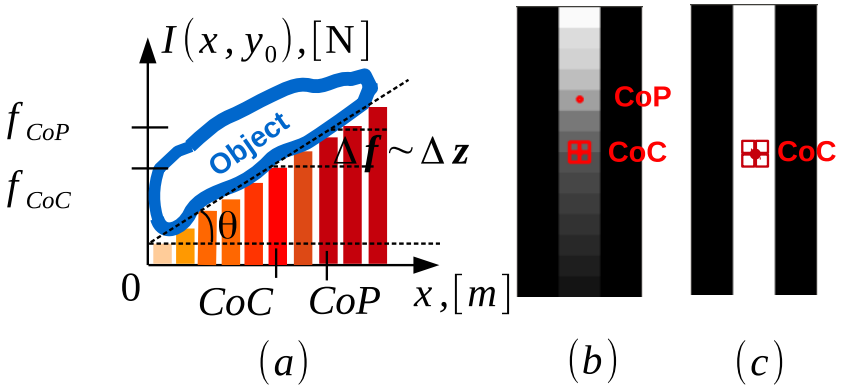


Figure 2.12: Generation of a moment of force by a non-uniform line force in a "one-dimensional" sensing array: (a) pressure distribution along x-axis; (b) tactile image – white color represents a higher pressure – with the calculated Center of Pressure, CoP, and the Center of Contact, CoC; (c) binarized tactile image that is used to compute CoC. f_{CoP} and f_{CoC} are the forces at the CoP and CoC, respectively. $\theta = d_{wy}$ is the rotational displacement caused by the applied moment of force μ_y. Δz is the Euclidean distance along z-axis between the force values at the points with the coordinates at CoP and CoC. Δf is the difference of the forces measured at the points CoP and CoC.

the center of pressure CoP and center of contact CoC. When the moment of force is zero, the Center of Pressure and the Center of Contact coincide. As the tactile sensing arrays measure the pressure (force) distribution of a contact, the displacement between CoP and CoC can be computed from the measured tactile images. Let us consider a "one-dimensional" case – when a sensing array measures the pressure distribution along x-axis only – of an edge-contact (Fig. 2.12a). The moment of force around y-axis μ_y is:

$$\mu_y = (CoP_x - CoC_x) \cdot (f_{CoP_x} - f_{CoC_x}) \tag{2.15}$$

where f_{CoP} and f_{CoC} are the forces at CoP and CoC, respectively. Considering eq. 2.14, the angle of rotational displacement is then proportional to this displacement:

$$\theta \propto CoP - CoC \tag{2.16}$$

We are interested in the rotational displacement of the contact frame. Therefore, the relationship between the distance between centers CoP - CoC and the angle of rotation

should be considered. For a compliant sensor with translational stiffness k_z, the difference between the force at CoP and the force at CoC is:

$$(f_{CoP} - f_{CoC}) = k_z \cdot \Delta z \tag{2.17}$$

where Δz is the Euclidean distance in the contact profile from CoP to CoC in the z-direction (normal to the sensor frame). It is an important assumption that all tactels have the same stiffness constants along z direction. Since the compliance $C_z = k_z^{-1}$, Δz is:

$$\Delta z = C_z(f_{CoP} - f_{CoC}) \tag{2.18}$$

The angle of rotational displacement due to the moment of force μ (around the x or y axis) is:

$$\theta = \arctan(\frac{\Delta z}{CoP - CoC}) \tag{2.19}$$

Considering eq. 2.18, the angle of rotation becomes:

$$\theta = \arctan(\frac{C_z \cdot (f_{CoC} - f_{CoP})}{CoP - CoC}) \tag{2.20}$$

Given the same non-uniform line force, this angle does not change. Therefore, $(CoP - CoC) \propto (f_{CoP} - f_{CoC})$. We can then assume that the angle of deformation is proportional to $(CoP - CoC)$. The Fig. 2.12b illustrates the corresponding centers CoC and CoP extracted from a tactile image. CoC is computed from a binarized image (Fig. 2.12c).

Finally, the six features, including the proposed new features, describe all Cartesian dimensions of the contact frame of a non-uniform line load (tactile edge):

$$\begin{gathered} O_c = [x, y, z, \omega x, \omega y, \omega z]^T \sim \\ [CoC_x, CoC_y, f, \left(CoP_y - CoC_y\right), \left(CoP_x - CoC_x\right), \alpha_z]^T \end{gathered} \tag{2.21}$$

The differences $\left(CoP_y - CoC_y\right)$ and $\left(CoP_x - CoC_x\right)$ are the ZMP along y axis (ZMP_y) and

the ZMP along x axis (ZMP_x). Hence:

$$O_c = [x, y, z, \omega x, \omega y, \omega z]^T \sim [CoC_x, CoC_y, f, ZMP_y, ZMP_x, \alpha_z]^T \tag{2.22}$$

Thus, we have seen how a contact frame (point- or edge- contact) can be represented by 6 tactile features. Nevertheless, the independence of the 6 components of this feature vector should be verified in order to guarantee that the robot will be able to control the 6 dimensions of the contact (3 coordinates for position and 3 coordinates for orientation).

2.3.4 Linear independence of the components of the tactile feature vector

It is important to verify that the derived vector of features eq. (2.21) uniquely represents the contact frame, since natural and artificial constraints, which will be described in Section 3.4.4, rely on the uniqueness of the representation of a contact frame.

The uniqueness of the representation can be verified by justifying linear independence of the vectors that define the contact frame. Considering the two types of contact, including the edge and point types, the corresponding feature vectors' components are the following:

$$\mathbf{s} = O_c = [CoC_x = CoP_x, CoC_y = CoP_y, f \propto d_z]^T \tag{2.23}$$

for the point-contact type (Fig. 2.6b), and:

$$\mathbf{s}' = O_c = [CoC_x, CoC_y, f, \left(CoP_y - CoC_y\right), \left(CoP_x - CoC_x\right), \alpha_z]^T \tag{2.24}$$

for the edge-contact type (Fig. 2.9b).

For the case of the point-contact, only three features are needed to define a contact frame in the $\mathbb{R}^3$ space. The features are expressed in the standard basis (i.e. orthogonal unit vectors in Cartesian space) of the three-dimensional space that is formed by the basis vectors $\mathbf{e_x} = (1,0,0)$, $\mathbf{e_y} = (0,1,0)$, $\mathbf{e_z} = (0,0,1)$. Expressing the coordinates of the center of pressure (eq. (2.2) and (2.3)) and the force ($f = M_{0,0}$) with respect to the base

$\mathbf{e_x}, \mathbf{e_y}, \mathbf{e_z}$ the matrix formed by taking the vectors as its columns is the following:

$$\boldsymbol{A} = \begin{bmatrix} CoP_x = \frac{\sum_{x=1}^{N_x}\sum_{y=1}^{N_y} x \cdot I(x,y)}{\sum_{x=1}^{N_x}\sum_{y=1}^{N_y} I(x,y)} & 0 & 0 \\ 0 & CoP_y = \frac{\sum_{x=1}^{N_x}\sum_{y=1}^{N_y} y \cdot I(x,y)}{\sum_{x=1}^{N_x}\sum_{y=1}^{N_y} I(x,y)} & 0 \\ 0 & 0 & f = \sum_{x=1}^{N_x}\sum_{y=1}^{N_y} I(x,y) \end{bmatrix} \tag{2.25}$$

In case of contact ($\exists(x,y) : I(x,y) \neq 0$), the determinant of this matrix is non-zero. It means that the vectors are independent [4]. When the force $\sum_{x=1}^{N_x}\sum_{y=1}^{N_y} I(x,y) = 0$ there is no contact with the environment.

For the case of the edge-contact, namely, the rotations of the contact frame are added, rotation around the axes x, y, and z by angles proportional to $\sim \left(CoP_x - CoC_x\right)$, $\sim \left(CoP_y - CoC_y\right)$, and α_z, respectively. These rotation angles can be written in terms of a (3×3) rotation matrix, $\boldsymbol{Rot}$. Since each feature represents the rotation about a single axis, the matrix $\boldsymbol{Rot}$ is orthogonal, the determinant of which is always $\det \boldsymbol{Rot} = \pm 1$. And, as a linear transformation, $\boldsymbol{Rot}$ preserves the linear independence of the above base vectors.

After deriving the tactile features that are proposed in this thesis and justifying that they represent contact frames in an unique way, we will experimentally evaluate them in the next section.

2.4 Experimental evaluation of the features

The evaluation of the proposed tactile features is done firstly in a simulator and later in a real robot. Since previous simulators present important deficiencies in the representation

[4] n vectors in $\mathbb{R}^n$ are linearly independent if and only if the determinant of the matrix formed by taking the vectors as its columns is non-zero.

of the proposed tactile features to be tested, a new simulation model of a pressure sensing array is developed, which will be described in the following.

2.4.1 Tactile sensor simulation model

Simulations in this thesis were carried out using Gazebo, which is a rigid multibody kinematics and contact open-source simulator. This simulator uses the Open Dynamics Engine (ODE) [5] physics engine. The engine provides information about collisions at the points of contact between touching objects.

The simulated robot is defined by an URDF (Unified Robot Description File) that contains the kinematic structure of the robot and its joint controllers. In addition, the simulator is integrated in the ROS[6] middleware software in order to be easily connected to different robots. A CAD (Computer Aided Drawing) model of a tactile sensor (0614 WTS WEISS Robotics) is constructed and installed on the palm of the Shadow hand and arm available at our laboratory. The simulated sensor estimates the forces that occur between the sensor and the objects in the simulator taking into account the contact information provided by each tactel of the sensor, as in the real sensor. Fig. 2.13 shows a screen-shot of the simulator that supports ROS.

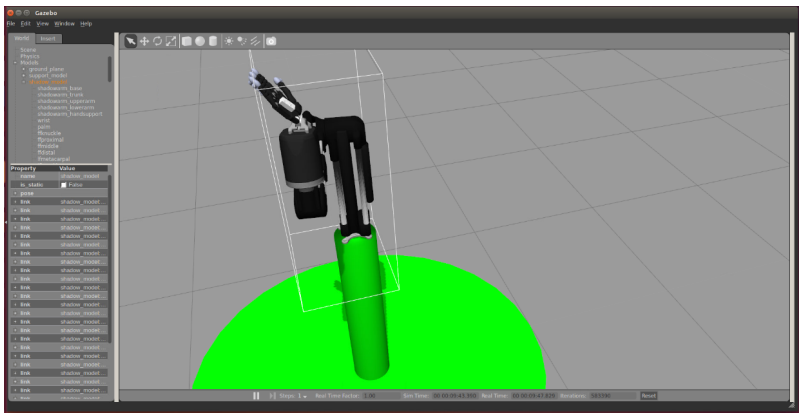


Figure 2.13: A screen-shot of the Gazebo simulator.

Most simulators restrict the number of contacts that can be simulated due to the high computational cost of the contact model (for instance, a second order mass-spring-

[5] Rigid body dynamics physical engine (http://www.ode.org/)
[6] Robot Operating System

damping system model). Since each calculated contact point increases dramatically the computation cost of the simulation, a high number of contacts can prevent its real-time execution. For this reason, Gazebo 2.2 is able to compute 32 contacts only for each body. Nevertheless, when we want to simulate a tactile sensor with a high number of tactels (for instance, 84 tactels for our sensor), this contact limitation should be bypassed so that contact features similar to the real ones can be obtained in simulation.

In order to overcome this limitation and simulate realistic tactile sensors, previous approaches propose different solutions (Table 2.1). For example, in the SkinSim [Habib 14] each tactel within a sensing array is modeled as an independent rigid body with a spherical shape. Solid tactels with a mass are attached to the base of the sensor through a virtual spring. Then the tactile image is constructed by the displacements of the springs.

The simulation model of the OpenGrasp skin [Moisio 13] includes one rigid body only. The body is represented by a triangularization of its surface. A force response at each triangle is then used to form a tactile image.

In the RobWorkSim simulator [Ellekilde 10], the authors empirically derived a polynomial function to describe the deformations of an elastic surface as a function of the distance from the point of contact towards each of the tactels of the Weiss Robotics piezoresistive sensor [Joergensen 10]. The simulator uses ODE for modeling the interaction forces.

In contrast to the above approach, Pezzementy et al. [Pezzementi 10] model a tactile array as an array of proximity sensors without inferring the physics of the material covering the sensing cells. Since a mechanically compliant surface of a real sensor creates cross-talk in a tactile image, the sensing array was characterized with a point spread function (PSF). PSF is the function that represents the spatial response of an array (image or tactile sensor) to a point (pixel or tactel) source.

These simulation models could not be directly taken to simulate the tactile sensing array used in the experiments with real robots. None of them was built for a (6×14) array. The SkinSim was presented to be used in Gazebo simulator but was under development. The RobWorkSim, OpenGrasp, and the tactile sensor simulation model built by Pezzementy could not be used in the Gazebo simulator.

Tactile Sensor Model	Sim./Engine	Friction Model	Geometry
SkinSim [Habib 14]	Gazebo/ODE	Mass-spring-damping model for each cell	Array of rigid spheres with high stiffness
OpenGrasp skin [Moisio 13]	OpenRave/ODE, BULLET, etc.	LuGre dynamic friction model	Triangularized meshes
RobWorkSim skin [Joergensen 10]	RobWorkSim/ODE	Coulomb model + deformations of elastic cover	Triangularized meshes
PSF model [Pezzementi 10]	Custom developed simulation environment	Each tactel ∼ proximity sensor without friction forces	Cubes on a planar surface
Developed model	Gazebo/ODE	Coulomb friction model	Triangularized meshes

Table 2.1: Existing simulators with the support of tactile sensor arrays.

Thus, this thesis proposes a custom sensor model that takes into account interaction forces (calculated by the ODE physical engine) and cross-talk in the tactile measurements due to the compliance of the elastic surface of the sensor. Although the developed sensor model does not directly take into account the deformations of the surface of the sensor (what would increase exponentially its computational cost), it is able to generate a tactile contact image coherent with the surface compliance of the sensor. In order to do this, a PSF based on a 2D Gaussian function is applied to the contact forces at each tactel with a convolution mask. The parameters of this Gaussian function are estimated from a calibration of the real sensor (appendix E.2)[7]. Therefore, when a force is applied directly to the rubber surface on top of a tactel, it is transmitted through the rubber to neighboring tactels leading to the following relationship:

$$I_{sim}(i,j) = F_{sim}(i,j) * g(a,b) \tag{2.26}$$

[7] In the following, the parameters of the Gaussian function are $\sigma = 0.5$ and the kernel size of 3.

where

- i and j being the coordinates of a tactile sensing cell in an NxM sensing array, $I_{sim}(i,j)$ is the resulting tactile image in the simulation
- $F_{sim}(i,j)$ is the input force profile
- $g(a,b) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2}\frac{a^2+b^2}{\sigma^2}}$ is the 2D Gaussian PSF function with σ the normal distribution and $a \times b$ the size of a kernel with $a = b$;
- $*$ denotes the convolution, i.e $I(i,j) = \sum_{a=a_{start}}^{a=a_{end}} \sum_{b=b_{btart}}^{b=b_{end}} F(i-a, j-b) g(a,b)$

Fig. 2.14 illustrates the NxM geometrical model of a tactile sensing array with $N = 6$ and $M = 14$. The physical coordinates of a cell are $x = i \cdot \Delta x$ and $y = j \cdot \Delta y$, where the size of each cell is $\Delta x = \Delta y = 3.4$ mm.

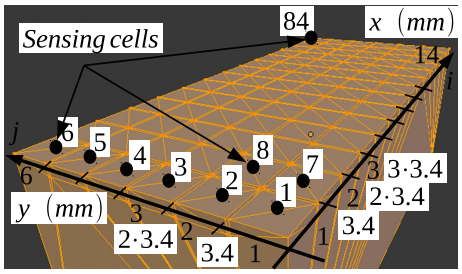


Figure 2.14: 3D model of the 6×14 WeissRobotics WTS 0614 tactile sensing array; its geometry is a triangle mesh.

The simulated sensor is added in the simulation by defining its attachment frame with respect to the robotic hand palm in URDF of the system. The term F_{sim} in eq. (2.26) is the force profile, which is built from the forces at each tactel. When the sensor collides with an object, ODE calculates the contact forces using a temporal spring-damper system at the intersection of the surface triangles (Fig. 2.14). The simulator allows small penetrations in the colliding objects[8] and estimates normal and tangential forces based on the stiffness and friction coefficients of the bodies in contact. The frictional coefficients are set to be low to allow sliding motions.

[8]ODE uses penalty methods, that are also called penetrative or soft contact models

The contact force F_{sim} at the tactel with coordinates (i_0, j_0) is computed as follows:

$$F_{sim}(i_0, j_0) = k_{scale} \sum_{x,y}^{N_x \Delta x, N_y \Delta y} F_n \exists(x, y) : \quad (2.27)$$
$$(|x - x(i_0)| \in \Delta x) \wedge (|y - y(j_0)| \in \Delta y) \wedge (|z - z(i_0, j_0)| \in h)$$

where h is the height of the sensor (in z direction), N_x and N_y are the number of tactels along x and y axes, respectively, F_n is a normal force provided by the "Bumper" plugin of the Gazebo simulator at the point of contact between an object and the given tactel. We take into account the normal forces only because the tangent forces tend to zero as the surface friction coefficients are zero. The resulting force is then normalized by a scaling factor k_{scale} to respect the maximum of the real sensor output, and down-sampled in time to make the simulated bandwidth equal to the bandwidth of the real sensor (Section 3.5).

Finally, Algorithm 1 shows how to construct a tactile image in the simulation environment.

Algorithm 1: How to get the simulated tactile image

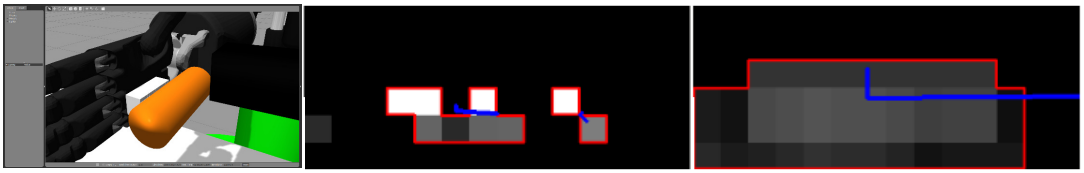
initialization: Create triangularized meshes of sensing cells; set the parameters of the sensor: stiffness and damping, Coulomb friction force coefficients in the two directions on the planar surface; set the maximum allowed penetration; define the attachment frame in the geometrical description of the robot
input : Contact forces F provided by the simulator
output : Simulated Tactile Contact Image

1 **foreach** *Tactel* (i, j) *of the* $N_x \times N_y$ *sensing array* **do** apply 2.27 to get a scaled force profile $F_{sim}(i, j)$;

2 **foreach** $F_{sim}(i, j)$ **do** apply 2.26 to get a contact image pixel $I_{sim}(i, j)$;

As indicated in Algorithm 1, the simulator computes all the normal contact forces that occur at the points of collision between the surface meshes of the sensor and the object. However, a predefined number of contact points could not be sufficient to describe a complete contact surface in a continuous way due to the "three chair leg" problem. When the four legs of a chair are lying on a planar floor with high stiffness, only three legs are really in contact at equilibrium. Similarly, if the sensor is considered as a rigid object (i.e., with high stiffness), the simulation will obtain independent contact regions (Fig. 2.15b), distorting the continuity of the real contact surface between the sensor and

the touching object. Therefore, the compliance of the sensor should also be added to the simulation. As shown in the second step of Algorithm 1 and explained in eq. (2.26), this compliance is modeled through a convolution of the contact forces at each tactel with 2D PSF. Fig. 2.15c shows how the problem of the "three chair leg" is solved with PSF function and a continuous and complete contact region is finally obtained.



(a) (b) (c)

Figure 2.15: (a) A cylindrical object in contact with the simulated sensor. (b) The tactile image when the 2D Gaussian PSF is not applied yet. (c) The tactile image when the function is applied.

2.4.2 Results

The next step is to verify the methods presented so far by evaluating them using simulated model and real sensor. The following results will present tactile images and the features extracted from these images. The tactile images were acquired when the sensing surface was in contact with a spherical object (Fig. 2.16a) and a cylindrical object (Fig. 2.16b). The objects were not floating and not subject to gravity. They were stiff and rigid. The tactile sensing array was implemented on the palm of the simulated robot hand. The axes of the sensing frame were oriented so that x-axis points towards the finger tips and z-axis points outwards the palm; the origin of the sensing frame is located at the corner (closest to the wrist joint) of the sensing surface of the sensor.

Point-contact

1) Simulated sensor and obtained features. As was shown in Section 2.3.1, the contact frame for the point-contact type is represented by 3 features (eq. (2.2), (2.3), (2.18)). A spherical object was placed at different locations. Fig. 2.17 illustrates the object and the sensor on the left-hand side and the tactile images for the corresponding locations of

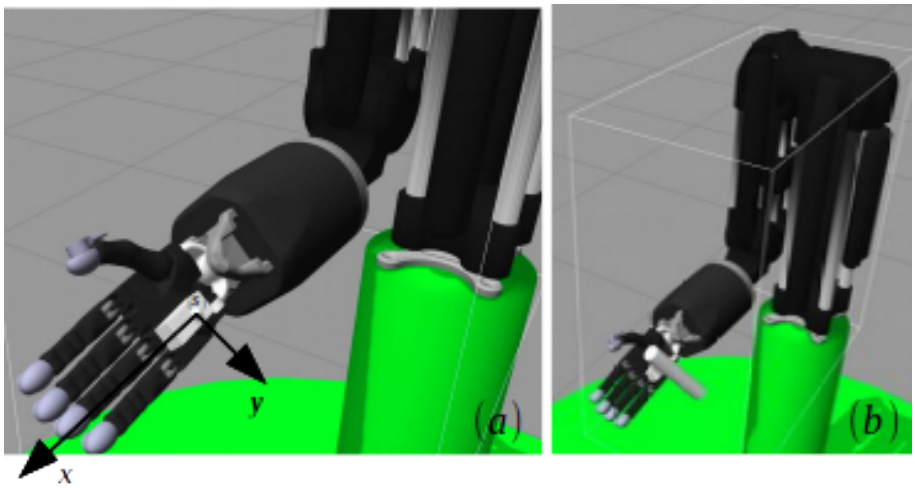


Figure 2.16: (a) Point- and (b) edge - contacts in Gazebo.

the sphere on the right-hand side. The red circle represents the locations of the object on the sensing surface depicted as a blue square. The spatial resolution is given by the dimension of a tactel (3.4 mm). The simulator provided a rather high repeatability for the CoC ($CoP = CoC$ for the point contact) feature, although the collision forces provided by ODE were not reliable. This is due to the small deviations of the joints of the robot to which the sensor is attached. Therefore, the simulation model works well for the features that do not depend on small deviations of interaction forces, e.g. the center of contact.

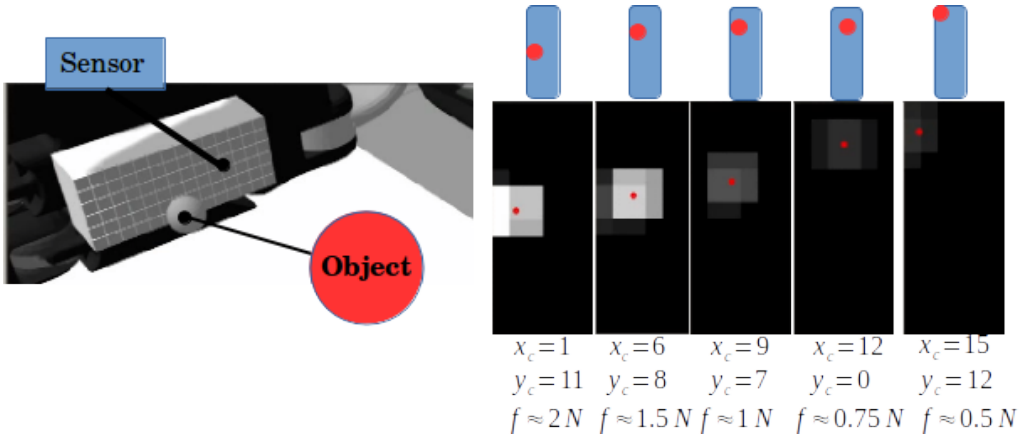
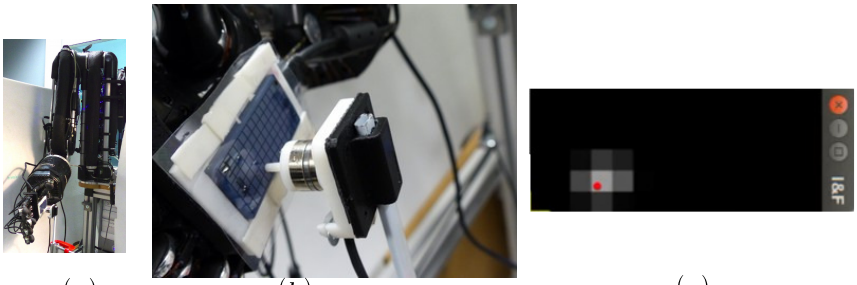


Figure 2.17: Spherical object in contact with the simulated sensor (left-hand side): tactile images and features (right-hand side). The tactile images are derived using Algorithm 1. Brightness in each pixel is proportional to the normal force in the corresponding tactel. The equation eq. 2.2 computes the center of contact (red dot). The dimensions of the centers of contact are in mm.

2) Real sensor and obtained features. The tactile feature extraction was also evaluated for point-contact using the real tactile array attached to the Shadow Hand (Fig. 2.18a). The sensor was in contact with an intender – an object that is used to push objects (Fig. 2.18b). As shown in Fig. 2.18c the extracted CoC (in red) allowed to locate the contact frame.



(a) (b) (c)

Figure 2.18: Computed *CoC* to find the location of a contact with an intender. (a) A sensing array on a robot hand. (b) Contact with the intender (environment). (c) The location of the contact within the corresponding tactile image.

Edge-contact

1) Simulated sensor and edge with uniform line force. As was shown in Section 2.3.2, the contact frame for the edge contact type is represented by 6 features according to eq. (2.24). Therefore, the moments of force about the x-(y-) axis and the orientation angle of an object edge around the z axis of the sensing frame were added. A cylindrical object was placed at different orientations. Fig. 2.19 illustrates the object and the sensor on the left-hand side and the tactile images for the corresponding orientations of the cylinder on the right-hand side. The red ellipse represents the configurations at which the object was placed on the sensing surface. The surface is depicted as a blue square. The spatial resolution for the extraction of the orientation angle depends on the smallest measurable deviations of the points of contact along x and y axes. Let us define δx and δy to be the smallest detectable change along x axis and y axis, respectively. Then the smallest

detectable change of the orientation angle $\delta\alpha_z$ is the following:

$$\delta\alpha_z = \arctan \frac{\delta y}{\delta x} \tag{2.28}$$

For the tactels with length and width equal to 3.4 mm, the precision of the orientation angle computation is 0.7 rad.

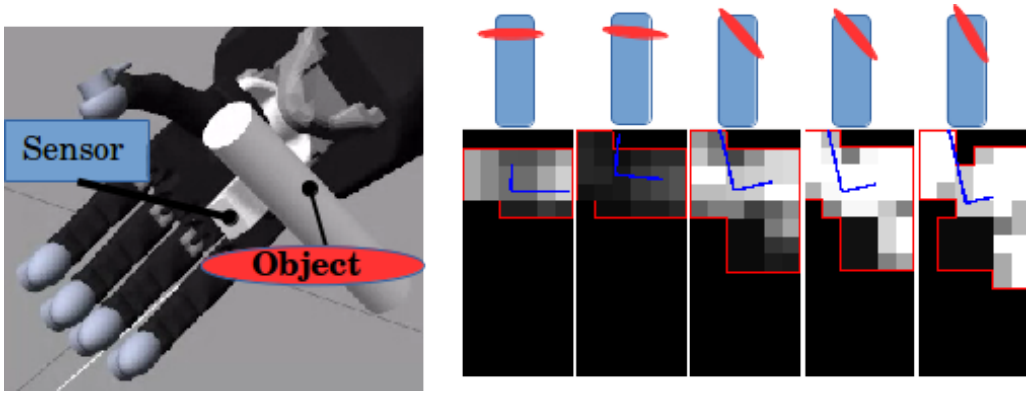


Figure 2.19: Cylindrical object in contact with the simulated sensor (left-hand side): tactile images and features (right-hand side). The tactile images derived using Algorithm 1. Brightness in each pixel is proportional to the normal force in the corresponding tactel. The equation eq. 2.10 computes the orientation of the edge in radians.

We assumed that line force was uniformly distributed as shown in Fig. 2.9a. For the evaluation of the ZMP feature, the tactile image was artificially constructed as explained in the following.

2) Manually created image of an edge with non-uniform line force. In order to compute the ZMP_x feature (estimation of the moment of force around the y axis), let us assume that the following tactile images are given (the tactile images were created manually and they were not derived in the Gazebo simulator): the full contact (Fig. 2.20a) and the partial contact (Fig. 2.20b). We compute the CoP feature by applying eq. (2.2) on the gray-scale image and the CoC feature by applying the same equation on the thresholded image. The red dots show the CoP feature and red squares the CoC feature.

3) Real sensor and derived features. The sensor on the hand is in contact with a 3D printed plastic stiff bar (Fig. 2.21a). This plastic bar is not infinite in the sensing surface. This plastic bar generates a partial contact with the sensor surface (Fig. 2.21b). The orien-

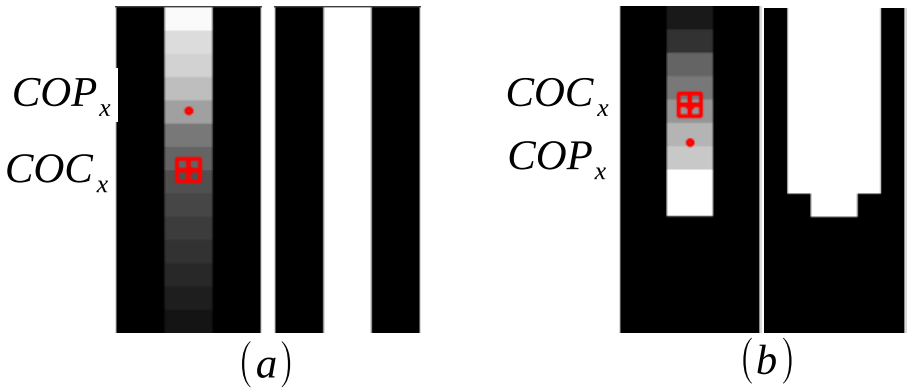


Figure 2.20: (a) ZMP feature when an edge is infinite in the tactile image and (b) partial. The center of pressure COP_x is marked as a red dot and the center of contact COC_x as a red square. The black and white images correspond to the thresholded tactile images.

tation angle of the edge caused by the bar is depicted with the red line. *CoP* is depicted as a red dot and *CoC* as a red square (Fig. 2.21b). Fig. 2.21c shows the thresholded tactile image. In both images, the highest pressure corresponds to the highest brightness. Fig. 2.21d shows the orientation angle of the edge caused by the bar in the Cartesian space. The difference between *CoP* and *CoC* shows that the support of the bar is generating a moment around the y axis (ZMP_x).

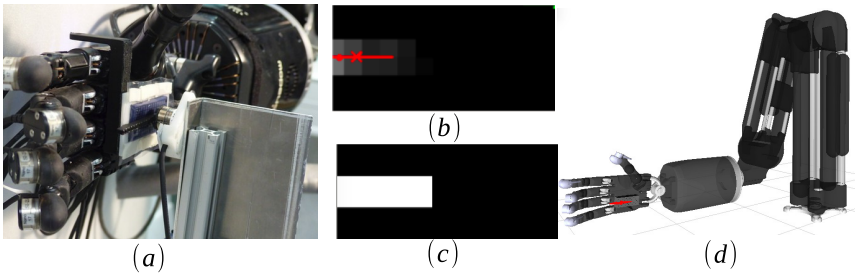


Figure 2.21: Evaluation of the edge extraction. (a) The contact with an edge of an object. (b) The corresponding tactile image and (c) the binarized form of it. (d) The orientation of the edge in the Cartesian space.

2.5 Conclusions

This chapter has presented the relation between the Cartesian components of contact frames and the corresponding proposed tactile features for the point contact (3D vector,

Fig. 2.22) and the edge-contact (6D vector, Fig. 2.23).

In the case of edge-contacts, existing tactile feature vectors of 4 components have been extended with two new features (ZMP_x and ZMP_y) that estimate moments of force around the axes lying on the sensing surface through rotational compliance constants. The inclusion of these compliances in the sensing array improves the continuity of the contact representation of non-uniform line loads.

All of these tactile features have been tested in simulation (with a new tactile sensor model that includes this compliance through a convolution of contact forces over tactels with 2D PSF) and with a real sensor. These tests do not only confirm that the proposed tactile features allow to reconstruct the contact frame between an object (spherical or cylindrical) and an compliant sensor, but also establish a clear unique relationship between the motions of the sensor and the object in contact with it. Thereby, if this sensor is installed over a robot, these tactile features could be used to control all the dimensions of the contact state of the robot with the object: 3D for the point-contact and 6D for the edge-contact. Next chapter will explain in detail a new tactile servoing framework developed in this thesis in order to use these tactile features for touch-driven robot control.

$$\begin{bmatrix} x_c & y_c & d_z & w_x & w_y & w_z \\ \downarrow & \downarrow & \downarrow & & & \\ [CoC_x & CoC_y & f & - & - & - \ ] \end{bmatrix}$$

Figure 2.22: Tactile features and contact frame for the point-contact.

$$\begin{bmatrix} x_c & y_c & d_z & dw_x & dw_y & w_z \\ & & & \downarrow & \downarrow & \\ [CoP_x & CoP_y & f & CoP_y - CoC_y & CoP_x - CoC_x & \varphi_z] \end{bmatrix}$$

Figure 2.23: Tactile features and contact frame for the edge-contact.

Chapter 3
Tactile servoing controller

The tactile features explained in the previous chapter are now considered within robot controllers in touch-driven robotic exploration and manipulation tasks. Firstly, the problem of how to perform physical interaction tasks between robots and their environment is formalized in Section 3.2. Section 3.3 outlines the existing interaction control approaches and reviews the state-of-the-art of previous works that use tactile feedback to servo a contact frame, analyzing their drawbacks and limitations (Sec. 3.3.2). In order to mitigate them, a novel tactile controller composed by a tactile Jacobian and a task projection matrix is developed in Section 3.4. These matrices map errors of tactile features to 3D Cartesian pose errors of the robot end-effector carrying a tactile sensor according to a contact task. The last Section 3.5 presents the responses of this tactile controller for two different robot arms: a 6 dof Shadow robot arm/hand and a 7 dof KUKA Light Weight Robot arm.

3.1 Introduction

Physical interaction tasks, including object exploration, dexterous manipulation, and recognition, can be performed even when a manipulated object is occluded from vision sensors [Johansson 09]. Indeed, the dexterity of manipulation is not affected by the blindness of a person. But, in the absence of tactile sensor, contact forces generated during the interaction with the environment can be estimated only. Then, the performance of a manipulation task depends on hard-coded internal models of the environment. However, the real world is highly variant; it can not be assumed, therefore, that the environment is perfectly known. Whereas some tasks can be performed even when vision is occluded, tactile feedback can not be removed if successful manipulation is desired. Hence, tactile feedback is a necessary requirement for the execution of physical interaction tasks. In addition, it is also interesting to study how this information can be used for enabling

reliable physical interaction when other sensors are not available.

In general, tactile feedback allows detecting events, such as the first contact and contact type evolution (point or edge). Thus, one of the ways of analyzing tactile feedback is by using event-based approaches. Nevertheless, tactile feedback can also be used continously in a control loop for keeping a contact frame in a desired state. Thereby, the execution of a physical interaction task can be divided into the following steps:

- A non-contact phase, corresponding to the approaching movements of the robot towards the object to be touched.
- An interaction phase, where the robot is in contact with the environment (object). In this phase, a closed-loop servo system of the robot controls the contact to perform an interaction task.
- A transient phase to a new interaction phase, where the robot detects an event that triggers another task to perform. We will see how tactile signals will be used to trigger between different manipulation phases in Section 3.4.2.

During the second (interaction) phase, when the robot is in contact with the environment, it is important to design a controller that can deal with unpredicted interaction states and adapt the robot motion accordingly. The contact feedback would then guide the robot to move in a desired manner on the object surface based on continuous data coming from the sensor, which must be processed on real-time. Therefore, a robot can be driven via ***tactile servoing*** [Sikka 94a]. The term *tactile servoing* stands for a touch driven control[1] using spatial contact information, usually provided by pressure sensing arrays [Cutkosky 14].

This chapter proposes a new tactile servoing framework (Fig. 3.1) that uses tactile features discussed in the previous chapter, mainly the ZMP feature for representing mechanical the compliance of sensor surfaces. In the baseline approaches, elastic covers of sensing arrays were used for keeping a continuous tactile feedback signal [Berger 91, Sikka 94a, Chen 95]. In contrast, we aim to use the ZMP to control the contact frame with more dofs. Thereby, the proposed tactile features altogether represent a contact frame as

[1] a tactile sensing system is used to close a position-control loop of a robot

discussed in the previous chapter. The current (s) and desired (s^{ref}) tactile features will be used as inputs to the new tactile controller. This tactile controller will make s converge to s^{ref} by applying two matrices: a task projection matrix (selecting and projecting the Cartesian components of a task) and a tactile Jacobian (mapping tactile features to robot pose). The desired pose in the Cartesian space (x^{des}) obtained as output from the tactile controller will be tracked by the robot with a standard position controller involving its kinematic model. This tactile controller is general enough to be applied to various tasks: manipulation by rolling, exploration, object recognition and co-manipulation, as will be explained in Chapter 4.

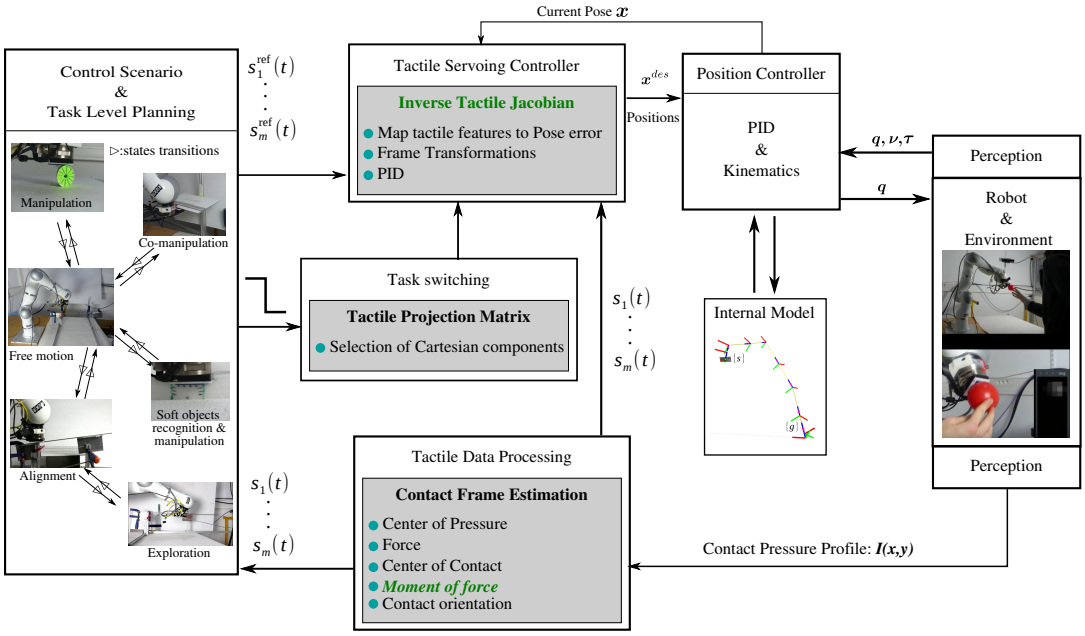


Figure 3.1: Overview of the methodology developed for achieving control of contact frame and transitions between tasks. s_i and s_i^{ref} are the current and desired tactile features; x and x^{des} are the current and desired poses; q, v, τ are the joint angles, velocities, and torques.

3.2 Problem statement: hypothesis and assumptions

Formalization of the problem: Fig. 3.2a and b, which illustrate two categories of contact cases that commonly occur. Fig. 3.2a represents the configuration in which the robot

makes direct contact with a rigid fixed environment in order to explore it. In Fig. 3.2b, the robot makes contact with the environment through an object. The robot can perform both tasks by physically interacting with the environment/object through the end-effector. For this reason, the controller should be based on the computation of the desired pose of the robot's end-effector. Hence, we can formulate the problems as:

1) given a constrained motion in the 3D Cartesian space of a robot end-effector equipped with an elastic tactile array, the contact frame must be defined and tracked in the Cartesian space.

2) given different robot contact tasks (exploration or manipulation), the robot should be able to switch between them in a continuous way.

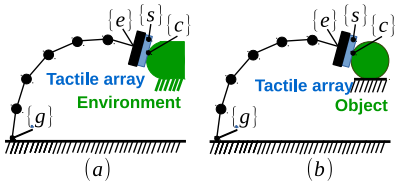


Figure 3.2: Types of physical interactions. A robot with a pressure sensing array interact with: (a) a stiff environment and (b) an object on the stiff environment. The frames $\{g\}, \{e\}, \{s\}, \{c\}$ correspond to the global, end-effector, sensor, and contact frames, respectively.

Apparatus:

1) A flat sensing array with an elastic surface (Section 2.3).

2) A serial link manipulator (robot arm). The sensing array is fixed on the end-effector (hand) of the robot arm (Fig. 3.2a). The frames of the arm with the sensor are defined as followed:

- $\{g\}$ is the global frame, i.e., base of a robot
- $\{e\}$ is the end-effector (palm) frame that is given by geometric transformations from the base frame to the end-effector frame ${}_{\{g\}}\mathbf{T}^{\{e\}}$, which includes translational and rotational transformations
- $\{s\}$ is the sensor frame that is statically defined in the end-effector frame

Hypothesis

1) The robot with the tactile array is able to perform compliant motion by satisfying kinematic constraints (Section 3.4.4) – including artificial and natural ones – and taking into account the compliance of the elastic surface of the array.

2) A "*tactile controller*" is designed as an outer control loop of the robot's motion controller, which can be implemented as a simple PD (Proportional and Derivative) and gravity compensation algorithm as given in eq. (3.13).

3) Artificial constraints are directly related to the tasks and the contact types that appear during the tasks.

Assumptions

- Contact appears on the sensing surface only and is continuous.
- Friction forces are small enough to be not considered during sliding motions.
- Friction forces are within the Coulomb friction cone during manipulation tasks.
- The environment is rigid.
- Since velocities are rather small in most of exploration and manipulation tasks, the quasi-static case is considered.
- The sensitivity of the sensor is high enough to detect any contact.
- Tactile images during interactions belong to one object only.
- Desired poses of the end-effector are always within the dexterous workspace, i.e. the Cartesian subspace which can be reached with arbitrary orientation.
- The changes of the contact state with the environment are relatively slow compared to the responses of the robot.

3.3 Related work

So far, we introduced and formulated the problem of controlling robot physical interactions with information coming from pressure sensing arrays. As we use a robot arm, the robot motion control approaches should be briefly reviewed first. Then the state of the art in robot control using pressure sensing arrays will be shown and discussed. As was already mentioned above, this control concept is referred to as tactile servoing, which was introduced in [Sikka 94b].

3.3.1 Motion control for robot interaction

The control approaches of robots for physical interaction with the environment can be divided into the following groups: 1) approaches that perform an indirect force control

(passive), 2) those that control the contact force explicitly (active), and 3) hybrid position/force approaches.

The first group includes passive compliance (stiffness) control approaches. Passive stiffness control is performed by attaching to the robot end-effector a Remote Center Compliance (RCC)[2] device, which can be deformed according to the external forces. The main advantage of this approach is its simplicity. However, there is no explicit control of the robot stiffness. But the stiffness parameters can only be modified by changing the devicewhich decreases the reconfigurability of the system.

The second group includes active stiffness (or **impedance**) **control** with position or torque controlled robots. In contrast to the passive approach, the active stiffness approach allows to control the desired stiffness at the end-effector according to the task requirements. The active stiffness and impedance approaches require usually force sensors and control the interaction force through position control laws [Hogan 84] for position-driven robots or through torque control laws for torque-driven robots. For the torque-driven robots, the dynamic model of the system must be precise and the actuating torque must be accurate.

The third group includes **hybrid position-force** control approaches that combine position control on some directions and explicit force control on the rest. The control signals from the control modes are combined before sending them to the robot actuators. The position mode ensures a suitable trajectory of the robot in free space, whereas the force mode controls the motion in directions where contact with the environment is expected [Raibert 81]. There is one more approach referred as external position-force control [Perdereau 96] that incorporates the advantages of the active stiffness control and the hybrid position-force control. It allows simultaneous explicit force control and position control along the same direction. It consists of an inner position control loop enclosed inside an outer force controller. Therefore, the output of the force controller modifies reference values of the inner position controller.

Thus far, we have outlined the main methods to control physical interactions. The force feedback can be replaced by tactile feedback for detecting contact with an object

[2] A mechanically compliant device in industrial manipulators.

and also for adapting the end-effector motion accordingly.

3.3.2 Tactile control

The control approaches described above provide general schemes for controlling physical interactions using position and force feedbacks. However, in order to increase the usability and functionality of robots, researchers since 1980s [Montana 88] have been trying to incorporate the tactile feedback (given by pressure sensing arrays), because it can provide an image of the contact with the environment. From these contact images, different tactile features can be extracted and used to control the robot's end-effector dofs as shown in Table 3.1.

Approach	Controlled dofs for the edge	Controlled dofs for the point	Transition between contact types.
[Berger 91]	2 dofs (force, orientation)	×	×
[Sikka 94a]	3 dofs (force, orientation, COC_x or COC_y)	×	×
[Chen 95] and [Zhang 00]	3 dofs (force, orientation, COC_x or COC_y)	3 dofs (force, COC_x, COC_y)	×
[Li 13a]	4 dofs (force, orientation, COC_x, COC_y)	3 dofs (force, COC_x, COC_y)	Selection matrix for different simple tasks (edge exploration, manipulation by rolling) without possibility to change the type of contact.
New approach	**6 dofs** (force, orientation, COC_x, COC_y, ZMP_x or ZMP_y)	3 dofs (force, COC_x, COC_y)	Task projection matrix for complex tasks (where the contact type can change): edge exploration, manipulation by rolling, co-manipulation.

Table 3.1: Summary of the state-of-the-art in the tactile servoing.

The thrust that has pulled control community towards tactile control is the fact that the local information about the interaction properties at the points of contact is more rich and robust compared to visual information. One of the first trials to incorporate the tactile feedback into robot controllers was conducted by [Allen 88]. The author used a

tactile sensing device that was developed at L.A.A.S.[3] in Toulouse, France. The sensor was used to detect contact and aid a visual system to distinguish between circular holes or bumps. But this tactile information was not integrated in a low level controller and, therefore, the motion of the robot was not driven by tactile feedback.

One of the first approaches of robot motion driven by tactile feedback was implemented by Berger et al. [Berger 91] in the edge tracking problem. The authors used a 16 by 16 sensing array with compliant surface to ensure a continuous feedback signal of the contact pattern. The sensor array provided a tactile image that was considered as a gray-scale vision image. The elastic cover of the sensor created undesirable cross-talk in the tactile image that was removed with a thresholding operation. An edge was detected by region-based segmentation and its orientation was obtained using the Hough Transform. This edge was tracked by the hybrid force and position controller depicted in Fig. 3.3. The orientation of the edge was tracked along one axis of the end-effector frame and the contact force with the edge (estimated based on the area of contact) was maintained in the normal direction to the edge.

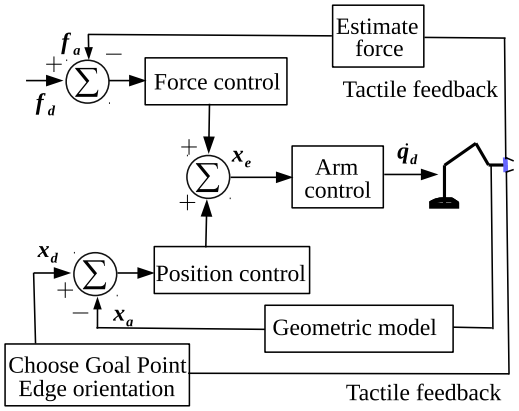


Figure 3.3: General approach of the hybrid parallel position-force control for tactile servoing [Berger 91]. The desired and feedback forces f_d and f_a, and positions x_d and x_a, respectively, are separated in such a way that they are orthogonal to each other. The outputs of the position and force control algorithms construct the end-effector's pose error x_e that is then transformed to the desired joint angle velocities $\dot{q}_d$.

In 1991, Sikka et al. [Sikka 94a] developed a tactile based control framework that allowed a robot to not only slide over an object's edge but also manipulate objects by rolling them. In contrast to the previous approach, the authors applied elasticity theory in order

[3]Laboratoire d'Analyse et d'Architecture des Systemes

to derive zeroth-, first-, and second- order moments of the contact pressure distribution; and the center of the edge was computed from these moments. Thus, there were 3 dofs that could be controlled based on the error in tactile feature values using the same hybrid position-force control as in the latter approach. The end-effector could move not only along the axis of the edge but also perpendicular to it so that the center of contact converges to a desired location. During this move, an object lying on a flat surface and pressed from the top by the end-effector rolled over the surface.

Drawing on the ideas from the latter approach, Chen and Zhang [Chen 95, Zhang 00] extended the concept by including edge- and point- contact types. The moments extracted in this approach were mapped to a contact state via the so-called inverse Tactile Jacobian. The same three features (normal force, center of contact, and edge orientation) were proposed to perform an edge following task with the assumption that the edge is infinite and any edge servoing task begins at one of the corners of the tactile array. The authors repeated the edge following and object rolling tasks, and showed that by servoing the point-contact they could reconstruct a spherical object. For the point-contact type, they derived three new features: the center of contact (in x and y axes) and force along z axis. The feature extraction was based on the sensors' morphology analysis (i.e., strain and stress analysis) with a number of assumptions. The main limitation of this approach was the fact that the controller could not switch between the point- and edge- contact types during the execution of the tasks.

Li et al. [Li 13a] came up with an unified tactile servoing framework that allowed switchable control of different servoing modes for the edge- and point- contact types. In contrast to the latter approach, the authors applied image processing techniques: image moments and the Principal Component Analysis (PCA) to extract the orientation of the edge with respect to the sensing frame. The switch between different tasks depending on the contact type was realized via a selection matrix. So, for the point-contact type, the orientation about the z-axis was disabled by this selection matrix. Nevertheless, when a contact type is fixed by the selection matrix during the execution of a task, this approach is not able to change to another contact type dynamically and in a continuous way. This could be required in more complex manipulation tasks such as:

- 1 - when the edge is straight but not parallel to the sensing surface. Then when a robot is in contact with the edge, there is only part of the edge in a tactile image. The robot senses a point-contact type. Therefore, the robot's end-effector should first align with the edge in order to increase the contact area. For this case, the robot should be able to detect the edge when it appears in the tactile image and consequently switch to the next task, e. g., the edge following task. In the existing tactile servoing approaches, this was not performed yet.

- 2 - when the edge is not infinite in a tactile image. In the previous case, when the end-effector was aligning with the edge, the edge was short, so that only a "*partial*" edge occurs in the tactile image. By "*partial*" edge, we refer to the edge that it is not covering all the sensing surface, i.e., when a line in the tactile image is not infinite. Then, during the alignment with the edge, the baseline approach would fail.

- 3 - when the edge is bended in the 3D Cartesian space and not in a planar plane. In the baseline approaches, this case was not considered because the moment of force applied on the edge could not be estimated.

- 4 - when the edge is not rigid. For example, a horizontal metal bar that is fixed from one side and floating from the other will bend more when an end-effector is closer to the floating end of the bar. Therefore, the bar will bend in the planar surface that is perpendicular to the sensing surface. And, as in the second case, the baseline approaches would fail.

In summary, the main drawback of the baseline approaches is that they are limited to specific contact configurations between objects and tactile sensors. The functionality and dexterity of manipulators are thereby limited. The main reason is that the number of dofs that can be controlled based on the tactile feedback in the baseline approaches is limited to 4 for the case of the edge contact type.

In order to overcome the disadvantages of the baseline approaches, a new **touch-dependent inverse Tactile Jacobian** and **task projection matrix** are proposed. The tactile Jacobian incorporates the new tactile features (ZMP) and allows a robot to perform complex manipulation tasks when contact types change during execution.

3.4 Tactile servoing controller

In the following the proposed tactile controller is explained in detail. And we will begin with a functional scheme of the controller.

3.4.1 Functional scheme

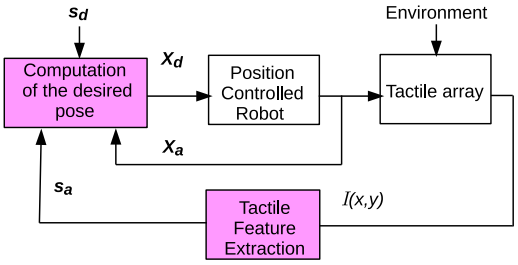
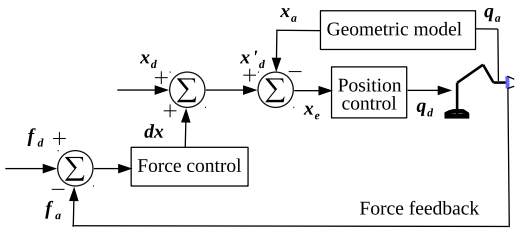


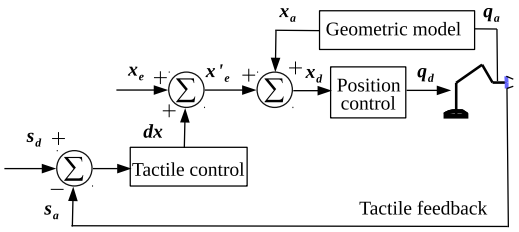
Figure 3.4: General control architecture for touch driven manipulators via tactile servoing with a position controlled robot. The desired pose $\boldsymbol{x_d}$ is computed from actual tactile features $\boldsymbol{s_a}$ that are extracted from two-dimensional pressure distribution $I(x, y)$ and current actual pose $\boldsymbol{x_a}$.

Fig. 3.4 represents a general control architecture of a tactile servoing scheme with a position controlled robot. The task in the tactile servoing is to control the pose $\boldsymbol{x}$ of a robot's end-effector using tactile information, *features*, extracted from a contact profile. This pose is represented by a 6 element vector encoding position and orientation in the three-dimensional Cartesian space. In general, tactile servoing approaches follows the same control scheme: the task is controlled by a chosen feature set $\boldsymbol{s}$, which is derived from the tactile image $I(x, y)$ acquired by a tactile sensing array. In the figure, the terms $\mathbf{s}_d$ and $\mathbf{s}_a$ are the desired and actual features, $\mathbf{x}_d$ and $\mathbf{x}_a$ are the desired and current (actual) poses; $\mathbf{x}_a$ is given by the robot's Forward Kinematics and $\mathbf{x}_d$ is computed based on $\mathbf{s}_d$ and $\mathbf{s}_a$. The desired $\boldsymbol{q_d}$ and actual $\boldsymbol{q_a}$ joint angles can be computed by the inverse Kinematics and joint angle position encoders, respectively.

In fact, this functional scheme for tactile servoing represents an evolution from classical external hybrid force-position control (Fig. 3.5a), where the forces $\boldsymbol{f}$ are replaced by tactile features $\mathbf{s}$. This is valid because the information extracted from tactile images (features) is used to reconstruct the current 3D pose (position/orientation) of a contact frame, which is defined in the 3D Cartesian space.



(a) General approach of the hybrid external position-force control



(b) General approach of the hybrid external position-**tactile** control

Figure 3.5: External hybrid force-position control scheme evolves to tactile servoing.

Thus far, we introduced the indirect tactile servoing scheme that receives as inputs tactile features extracted from tactile images. Next, we will describe how they are related to a contact frame.

3.4.2 From the tactile image space to the Cartesian space

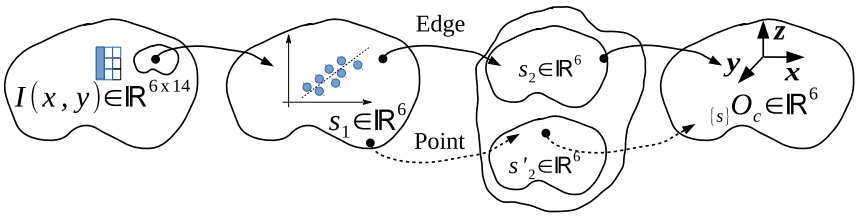


Figure 3.6: Mapping an Image $I(x,y)$ to a contact frame ${}_{\{s\}}\mathbf{O}_c$ in Cartesian space. The first feature vector $\boldsymbol{s_1}$ consists of features that are common for point- and edge-contact types. The second feature vector $\boldsymbol{s_2}$ or $\boldsymbol{s'_2}$ is the set of features in the case of the point-contact or edge-contact type, respectively. It provides the contact frame in the Cartesian space.

A tactile array gives a two-dimensional grey-scale image of a contact state. Using the image processing techniques described in the previous chapter, the image is reduced from $\Re^{(N_x \times N_y)}$ (where N_x and N_y are the numbers of tactels in x and y axes, respectively) to a feature vector $\boldsymbol{s_1} \in \Re^6$, including the following features:

$$\boldsymbol{s_1} = [CoP_x, CoP_y, f, CoC_x, CoC_y, \alpha_z]^T \tag{3.1}$$

Then the contact image can be classified to one of the two contact types: the edge- or

point-contact. $\boldsymbol{s_1}$ is, therefore, mapped to another feature vector $\boldsymbol{s_2}$ corresponding to the case of contact with a point or feature vector $\boldsymbol{s_2'}$ corresponding to the case of contact with an edge. The first feature vector has the dimension of $\Re^3$:

$$\boldsymbol{s_2} = [CoP_x, CoP_y, f, CoC_y, CoC_x, -]^T = [CoC_x, CoC_y, f]^T \tag{3.2}$$

We keep CoC because it is identical to CoP for the point-contact type. The second feature vector (for the edge) has the dimension of $\Re^6$:

$$\boldsymbol{s_2'} = [CoP_x, CoP_y, f, ZMP_y, ZMP_x, \alpha_z]^T \tag{3.3}$$

In fact, $\boldsymbol{s_2}$ (or $\boldsymbol{s_2'}$) is directly related to the deformations of the sensing surface in Cartesian space, which includes three position variables for the point-contact type:

$${}_{\{s\}}\boldsymbol{O_c} = [x_c, y_c, C_z \cdot f]^T \tag{3.4}$$

or three position and three orientation variables for the edge-contact type:

$${}_{\{s\}}\boldsymbol{O_c} = [x_c, y_c, C_z \cdot f, C_{\varphi_y} \cdot \mu_y, C_{\varphi_x} \cdot \mu_x, w_z]^T \tag{3.5}$$

The coordinates of tactels are represented in the 3D world with meters and radians: x_c, y_c, $C_z \cdot f$ are in meters and $C_{\varphi_y} \cdot \mu_y, C_{\varphi_x} \cdot \mu_x, w_z$ are in radians. These features are the kinematic constraints that should be respected by the relative movements between the object and robot. They can be switched on/off according to the task as will be explained in the following section (Section 3.4.2). However, we should understand first what is the condition that help distinguishingbetween the point- and edge-contact types.

Edge- and point-contact type identification

In eq. (2.10) the PCA was applied to a tactile image to detect two principal components U_1 and U_2. The resulting principle components can be used to differentiate the two types

of contact. If the inequality:

$$\lambda_1 > k \cdot \lambda_2 \tag{3.6}$$

is satisfied (Fig. 3.7 right-hand side), it means that there is one component at least k-times bigger than the other one. Thereby, the contact pattern can be considered as an edge and the corresponding eigen vector will determine its direction. Otherwise, there is a point-contact type in the tactile image (Fig. 3.7 right-hand side). The variable k is to be experimentally obtained. λ_1 and λ_2 are the eigenvalues corresponding to U_1 and U_2, respectively.

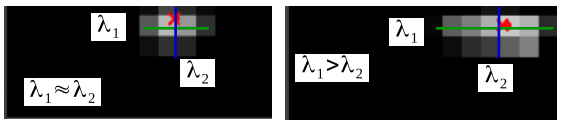


Figure 3.7: A point-contact type contact on the left and an edge-contact type on the right. The ratio between the lengths of resulting principal components of PCA applied to the tactile images are different for these two types of contact.

Feature error

Thus far, we introduced the tactile features from the control point of view: our features reduced the dimension from the tactile image space ($\Re^{N_x \times N_y}$, which is usually higher than 6) to Cartesian space ($\Re^6$). The next step is to derive the tactile feature error and develop the inverse tactile Jacobian matrix and task projection matrix (Fig. 3.8). The

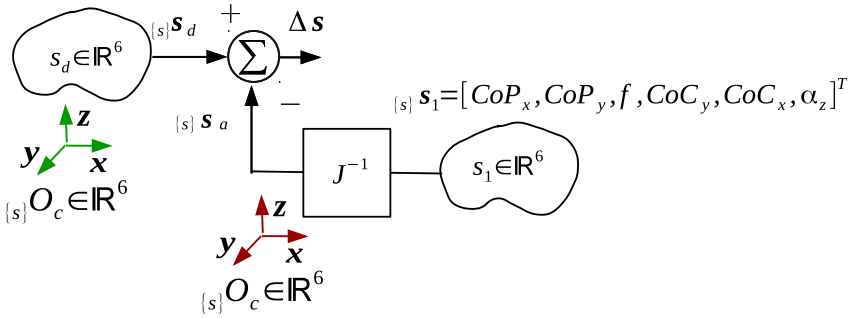


Figure 3.8: Feature error

actual state of the contact frame $\{c\}$ (given by eq. (3.4) or eq. (3.5)) in the sensor frame ${}_{\{s\}}\mathbf{s}_a$ is represented by either the feature vector $\boldsymbol{s_2}$ (for the point-contact) or $\boldsymbol{s_2'}$ (for the

edge-contact). The desired state of contact $\boldsymbol{s_d}$ depends on the task to perform. Then the feature error $\Delta \boldsymbol{s}$ is, in fact, the error of the contact frame with respect to the sensing frame Cartesian space, which also depend on the type of the contact (i.e., edge- or point-contact type). $\Delta \boldsymbol{s}$ is given by:

$$\Delta \boldsymbol{s} = \boldsymbol{s}_d - [\mathbf{J^{-1}}]\boldsymbol{s_1} \tag{3.7}$$

where $\mathbf{J^{-1}}$ can be thought as a tactile feature transformation matrix that maps one set of features to another set that can include either the ZMP feature for an edge-contact type or CoP feature for a point-contact type. This touch dependent inverse Tactile Jacobian is a matrix with the following form and elements:

$$\mathbf{J^{-1}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -\alpha_2 & 0 \\ 1 & 0 & 0 & -\alpha_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

The variables α_1 and $\alpha_2 \in [0, 1]$ as follows: α_1 or $\alpha_2 = 1$ for the edge-contact type (for controlling the ZMP_x or ZMP_y features) and α_1 or $\alpha_2 = 0$ for the point-contact type (this will allow rotation around a point-contact.

Considering Hypothesis 3 in the problem statement (Section 3.2), a task projection matrix $\boldsymbol{P} \in \Re^{6\times 6}$ can be introduced (Fig. 3.9). The matrix $\mathbf{P}$ selects and maps feature errors so that the initial tactile feature error $\Delta\mathbf{s}$ is mapped to a modified feature error $\Delta\mathbf{s}'$.

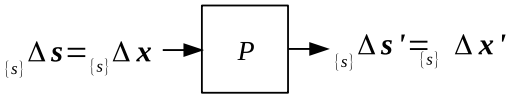


Figure 3.9: The projection matrix that selectively maps an error vector of tactile features $\Delta\mathbf{s}$ to the feature error $\Delta\mathbf{s}'$ with selected components. $\Delta\boldsymbol{x}'$ is then the pose error.

As the tactile features represent a contact frame:

$$\boldsymbol{O}_c = \boldsymbol{s} \Rightarrow \Delta \boldsymbol{O}_c = \Delta \boldsymbol{s} \tag{3.9}$$

where $\Delta \boldsymbol{O}_c$ is an error between the desired and actual contact frames. The error in the pose of the end-effector $\Delta \boldsymbol{x}$ is directly given by the error in the contact frame:

$$\Delta \boldsymbol{x} \sim \Delta \boldsymbol{O}_c = \Delta \boldsymbol{s} \tag{3.10}$$

Hence, after multiplication with the projection matrix, the pose error in the Cartesian space $\Delta \mathbf{x}'$ is given by:

$$\Delta \boldsymbol{x}' = \mathbf{P} \cdot \Delta \boldsymbol{x} = \mathbf{P} \cdot \Delta \boldsymbol{s} \tag{3.11}$$

The task projection matrix selects how a robot's pose is adjusted based on the feature error:

$$\boldsymbol{P} = \begin{bmatrix} \gamma_x & 0 & 0 & 0 & \gamma_{x \to wy} & 0 \\ 0 & \gamma_y & 0 & 0 & 0 & 0 \\ 0 & 0 & \gamma_z & 0 & 0 & 0 \\ 0 & 0 & 0 & \gamma_{wx} & 0 & 0 \\ 0 & 0 & 0 & 0 & \gamma_{wy} & 0 \\ 0 & 0 & 0 & 0 & 0 & \gamma_{wz} \end{bmatrix} \tag{3.12}$$

where $\gamma_x, \gamma_y, \gamma_z, \gamma_{wx}, \gamma_{wy}, \gamma_{wz} \in [1, 0]$ is a binary value to select a set of features depending on the task. There is one special element $\gamma_{x \to wy}$ that is also a binary value: when $\gamma_{x \to wy} = 1$ the rotational component around y axis of the tactile feature error is related to the translation error along x axis of the end-effector's pose. This will allow to perform co-manipulation tasks as will be shown in Section 4.6. Thus, the variables allow the use of the same controller in different tasks that have different kinematic constraints and even dynamic changes of the contact types.

Thus, the equations eq. (3.8) and (3.11) relate the error in the feature space with an increment of the end-effector pose in the Cartesian space.

3.4.3 From Cartesian space to the joint space

Thus far, we introduced the tactile feature error and the ways how this feature error can be mapped to the end-effector pose error. This error vector should be controlled with a robot controller, which will be described in the following.

Control inputs to the robot controller are defined by the set of features ($\Delta \boldsymbol{x}'$ given in eq. (3.11)) that represent the error in Cartesian space. In order to servo this Cartesian er-

ror, the end-effector should be driven by a law making this error converge to zero. Finally, this error should be mapped to the joint space of the robot's end-effector, which will be represented by the vector of robot joint variables: joint angles $\boldsymbol{q}$ for the position- and joint torques $\boldsymbol{\tau}$ for the torque- controlled robots. These variables constitute the joint space, which is also referred to as the configuration or generalized space. The computation of $\boldsymbol{\tau}$ (or $\boldsymbol{q}$) will constitute the motion controller of the robot explained in the next section. In addition, in most robotics hands, the actuation is realized via tendon driven mechanisms, for what one more mapping is needed for the transformation from the actuation space to the joint space (Fig. 3.10).

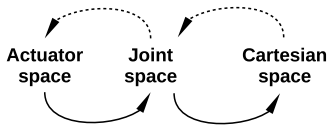


Figure 3.10: Mapping between the kinematic descriptions.

Robot motion controller

The robotics arms and hands that we are going to use will include rotary joints, also integrating joint angle and torque (or motor current) sensors. In the robotics literature, the "proportional-derivative (PD) plus gravity compensation" motion control law has been shown to be sufficient when these sensors are present in the robot's joints [Siciliano 08]:

$$\boldsymbol{\tau}_d = \boldsymbol{J}_\theta^T(\mathbf{K_P}(\boldsymbol{x}_d - \boldsymbol{x}_a) - \mathbf{K_D}\dot{\boldsymbol{x}_a}) + \boldsymbol{g}(\boldsymbol{q}) \tag{3.13}$$

where $\mathbf{K_P}$ and $\mathbf{K_D} \in \Re^{n\times n}$ are the positive-definite gain matrices, $\dot{\boldsymbol{x}_a}$ is the motion twist (velocity vector) of the end-effector, n is the number of joints, $\boldsymbol{g}(\boldsymbol{q})$ is the $(n \times 1)$-vector of gravity force, $\boldsymbol{q} \in \Re^n$-vector of joint variables, $\boldsymbol{J}_\theta^T$ is the transpose matrix of an $n \times 6$ robot Jacobian matrix, and $\boldsymbol{\tau}_d$ is the $(n \times 1)$ is vector of joint torque control inputs.

In the above equation, $\boldsymbol{x}_d$ is the desired Cartesian pose of the end-effector, which will be continuously tracked by the robot[4]. Assuming the quasi-static case, when the joint controllers can maintain their desired set points independently and dynamic effects can

[4]The ability to follow the pose is limited by the bandwidth of the robot in its control system.

be neglected, the equation eq. (3.13) can be considered as the Cartesian **inner** position controller of our robot.

External hybrid tactile-position controller

Thus far, we defined the tactile contact frame error vector $\Delta x'$ (eq. (3.11)) and presented the inner Cartesian pose controller for a serial-link manipulator in the previous section, but now we need to relate them. In order to include the tactile error vector into the robot's motion controller, we will modify the desired pose x_d from eq. (3.13) so that it is defined as follows:

$$x_d = x_a + \Delta x'' \tag{3.14}$$

where $x_d = x(t+1)$ is the desired pose, $x_a = x(t)$ is the current (actual) pose calculated by the Forward Kinematics $G_\theta(q)$ of the robot, and the increment $\Delta x''$ is the contact frame error in the Cartesian space obtained by applying proportional-integral (PI) regulators to $\Delta x'$. Thereby, $\Delta x''$ is the contact frame error in the Cartesian space given by:

$$\Delta x'' = {}_{\{g\}}\mathbf{T}^{\{s\}} \cdot \left(K_P \Delta x'(t) + K_I \int \Delta x'(t) dt\right) \tag{3.15}$$

where ${}_{\{g\}}\mathbf{T}^{\{s\}} = {}_{\{g\}}\mathbf{T}^{\{e\}}\,{}_{\{e\}}\mathbf{T}^{\{s\}}$ is the translational plus rotational transformation from the sensor frame $\{s\}$ to the global frame $\{g\}$, passing through the end-effector frame $\{e\}$; K_P and K_I denote diagonal matrices of the proportional and integral regulators' gains, respectively.

Substituting eq. (3.15) to eq. (3.13) leads to the following:

$$\tau_d = J_\theta^T [K_P \left(x_a + {}_{\{g\}}\mathbf{T}^{\{s\}} \cdot \left(K_P \Delta x'(t) + K_I \int \Delta x'(t) dt\right) - x_a\right) - K_D \dot{x_a}] + g(q) \tag{3.16}$$

where $\Delta x'$ is given in eq. (3.11) and eq. (3.7). By substituting these equations we get the control law that relates the tactile feature vector s with the joint vector τ:

$$\begin{aligned} \tau_d = J_\theta^T [K_P \big(x_a + {}_{\{g\}}\mathbf{T}^{\{s\}} (K_P \mathbf{P}(s_d(t) - \mathbf{J^{-1}} s_1(t)) + \\ K_I \int \mathbf{P}(s_d(t) - \mathbf{J^{-1}} s_1(t)) dt\big) - x_a) - K_D \dot{x_a}] + g(q) \end{aligned} \tag{3.17}$$

The PI regulators are added after the projection matrix in order to control how fast the tactile feature error-vector will be regulated to zero. Thus, each element in this error vector of the pose of the robot's end-effector will be subjected to its own regulator.

Figure 3.11 shows how these PI regulators are added in the global control strategy. When the end-effector interacts with the environment, the tactile sensing array provides the information about the contact frame given by a two-dimensional distribution of pressures $I(x, y)$, from which the actual tactile features are extracted. The control cycle starts by computing the deviation $\Delta \boldsymbol{s}$ of the current tactile feature vector $\boldsymbol{s_a} = \boldsymbol{s_2}$ (point-contact) or $= \boldsymbol{s_2'}$ (edge-contact) from the desired set of features $\boldsymbol{s_d}$. This error is fed into the mentioned above PI controllers of eq. (3.17), acting independently on all feature-error components ${}_{\{s\}}\Delta x'$ (eq. (3.11)) in the Cartesian sensor frame $\{s\}$. The contact frame error ${}_{\{s\}}\Delta x''$ subjected to PI regulators is then translated and rotated from the sensor frame to the base (global) frame by applying the transformation matrix ${}_{\{g\}}\boldsymbol{T}^{\{s\}}$ (more details are given in Appendix C). This error ${}_{\{g\}}\Delta x''$ will be sent as an input to the inner Cartesian controller of the robot (eq. (3.13)) so that the end-effector of the robot is driven to the desired pose $\boldsymbol{x_d}$.

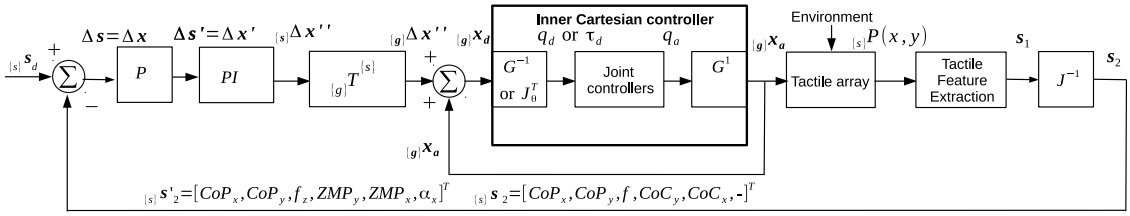


Figure 3.11: Block diagram of the external hybrid tactile-position controller.

Therefore, the proposed tactile servoing scheme is based on two different control loops: an outer loop using the tactile information and an inner loop implementing the robot's motion controller. The outer loop gives the pose set points to the inner loop based on the convergence of the tactile feature error. In fact, this developed structure is the extension of the external hybrid force-position control [Perdereau 96] by substituting the use of the force feedback with the tactile feedback. Thus, this new tactile servoing scheme is called "**external hybrid tactile - position control**" and is one of the main contributions described in this chapter. The basic steps of this external hybrid tactile - position control

are given in the following algorithm:

Algorithm 2: External hybrid tactile-position control scheme

input : Current $\mathbf{s_a}$ and desired $\mathbf{s_d}$ features from tactile sensor, current robot pose $\mathbf{x_a}$
output : Robot's end-effector command $\mathbf{x_d}$ for the inner position controller

For $(t = 0$ to t_{end} by $\Delta t)$
  $\mathbf{J^{-1}} \leftarrow \mathbf{J^{-1}}(\mathbf{s}, \mathbf{x_a})$ *// Tactile Jacobian computation (eq. (3.8)) and update (eq. (3.6))*
  $\Delta\mathbf{s} \leftarrow \mathbf{s_d} - \mathbf{J^{-1}}\mathbf{s_a}$ *// Computation of tactile feature error (eq. (3.7))*
  $\Delta\mathbf{x}' \leftarrow \mathbf{P}\Delta\mathbf{s}$ *// Applying the task projection matrix (eq. (3.11))*
  $\Delta\mathbf{x}'' \leftarrow PI(\Delta\mathbf{x}')$ *// Convergence of the tactile error using the PI regulators (eq. (3.15))*
  $\Delta\mathbf{x}'' \leftarrow {}_{\{g\}}\boldsymbol{T}^{\{s\}}\Delta\mathbf{x}''$ *// Transformation from the sensor frame to the global frame (Appendix C)*
  $\mathbf{x_d} \leftarrow \mathbf{x_a} + \Delta\mathbf{x}''$ *// Update of the desired end-effector pose with the tactile error (eq. (3.14))*
  Send $\mathbf{x_d}$ to the inner Cartesian pose controller (eq. (3.13))

3.4.4 Geometric consistency: natural and artificial constraints

The robustness of the controller should be analyzed with regard to the geometric consistency (constraints compatibility) since there are different kinematic constraints that can change during physical interactions with objects (the tactile error convergence is described in appendix D).

In order to control position/orientations and forces at the end-effector of a manipulator, Raibert and Craig [Raibert 81] proposed hybrid position/force control approach and M. Mason [Mason 81] proposed to decompose the constraints in the Cartesian space. Following this decomposition, the artificial constraints on the forces and velocities can be set separately. Not all of the directions in the Cartesian space can be artificially constrained when there is a physical interaction of the robot with a rigid environment. Since the robot's end-effector can not move freely in some directions due to the contact with the environment, there are natural constraints on motions (i.e. velocities are zero) along these directions. There are also directions in which forces maybe naturally constrained (forces/moments are zero) depending on the contact type and the task given to the robot. The constraints related to the tasks will be described in the following Chapter 4. The natural constraints are different for the point- and edge-contact types; therefore, the artificial constraints must obey the changes in natural constraints. For instance, for the edge-contact type, the orientation of the edge around the normal axis of the sensing frame (i.e.

z-axis) can be measured and controlled, we can thus artificially constrain the velocity w_z. However, the robot cannot transmit any moment of force around this axis since there is free motion around it and, this moment of force is thus naturally constrained ($\mu_z = 0$).

In order to represent this kind of constraints, we should decompose the interaction forces and possible motions as follows:

$$\boldsymbol{v}^T \boldsymbol{f} = v_x f_x + v_y f_y + v_z f_z + w_x \mu_x + w_y \mu_y + w_z \mu_z \tag{3.18}$$

where v and w are the translational and rotational velocities (twist), f and μ are the forces and moments (wrench). When the forces and velocities are decomposed, we can establish a **reciprocity condition** indicating that we cannot control the force and the velocity in the same direction simultaneously:

$$\boldsymbol{v}^T \boldsymbol{f} = 0 \tag{3.19}$$

Following this condition, we can establish all artificial and natural constraints (Table 3.2) for the point- and edge- (along x- or y-axis) contact types. These constraints also consider the assumptions given in Section 3.2. For instance, the tangential forces in the following directions are considered to be negligible: along x and y axes, and around z-axis. Therefore, forces in these directions cannot be constrained artificially for any type of contact as they are always naturally constrained ($f_x = f_y = \mu_z = 0$). In addition, the translational velocity along z-axis could be also considered negligible since the sensor's surface deformations are relatively small and the environment is rigid during physical interactions v_z is thus always naturally constrained ($v_z = 0$). Similarly, the rotational motion around y (or x) axis is naturally constrained during the contact with an edge of an object when the edge is along x (or y) axis of the sensor attached frame due to the infinitesimal deformations of the sensing surface $w_y = 0$ (or $w_x = 0$).

In the case of the point-contact, the moments of force around all axes are naturally constrained ($\mu_x = \mu_y = \mu_z = 0$) since the contact area is infinitesimally small and the robot can not transmit these moments of force to the planar sensor and the friction forces are considered to be negligible. All other motions and forces are directly related to the variations of the tactile features (described in Chapter 2) in the contact frame as

depicted in Fig. 2.22 for the tactile features related to the point-contact and in Fig. 2.23 for the features related to the edge-contact. Thereby, these tactile feature variations can artificially constrain (as catalogued in third column of Table 3.2) the remaining motions and forces through the projection matrix and the touch dependent inverse tactile Jacobian introduced in Section 3.4.2.

Contact type	Natural constraint	Artificial constraint	$\boldsymbol{v}^T\boldsymbol{f}$
Point	$f_x = 0$ $f_y = 0$ $v_z = 0$ $\mu_x = 0$ $\mu_y = 0$ $\mu_z = w_z = 0$	$v_x \sim \Delta CoC_x$ $v_y \sim \Delta CoC_y$ $f_z \sim \Delta force$ $w_x \sim \Delta CoC_y$ $w_y \sim \Delta CoC_x$ $-$	0
Edge (along x-axis)	$f_x = 0$ $f_y = 0$ $v_z = 0$ $\mu_x = 0$ $w_y = 0$ $\mu_z = 0$	$v_x \sim \Delta CoC_x$ $v_y \sim \Delta CoC_y$ $f_z \sim \Delta force$ $w_x \sim \Delta CoC_y$ $\mu_y \sim \Delta ZMP_x$ $w_z \sim \Delta\alpha_z$	0
Edge (along y-axis)	$f_x = 0$ $f_y = 0$ $v_z = 0$ $w_x = 0$ $\mu_y = 0$ $\mu_z = 0$	$v_x \sim \Delta CoC_x$ $v_y \sim \Delta CoC_y$ $f_z \sim \Delta force$ $\mu_x \sim \Delta ZMP_y$ $w_y \sim \Delta CoC_x$ $w_z \sim \Delta\alpha_z$	0

Table 3.2: Natural and artificial constraints during the interactions.

In fact, during the point-contact type, by setting $\alpha_i = 0$ in J^{-1} and $\gamma_x = \gamma_y = \gamma_z = \gamma_{wx} = \gamma_{wy} = 1$ in P, the end-effector's pose error is given by:

$$_{\{s\}}\boldsymbol{\Delta x} \propto [\Delta CoC_x, \Delta CoC_y, \Delta force, \Delta CoC_x, \Delta CoC_y, -]^T \tag{3.20}$$

where the artificial constraint $\Delta CoC_x = CoC'_x - CoC_x$ (and $\Delta CoC_y = CoC'_y - CoC_y$) is the error between the desired CoC'_x (and CoC'_y) and actual CoC_x (CoC_y) centers of contact in x- (and y-) axis and the artificial constraint $\Delta force = f_d - f_a$ is the error between the desired f_d and actual f_a normal force along z-axis. During the edge-contact type detected according to eq. (3.6), J^{-1} switches to controlling the moment of force (represented by

ZMP) by setting $\alpha_i = 1$; the P remains the same. Then the error of the end-effector's pose is given by:

$$ {}_{\{s\}}\Delta\boldsymbol{x} \propto \begin{cases} [\Delta CoC_x, \Delta CoC_y, \Delta force, \Delta CoC_y, \Delta ZMP_x, \Delta\alpha_z]^T, & \text{Edge along x-axis} \\ [\Delta CoC_x, \Delta CoC_y, \Delta force, \Delta ZMP_y, \Delta CoC_x, \Delta\alpha_z]^T, & \text{Edge along y-axis} \end{cases} \tag{3.21} $$

where the artificial constraint $\Delta ZMP_x = ZMP'_x - ZMP_x$ (or $\Delta ZMP_y = ZMP'_y - ZMP_y$) is the error between the desired ZMP'_x (or ZMP'_y) and actual ZMP_x (or ZMP_y) Zero-Moment-Point along x (or y) axis; $\Delta\alpha_z = \alpha'_z - \alpha_z$ is the error between the desired α'_z and actual orientation of the edge α_z. When the edge is along x-axis we can constrain the rotational motion around this axis with ΔCoC_y, whereas when the edge is along y-axis we can constrain the ΔZMP_y. These tactile features were described in Sections 2.3.1 and 2.3.2 for the point- and edge-contact types, respectively. Their increments are the input to the hybrid external tactile-position controller described in the previous section. Finally, different sets of tactile features will be selected according to the tasks as described in Section 4.2.

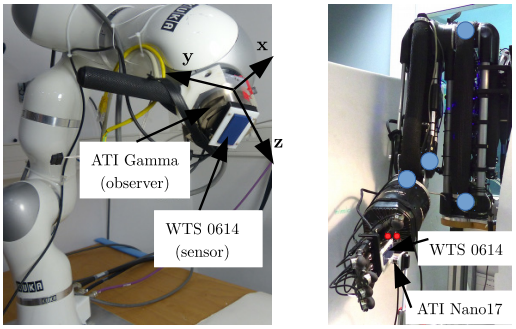
3.5 Experimental evaluation

In the previous section, we introduced the methodology of the tactile controller. The next step is to validate this controller with a real pressure sensing array installed on the end-effector of a real robot arm. Before conducting the experiments, the pressure sensing array was calibrated in order to identify its measuring force ranges and justify its applicability. The 6×14 (3.4×3.4 mm per tactel, with total size of 25×50 mm) Weiss Robotics WTS0614 [Robotics 14] piezo-resistive tactile sensing array (tactile sensor) was used The sensor is covered with a layer of an elastic material – electro conductive rubber. In Appendix E, its sensitivity, force range, compliance constant and repeatability are obtained through calibration.

3.5.1 Real robots

Experiments have been carried out using the pressure array described above and one of two robot arms. The first arm is a 7 dof KUKA LWR 4+ robot arm and the second arm is a 6 dof Shadow robot arm with a robot hand.

These two robot platforms used in the experiments are illustrated in Fig. 3.12. The KUKA arm had the tactile sensing array attached onto the end-effector so that the geometrical centers of the end-effector and the sensor surface coincide. The origin of the sensor attached frame is shown in Fig. E.3. On the Shadow hand, the sensor was attached onto the palm so that the middle of the palm and sensor frames coincide.



(a) (b)

Figure 3.12: (a) 7-dof KUKA LWR robot arm and (b) 4-dof Shadow robot arm plus 24-dof Shadow robot hand platform with the Weiss Robotics WTS 0614 tactile sensing array on the end-effectors.

Shadow Robot arm and hand

The Shadow robot arm and hand manipulation platform consists of two robots: a 4-dof (joints are depicted with the blue circles in Fig. 3.12a) pneumatic arm actuated by McKibben[5] air muscles and a 24-dof tendon driven electric anthropomorphic robot hand. With the two tendon driven joints (depicted with the red circles in Fig. 3.12a) at the wrist of the hand the manipulation system, gets six dof. The arm controller is implemented using Ubuntu Lucid upgraded for Real Time Application Interface 3.8 with the installed ROS Electric distribution. The two wrist joints of the robot hand are controlled using ROS

[5]after the developer J.L. McKibben, 1950 [AirMuscle 16]

Indigo distribution installed on the computer (PC) dedicated to control the hand and the joints of the fingers will not be used. The PC dedicated to the hand is connected to the one one dedicated to the arm controllers via a local network.

Each joint tendon of the hand is equipped with a strain gauge measuring the tendon's tension, which is controlled by a PID control loop with feed-forward term running at a rather high rate of 5 kHz and compensating friction forces. The joint position control scheme is composed of velocity PID and position PI control loops and runs at the frequency of 1 kHz. Due to inherent backlash of the tendon driven mechanism, a dead-zone has been introduced in the position controller[6]. The steady-state error of this control loop is typically 2 degrees. This implies that the precision at the palm that the controller can realize is about 2 mm.

The air muscles of the arm are very compliant[7], which complicates the controller design to get precise positioning in Cartesian space. The compliance of the arm in the normal direction of the palm is shown in Fig. 3.13. The forces are measured with the ATI Nano17 force sensor and the displacement in the Cartesian space derived from the Forward Kinematics of the arm. Thereby, the stiffness of the arm in the z direction of the palm frame is ~ 180 N/m (0.18 N/mm). The joint control loops of the arm run at 1 kHz.

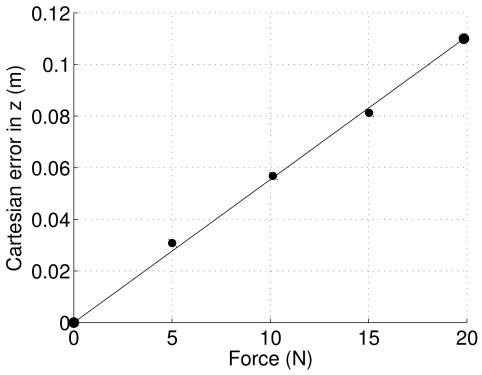


Figure 3.13: Compliance of the arm in the Cartesian space along z-axis of the palm. Two measurement points are in bold to illustrate that the arm can deviate from its desired position up to 120 mm when a 20 N disturbance force is applied.

[6]the objective of these dead-zones is to stop the motor from rewinding when the joint is "sufficiently" close to the desired position, making the system more energy efficient.

[7]which is good in regards to the safety issues during the human-robot interactions.

KUKA arm

The 7 dof Kuka arm is operated in the Cartesian space compliance mode using the default stiffness and damping parameters of 1000 N/m and 0.7 Ns/m (in the joint-space, the stiffness and damping parameters are set to 200 Nm/rad and 0.7 Nms/rad, respectively). The software implementation is realized via the Orocos Real-Time Toolkit (RTT) and the RTT to ROS integration package installed on the KUKA arm PC with Linux Ubuntu 14.04 distribution patched with a Xenomai real-time development framework. The Cartesian position controller sends new commands at 1 kHz through the Fast Research Interface (FRI) library for the KUKA Lightweight Robot [Schreiber 10]. The control law of the Cartesian motion controller represents a Cartesian virtual spring $\boldsymbol{k_c}(\boldsymbol{x}_d - \boldsymbol{x}_{msr})$, where $\boldsymbol{x}_{msr}$ is the actual measured pose, and the joint torques are computed by the Cartesian law according to eq. (3.13). The gravity (and dynamic disturbances) compensation term are calculated within the KUKA robot controller (KRC).

Sampling rates and special considerations of the implementation with the KUKA arm.

The maximum sampling rate of the sensor connected via USB2.0 with a computer is around 270 Hz. However, the sensor data reading loop is set to operate at 200 Hz. The image processing loop for tactile data processes the data at 140 Hz. Therefore, the tactile servoing controller has to operate at 100 Hz to ensure that the master computer acquires data from the overall system. The frequencies of the control loops, the time for tactile data processing and the data acquisition are summarized in Table 3.3.

Thanks to the use of the ROS framework, the implementation of the tactile servoing controller was not limited to one specific runtime system. In order to not overload the computer that was communicating with the KUKA controller, the tactile image processing was running on a separate PC connected with the latter through a local network.

In the real implementation of the tactile controller (eq. 3.14), the current pose of the KUKA arm was measured once at the first cycle and then at each next cycle the commanded pose was assumed to be the current pose for the future cycle:

$$\boldsymbol{x}(t+1) = \boldsymbol{x}(t-1) + \Delta\boldsymbol{x}'' \tag{3.22}$$

where $\boldsymbol{x}(t+1)$ is the desired pose, $\boldsymbol{x}(t-1)$ is the desired pose from the previous cycle and assumed to be the current actual pose $\boldsymbol{x}_a$, $\Delta\boldsymbol{x}''$ is the contact frame error. The only difference with the theoretical one is that the current pose is not measured at each cycle, but assumed to be equal to the previously commanded pose. This assumption is valid as long as the robot reaches the desired pose at the given cycling rate. Fig. 4.30 in Section 4.5.3 provides justification for this assumption. That figure shows that the controller can track the desired feature values given by a square wave function within its bandwidth.

Description	Rate
Cartesian motion controller (PD + Gravity, eq. (3.13))	1 kHz
Tactile servoing controller (eq. 3.14). The controller does not send new commands if there is no contact.	100 Hz
Tactile Image Processing (feature extraction).	140 Hz
Tactile sensor data acquisition.	200 Hz

Table 3.3: Rates of control loops and data acquisition/processing for the tactile servoing with the KUKA arm.

3.5.2 Evaluation of the tactile controller

The following section benchmarks the performance of the tactile controller by different experiments using stiff and deformable objects (Table 3.4). The controller servos one or multiple tactile features simultaneously during interactions with these objects.

Exp.	Object	Contact type	Features	Jac. matrix J^{-1}	Task matrix P	Description
1	Indenter (rigid)	Point	CoC_x, CoC_y	$\alpha_1 = \alpha_2 = 0$	$diag$ $(1,1,0,0,0,0)$	Controller response for the KUKA arm
2	Indenter (rigid)	Point	CoC_x, CoC_y	$\alpha_1 = \alpha_2 = 0$	$diag$ $(1,1,0,0,0,0)$	Controller response for the Shadow arm
3	Cylinder (rigid)	Edge	Force	$\alpha_1 = \alpha_2 = 0$	$diag$ $(0,0,1,0,0,0)$	Controller response for the Shadow arm. Force $\propto$ Pressure & Contact Area
4	Cylinder (rigid)	Edge	Force	$\alpha_1 = \alpha_2 = 0$	$diag$ $(0,0,1,0,0,0)$	Estimation of force by two approaches: Force $\propto$ Pressure & Contact Area and Force $\propto$ Contact Area
5	Cylinder (rigid)	Edge	Force	$\alpha_1 = \alpha_2 = 0$	$diag$ $(0,0,1,0,0,0)$	Controller response for the KUKA arm. Force $\propto$ Contact Area only
6	Indenter (rigid)	Point	CoC, Force	$\alpha_1 = \alpha_2 = 0$	$diag$ $(1,1,1,0,0,0)$	CoC & Force responses for the KUKA arm
7	Metal bar	Edge	Orientation	$\alpha_1 = \alpha_2 = 0$	$diag$ $(0,0,0,0,0,1)$	Orientation response
8	Metal bar	Edge	Orientation, force, CoC_y	$\alpha_1 = \alpha_2 = 0$	$diag$ $(0,1,1,0,0,1)$	Response of set of three features
9	Foam (soft)	Edge	ZMP_x	$\alpha_1 = 1, \alpha_2 = 0$	$diag$ $(0,0,0,0,1,0)$	Controller response for ZMP_x
10	Bar (rigid)	Partial edge	ZMP_x	$\alpha_1 = 1, \alpha_2 = 0$	$diag$ $(0,0,0,0,1,0)$	Controller response for ZMP_x with partial edge
11	Cylinder (rigid)	Edge	ZMP_y	$\alpha_1 = 0, \alpha_2 = 1$	$diag$ $(0,0,0,1,0,0)$	Controller response for ZMP_y

Table 3.4: List of experiments with descriptions for validating tactile controller responses.

Experiments 1 and 2: evaluation of CoC tactile feature control

Fig. 3.14a and b show the setups used to evaluate the CoC control for the Shadow and KUKA arms, respectively. The Shadow arm was pushing against the indenter that was attached on top of the ATi Nano17 force and torque sensor for ground-truth observations. In the case of the Kuka arm, it was also pushing against the indenter but the observer (i.e. the force/torque sensor) was installed directly on the end-effector. In order to validate CoC tracking visually, a transparent grid and a marker were attached onto the sensing surface (Fig. 3.14, a). In addition, the surface of the sensor was covered with a thin, flexible and transparent plastic to protect the resistive rubber from wear-off.

The inverse tactile Jacobian matrix's variable elements are set to zero $\alpha_1 = \alpha_2 = 0$ and the tactile projection matrix is equal to $diag\ (1, 1, 0, 0, 0, 0)$. The desired tactile feature value is given by the Heaviside step function.

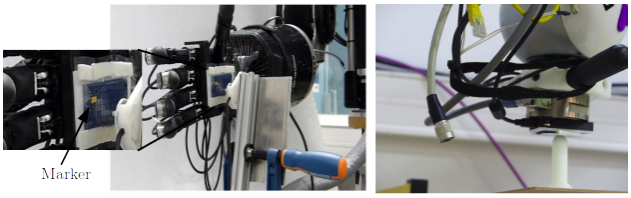


Figure 3.14: Experimental setups for the evaluation of the CoC control for (a) the Shadow arm and (b) the KUKA arm.

The results of servoing the CoC with the KUKA arm[8] and the Shadow arm[9] are shown in Fig. 3.15 and Fig.3.16, respectively. Fig. 3.15a and b illustrate the CoC_x and CoC_y tactile features when controlling them with the KUKA arm. The history of the desired and feedback CoC values are depicted in green and red, respectively. The first marker "x" indicates the time of the new set point for CoC' and the second marker "x" indicates the time when the level of the feedback signal reaches 95% (dashed line) of the level of the new set point. The response is over-damped (i.e. there is no overshoot) to prevent the point of contact from moving outside the sensing surface.

[8]https://www.youtube.com/watch?v=swYIfc81fQg&index=12&list=UUQEGPPSY-gQjDm9TGokCDuA

[9]https://www.youtube.com/watch?v=YnFMTfZCZNk&list=UUQEGPPSY-gQjDm9TGokCDuA&index=2

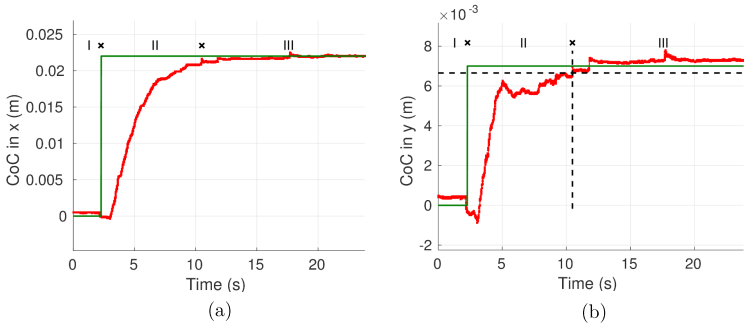


Figure 3.15: CoC control responses for the KUKA arm: (a) in x-axis , (b) in y-axis.

In the case of the Shadow arm, the new set point was given along x-axis only (length is bigger along x-axis), while the CoC in y-axis was maintained at the same coordinate.

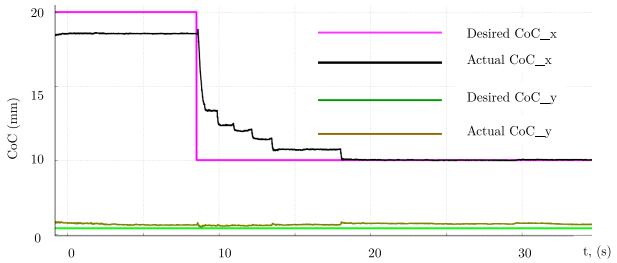


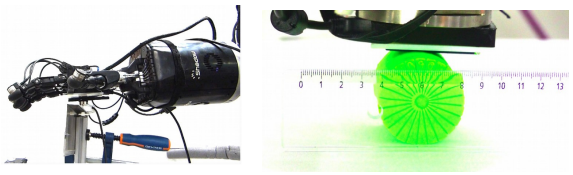
Figure 3.16: CoC control responses for the Shadow arm.

The responses for the Shadow air muscled arm are not as smooth as for the motor driven KUKA arm. This is due to the difference between the way of how the torque controlled motor arm and the air pressure controlled arm are actuated. In order to actuate the Shadow arm, a pressure controller opens and closes the valves of the arm in order to inflate up the soft stretchable muscles (the muscles are composed of inner rubber tubes and external braided polyester mesh sleeves). At the times of pumping the tubes, there are distortions that are negligible in the human - robot interactions due to the compliance of the arm but noticeable in the tasks requiring a higher precision, e.g. tactile servoing tasks. Due to these imprecisions of the position control of the pneumatic arm, in the following experiments (except the force control) the KUKA arm will be used to validate the effectiveness of the tactile controller only. However, the Shadow robot arm/hand manipulating platform can be used for safe human-robot co-manipulation tasks and for

dexterous in-hand manipulation tasks.

Experiments 3, 4, 5: evaluation of force tactile feature control

Fig. 3.17 shows the setups used for the evaluation of the force responses for the Shadow and KUKA arms. The inverse tactile Jacobian matrix's variable elements are manually set to zero $\alpha_1 = \alpha_2 = 0$ and the tactile projection matrix is equal to $diag\ (0, 0, 1, 0, 0, 0)$ so that the robots were commanded to servo Δz, which is proportional to the normal force. The desired set points were given by a square wave function.



(a) (b)

Figure 3.17: Experimental setups for the force control evaluation: (a) the Shadow Robot arm with hand and (b) the KUKA arm are in contact with a stiff environment.

The Shadow arm[10] was pushing against the ATi Nano17 sensor attached to a stiff environment and the KUKA arm[11] (with the ATi Gamma force sensor on the end-effector) was pushing against a rigid object lying on a stiff table. The force sensors were acting as ground-truth observers providing absolute force measurements. Fig. 3.18 shows the force tactile feature responses controlled with the Shadow arm. This force was estimated by the sum of all the pressure values of tactels (eq. (2.4)) and later multiplying this sum with the area of contact. According to the results of servoing the force, the overall performance is acceptable and the force responses can track the set-points despite the noise of the sensor. Nevertheless, the imprecision of the pneumatic arm and its compressor's vibration noise does not allow to finely control the contact frame. Moreover, when we began experiments with the KUKA arm, that is more precise in the Cartesian motion, we found that the tactile servoing diverged mainly due to the tactile sensor noise. This noise obliged us to

[10] https://www.youtube.com/watch?v=UJcwKWasWmE&index=4&list=UUQEGPPSY-gQjDm9TGokCDuA

[11] https://www.youtube.com/watch?v=HK7x51U7CiA&index=13&list=UUQEGPPSY-gQjDm9TGokCDuA

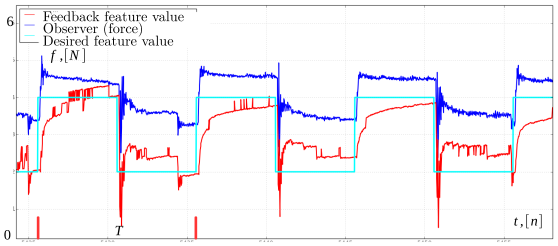


Figure 3.18: Experiment 3: force responses for the 4 dof pneumatic Shadow arm with 2 dof wrist joint of the tendon-driven Shadow hand. The period of the square-wave function is T = 10 s.

find another way to estimate the force tactile feature.

Two ways to estimate the applied force

In fact, as discussed in Section E.2 (calibration of the sensor), the measurements of each tactel in the tactile array vary depending on the environmental conditions and the location of the tactel within the sensing frame. In order to mitigate the nosiness of the estimated force values, the force can be set proportional to the area of contact[12] only, because it increases with the applied force due to the compliant surface. Fig. 3.19 depicts the forces estimated in both ways: the pressure multiplied by the contact area (the red line) and the contact area multiplied by a scaling factor (the green line). The ground-truth observer's values are depicted with the blue line. The force estimate based on the area of contact is far less noisy compared to the force estimate based on the pressure values. This is due to the inherent sensor noise and the creep behavior of the sensor discussed in Section E.2.

From here on, we will use the contact area for force servoing with tactile sensors. Fig. 3.20 shows the experimental results of servoing the force estimated in this way with the KUKA arm. We observe that there is no divergence of the force response, when pressure values are not considered. Finally, the precise estimate of the force can be derived from the area by multiplying it with an appropriate scaling factor, which is a ratio between the area of contact and the ground-truth force measurements (difference between the green and blue lines in Fig. 3.20). This ratio will be calibrated for each task according

[12]https://www.youtube.com/watch?v=HFqa3PVGjWU&list=UUQEGPPSY-gQjDm9TGokCDuA&index=6

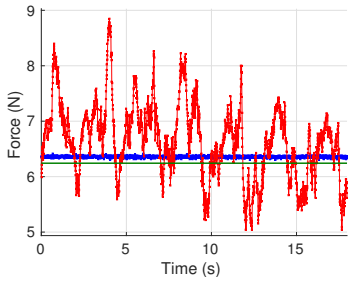


Figure 3.19: Experiment 4: comparison of the force estimations: Force $\sim$ Pressure & Contact Area (red), Force $\sim$ Contact Area (green), ground truth observer (blue).

to the force range.

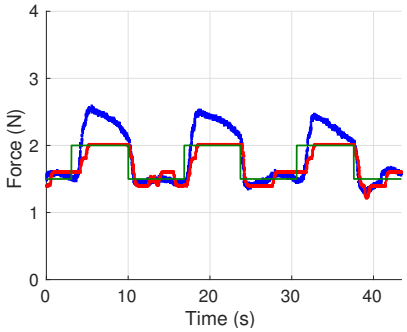


Figure 3.20: Experiment 5: force responses with the KUKA arm. The desired, feedback, and ground-truth observer force measurements are depicted with green, red, and blue lines.

Experiment 6: evaluation of simultaneous CoC and force tactile features control

The experimental setup for the evaluation of the feature error obtained for the force and CoC features was the same as for the evaluation of *CoC* control (Fig. 3.14 b). The force and CoC can be controlled simultaneously[13] by setting the tactile projection matrix equal to $diag\ (1, 1, 1, 0, 0, 0)$ and the inverse tactile Jacobian matrix's variable elements set to zero ($\alpha_1 = \alpha_2 = 0$). Since the Kuka arm was pushing against the indenter with a rather small contact surface, the force was estimated from the pressure and contact area. In order to evaluate the performance of the controller, the robot was commanded to maintain the

[13]https://www.youtube.com/watch?v=8WVTYQwmZQQ&list=UUQEGPPSY-gQjDm9TGokCDuA&index=11

force at some level and servo the desired CoC in x- and y- axes (Fig. 3.21a-c).

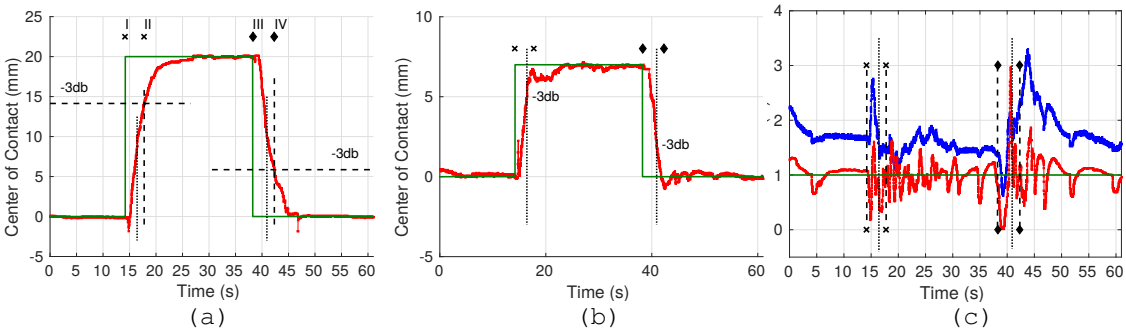


Figure 3.21: Results of servoing with the KUKA arm the features CoC_x, CoC_y and force simultaneously: a – CoC_x, b – CoC_y, c – $force$. The desired, feedback, and observers values are depicted with the green, red, and blue colors. The first marker "x" (event I) indicates the time of the new set point (rising edge) for CoC_x and the second marker "x" (event II) indicates the time when the level of the feedback signal for CoC_x (CoC_y) reaches $-3dB$ of the level of the new set point. The first and second markers "diamond" (event III and IV) indicate the same but for the falling edge.

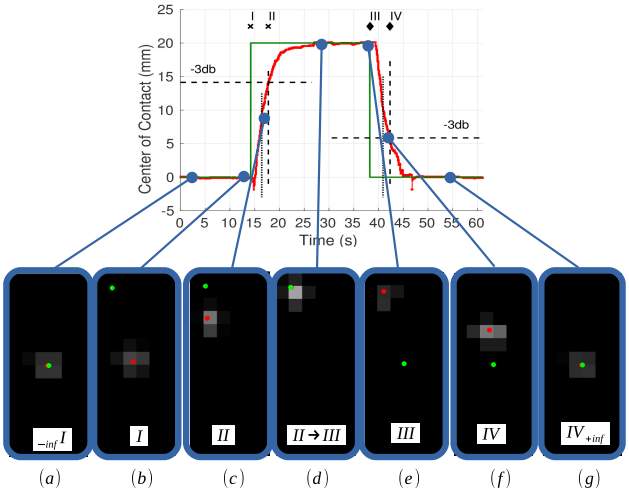


Figure 3.22: Tactile images with CoC' (green) and CoC (red). a - initial event, b - event I: new set point (state of contact), c - event II: transition to the new state of contact, d - desired state, e - event III: new set point, f - event IV: transition to the new state of contact, g - desired state.

In the experiment we can identify four main events according to the evolution of the CoC tactile feature:

- Event I: Set points for COC are fixed (Fig. 3.22b). The desired CoC_x and CoC_y are given by the Heaviside step function.

- Event II: CoC_x reaches $-3dB$ of its desired value (vertical dashed line in Fig. 3.21a). In the case of CoC_y, it already converged (vertical dotted line in Fig. 3.21a and b). The robot reaches the desired value in one axis faster than in the other (Fig. 3.22c, illustration of the tactile image at this event), because the physical size in the y-axis is smaller than in the x-axis. A motion planner could be added to calculate a trajectory for equalizing error evolution of velocities along the two axes.

- Event III: The set point of the event I is reached and a new set point for CoC is given to return to the initial position (Fig. 3.22e).

- Event IV: CoC_x reaches $-3dB$ of its new desired value (Fig. 3.22f). Similarly to the event II, CoC_y converges faster than CoC_x. Finally, CoC arrives to its desired location as shown in Fig. 3.22g, both in x- (Fig. 3.21a) and in y- (Fig. 3.21b) axis.

In the case of the normal force, as shown in Fig. 3.21c, the desired value is tracked smoothly, except during two relatively short time periods (when contact force increases and oscillates): between the events I and II, and between the events III and IV. These oscillations during these periods of CoC feature convergence are caused by the following reasons:

- Impact of friction forces: Even though a thin and slippery film was attached to decrease friction forces, there is still a small friction coefficient and also the stick-to-slip effect [Ho 12] (i.e. the friction forces are higher at the incipient of a slip) that generate undesired gear forces.

- Impact of non-uniform responses of neighbor tactels in the tactile array: The neighbor tactels may have slightly different responses as discussed in Section E.2. This issue could be mitigated by artificially increasing the spatial resolution of the tactile image and then applying a two-dimensional Gaussian filter in order to smooth the discrete sensor values.

The histories of the end-effector position coordinates in x-, y-, and z- axes are shown in Fig. 3.23 from left to right, respectively.

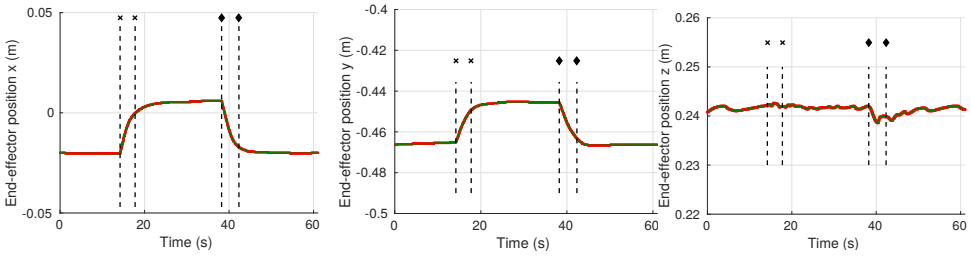
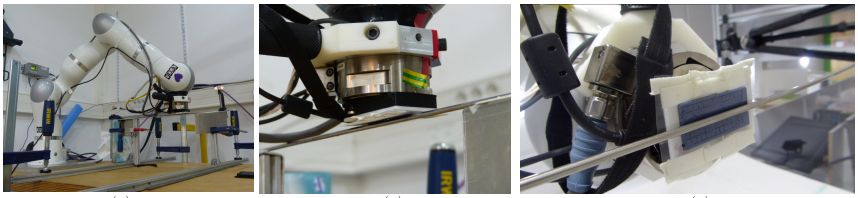


Figure 3.23: End-effector positions along the x-, y-, and z-axes left to right, respectively.

Experiment 7: evaluation of edge orientation tactile feature control

Fig. 3.24 shows the setup for the evaluation of the controller responses for the edge orientation control with the KUKA arm. In particular, the KUKA arm will rotate around a metal bar in order to control the orientation between its end-effector and the bar. The



(a) (b) (c)

Figure 3.24: Experimental setup for the evaluation of the edge orientation control: overall view - a; contact with metallic bar - b; local view at the contact frame - c.

inverse tactile Jacobian matrix's variable elements are manually set to zero $\alpha_1 = \alpha_2 = 0$ and the tactile projection matrix is equal to $diag\ (0,0,0,0,0,1)$. The desired angle α_z is given by the Heaviside step function and the orientation of the bar around the z axis of the sensing frame should converge to this value. This angle was limited within the range from -1.57 rad to + 1.57 rad (-90 to +90 degrees) in order to avoid singularities.

In this experiment, the desired orientation of the extracted line in the tactile image changed from -0.15 (phase I in Fig. 3.25, left-hand side) to +0.15 rad (phase III). And the end-effector will rotate around z-axis (right-hand side of Fig. 3.25) in order to make the error in this tactile feature converge to zero in less than 5 sec, as depicted by phase II in Fig. 3.25. The controller response for tracking the orientation of the detected edge is not smooth (phase II) due to the spatial resolution of the sensor (6×14). The history of the

force (not controlled) during the rotation is shown in the middle.

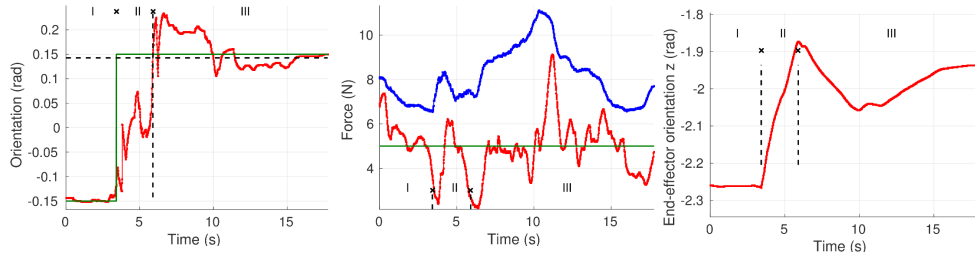


Figure 3.25: Control of α_z feature. From left to right, the controller responses for the edge orientation around the z-axis, force, and the effector's rotation around the z axis of the sensor frame for the KUKA arm during the control of the orientation of the edge of the metallic bar. The desired and feedback values are depicted with the green and red colors, respectively. The first marker x indicates the time of a new set point for the orientation. The second one corresponds to the transient level (-3dB of the desired value). The initial phase I is the phase of the new set point (state of contact), phase II is the transition to the new state of contact, phase III is the final state.

Experiment 8: evaluation of edge orientation, force and CoC tactile features simultaneous control

The setup in this experiment was the same as in the previous experiment (Fig. 3.24). At this time, instead of controlling one feature (α_z) only, the force and CoC_y were added into the tactile controller by setting the tactile projection matrix equal to $diag\ (0,1,1,0,0,1)$ and the inverse tactile Jacobian matrix's variable elements remained the same ($\alpha_1 = \alpha_2 = 0$). Fig. 3.26 illustrates the results of the controller response for α_z, force, and CoC_y. The

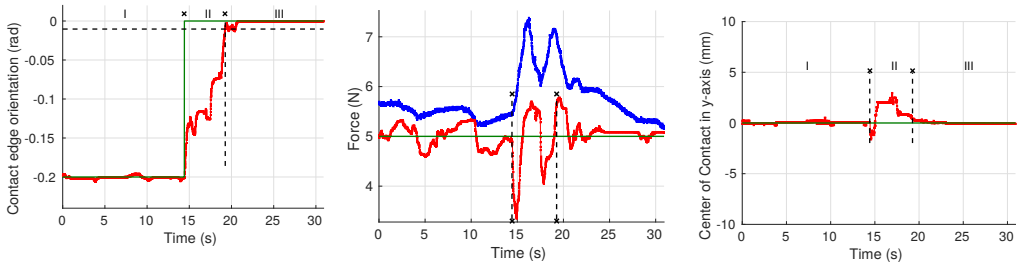


Figure 3.26: Simultaneous control of α_z, force, CoC_y features. From left to right, the controller responses for α_z, force, and CoC_y for the KUKA arm interacting with an edge of a metallic bar. The desired, feedback, and observer's values are depicted with the green, red, and blue colors. The first marker x indicates the time of a new set point for the orientation. The second one corresponds to the transient level (-3dB of desired value). The initial phase I is phase of a new set point (initial state of contact), phase II is transition to the new state of contact, phase III is the final state.

new set point (given by the Heaviside step function) was applied to α_z; the feature error converged in less than 5 s. The tactile controller maintained force and CoC_y feature values at 5 N and 0 mm, respectively. Phases I, II, and III in Fig. 3.26 represent different states: initial, transitional, and final. The border between the states is indicated with the marker "x". The error in force oscillated during the transition phase (due to the same reasons as in the experiment 6) and converged to zero during the final phase (Fig. 3.26, middle).

Fig. 3.27 illustrates the tactile images during the transition from the initial state to the final state of the contact. During this transition, not only the edge orientation is handled by the controller (as in the previous experiment) but also the deviation of the CoC_y feature (at the center of the sensor) as also shown in Fig. 3.26, right-hand side.

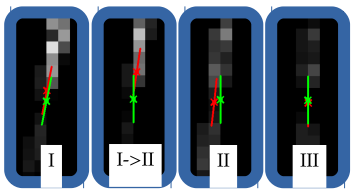
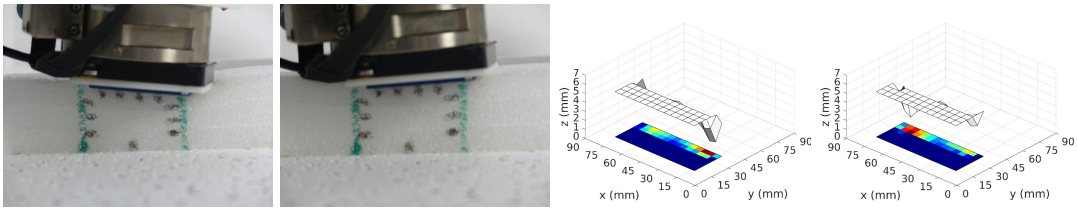


Figure 3.27: Tactile contact images corresponding to the states of the edge orientation α_z and CoC_y. From left-hand side to right-hand side, the first tactile image corresponds to the initial state I, the second one corresponds to the state with the new set point for the orientation $I \rightarrow II$, the third image illustrates the transient state (phase II), the last one corresponds to the final state (phase III).

Experiment 9: evaluation of ZMP feature control

ZMP_x feature (moment of force around y-axis) can be controlled by setting the tactile projection matrix to $diag\,(0,0,0,0,1,0)$ and switching on the variable $\alpha_1 = 1$ in the inverse tactile Jacobian matrix (with $\alpha_2 = 0$). Fig. 3.28a and b show the deformable foam that was used to evaluate the ZMP_x feature. It was painted with markers (black points) with a distance of 5 mm for visual illustration of the deformations. In addition, Fig. 3.28c and d show the pressure values of the sensing surface corresponding to the states of the deformations shown in Fig. 3.28a and b, respectively. The foam was located on one side of the sensor in order to have the edge-contact type.

Fig. 3.29 illustrates the ZMP feature servoing results: the proposed tactile controller



(a) (b) (c) (d)

Figure 3.28: Control of rotational motion around the y axis of the sensor frame using the proposed ZMP feature representing the moment of force about the axis. The robot is applying the moment of force in (a) the negative and (b) in positive directions. The deformations of the sensing surface when the ZMP_x is (c) minimum and (d) maximum correspond to the deformations of the foam in (a) and (b), respectively. The surface deformations are projected to one plane with different colors. The colors represent the pressure values of each tactel: from the lowest value in blue to the highest in red.

can follow the desired ZMP feature given by a square-wave function (green line), varying from -0.1 Nm (negative direction, Fig. 3.28, a) to +0.1 Nm (positive direction, Fig. 3.28, b), the blue curve represents the measured ground-truth torque signals (obtained with the ATi Gamma force/torque sensor), the red curve is the evolution of the actual value of the feature. The ground-truth torque measurements are larger than its estimation (ZMP_x feature) because the scaling factor between the actual torque and the derived ZMP feature was not high enough, but it could be better adjusted at the beginning of each experiment by a calibration procedure. In any case, as mentioned in the previous sections, this scaling factor is not so important because we can have only the estimation of the real forces and torques only, not their precise values due to sensor noise described in Section E.2.

We can also observe a time delay in the convergence of the feature error in ZMP_x produced by the compliance of the flexible foam (the foam behaves as a damper). Moreover, after each deformation, the resistive rubber may not return to its initial shape due to memory effect, which also contributes to this time delay.

Experiment 10: evaluation of ZMP feature control during a partial contact

The advantage of the ZMP feature is the ability to estimate a moment of force, even when the edge that is in contact with the sensor is not fully covering the sensing surface (touching two opposite boarders of the sensor). This partial edge-contact is shown in

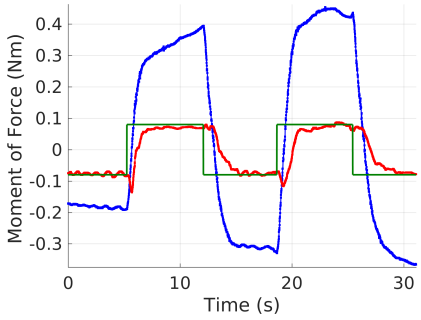
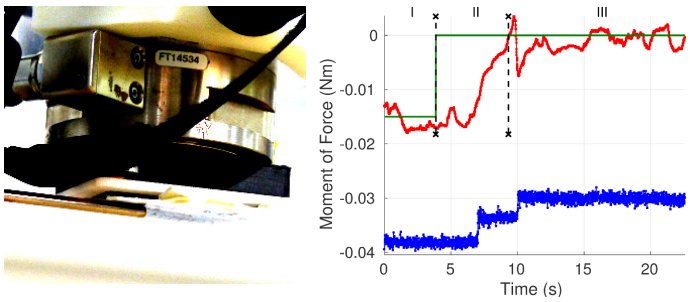


Figure 3.29: The ZMP_x feature control responses. The desired, feedback, and ground-truth observer's values are illustrated in the green, red, and blue colors, respectively.

Fig. 3.30a, in which a stiff metal bar is in contact with the sensing array. The KUKA robot is driven again (as in the experiment 9) by the tactile servoing controller to track the ZMP_x feature.

The results[14] of servoing the estimated moment of force during the partial edge-contact are shown in Fig. 3.30b. The controller reaches the desired ZMP_x feature value within 5 s (phase II): from $-0.015Nm$ (phase I) to $0Nm$ (phase III) in order to align the end-effector with the bar so that the applied line load is distributed uniformly. This control of the estimated moment of force with the tactile array over a rigid body is limited by the rotational compliance constant C_{wy} of the elastic layer of the sensing surface. Since this compliance constant is too small, the deformations are not noticeable as in the experiment with the flexible foam (it is less than 2 degrees given the sensor's length of 50 mm and the maximum compressible displacement of 2 mm). Nevertheless, the use of this feature will be useful in the tasks involving alignment and tracking of long edges (more details in Section 4.3). As in Fig. 3.29, there is a difference between the ground-truth torque measurements and the ZMP_x feature values that could be corrected by a scaling ratio obtained from previous calibrations.

[14]https://www.youtube.com/watch?v=HZgXu-Neh9w&list=UUQEGPPSY-gQjDm9TGokCDuA&index=8



(a) (b)

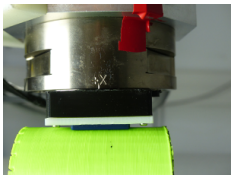
Figure 3.30: ZMP feature response with "partial" edge-contact. (a) Photo illustrating the setup with a stiff bar for the "partial" contact. (b) The ZMP_x feature response to a Heaviside step function. The desired, feedback, and ground-truth observer's values are illustrated in the green, red, and blue colors, respectively. The initial phase I is the phase of a new set point (initial state of contact), the phase II is the transition to the new state of contact and the phase III is the final state.

Experiment 11: evaluation of the ZMP_y feature control

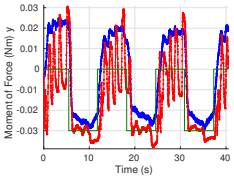
It was shown in the previous two experiments that the ZMP_x feature can be used to control the moment of force around the y-axis of the sensor's frame. In order to control the moment of force around the x-axis, the tactile controller incorporates the ZMP_y feature by setting the tactile projection matrix equal to $diag\,(0,0,0,1,0,0)$ and the variable $\alpha_2 = 1$ in the inverse tactile Jacobian matrix (with $\alpha_1 = 0$). Fig. 3.31a - d show the experimental setup for testing the ZMP_y feature control response, the controller's response, the contact surface deformations (tactels' values) when ZMP_y is minimum, and the contact surface deformations when ZMP_y is maximum, respectively. As in the previous experiments 9 and 10, the controller managed to make the ZMP value converge to the desired one. However, there are oscillations of the ZMP_y at zero moment, which are due to the sensor noise and the reduced spatial resolution along the y-axis: there are only six tactels.

3.6 Conclusions

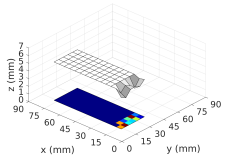
In this chapter, we have succeeded in elaborating a new tactile servoing controller for a robot arm equipped with a pressure sensing tactile array. This controller is built over the robot's joint position controllers by introducing a new touch dependent inverse tactile Ja-



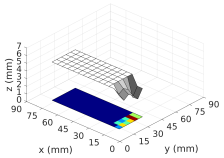
(a) Setup for the evaluation of ZMP_y



(b) ZMP_y and the moment force



(c) Contact profile at minimum ZMP_y



(d) Contact profile at maximum ZMP_y

Figure 3.31: Controller response for the ZMP_y feature. The desired, feedback, observer's values are depicted in green, red, and blue colors, respectively.

cobian and a new task dependent projection matrix for mapping tactile feature (described in the previous Chapter) errors into the robot's end-effector pose error. With this control structure, we can efficiently deal with the contact frames that occur during physical interactions. The experiments on the 6 dof Shadow robot arm-hand and the 7 dof KUKA arm show that this tactile controller produces good performance on different types of set-points (Heaviside's step and square wave functions) with objects of different stiffness and shapes without the need of changing robot controllers. Moreover, the controller can servo multiple tactile features simultaneously. The generality of the proposed control scheme will enable many other advances in the domain of dexterous manipulation as it will be shown in the next Chapter dedicated to the applications of the tactile controller.

Chapter 4
Task descriptions and applications

The tactile controller explained in the previous chapter is now used as a closed-loop servo system in different touch-driven robotic tasks. The description of a task is given by a history of the desired tactile feature values, the tactile projection matrix and the tactile Jacobian (Section 4.2). The tasks include: exploration of an unknown environment, classification of objects based on their stiffnesses, manipulation by rolling objects with cylindrical and ellipsoid shapes, and human-robot co-manipulation tasks. Section 4.3 describes the results of exploring the environment by sliding over edges. For instance, this functionality of the tactile controller is illustrated in an experiment of aligning the tactile sensing surface with a metal bar. In Section 4.4.2 deformable and rigid objects are distinguished based on their tactile responses by applying the Dynamic Time Warping algorithm. Results of manipulation by rolling objects with cylindrical, spherical and elliptical shapes are described in Section 4.5. Moreover, thanks to the tactile feedback in the controller, the shape of an object with unknown geometry can be reconstructed while manipulating it (Section 4.5.6). An example of a human-robot co-manipulation task is described in Section 4.6. The task is dedicated to share a load between a human operator and the robot while keeping the load at its equilibrium.

4.1 Introduction

In the past decades, tactile feedback was confined to to some applications like telemanipulation, haptics, legged robotics, and event-driven manipulation [Nicosia 01]. However, with advances in the tactile sensing technology, e.g. the bionic skin [Someya 14], the area of use of tactile feedback has recently increased. Nowadays, the use of tactile feedback for object exploration and recognition, material classification, or slip prediction has become rather popular [Pezzementi 11, Bekiroglu 11, Martinez-Hernandez 13, Wettels 14]. Thus, robot arms/hands equipped with tactile arrays can explore and recognize objects, control

forces in human-robot co-manipulation tasks, manipulate objects, and follow surfaces.

In the previous chapter, a new tactile servoing controller for touch-driven tasks was described. The main idea was that the pose of the robot's end-effector can be controlled by steering the contact frame, which is represented by the tactile features (given in Chapter 2), in a desired way. The current chapter focuses on the application of this framework to make a serial link robot manipulator perform different tasks thanks to the dynamic reconfiguration of the controller's projection matrix and the tactile Jacobian matrix. This makes the robot more dexterous, and allows it to perform different tasks and even a sequence of different tasks without any manual reconfiguration of the controller. In order to illustrate this functionality, we will describe four robot applications: deformable/rigid object recognition, exploration of objects, manipulation and shape reconstruction by rolling, and co-manipulation for controlling the center of mass of the object (Fig. 4.1). These robotic tasks are validated with several experiments that have been conducted with real data from the real noisy sensor. The main intent of these experiments is to show: 1) the efficacy of the presented controller and 2) the ability of touch to succeed in situations where vision would fail. Each of the applications illustrated in Fig. 4.1 requires its own **task description** including the desired tactile feature values to servo, which will be described in the following.

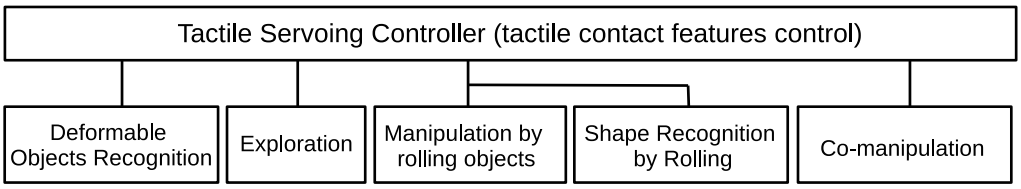


Figure 4.1: Applications of the proposed tactile servoing controller.

4.2 Task descriptions

Our goal is to develop a new unified framework that can be easily adopted in order to equip service and industrial robots with physical interaction capabilities needed for the successful accomplishment of widespread touch-driven robotic tasks. The framework unifies multiple applications and supports a wide range of actions far beyond the gener-

ally adopted edge following and object rolling tasks [Kappassov 15].

Any task is described by the motion that must be performed on an object in order to accomplish a particular goal by steering the contact frame in a desired way. This way is defined by the time-history of the set of the desired tactile features $\boldsymbol{s}^{ref}(t)$:

$$\boldsymbol{s}^{ref}(t) = \boldsymbol{s_d}(t) = \begin{cases} [CoC_x(t), CoC_y(t), f(t)], & \text{Point contact} \\ [CoC_x(t), CoC_y(t), f(t), ZMP_y(t), ZMP_x(t), \alpha_z(t)], & \text{Edge contact} \end{cases} \tag{4.1}$$

The time-history of the desired tactile features $\boldsymbol{s}^{ref}(t)$ together with the projection matrix $\boldsymbol{P}$ and the tactile Jacobian $\boldsymbol{J}^{-1}$ constitute the description of the physical interaction task (Fig. 4.2).

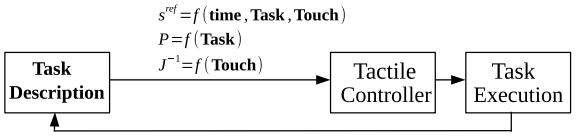


Figure 4.2: Task description.

Each task can be described by the following procedures, whose parameters are summarized in Table 4.1:

I Object exploration by edge tracking (Section 4.3):
0) initialize empty vector of contact points ${}_{\{g\}}\boldsymbol{c}$
1) detect new contact points ${}_{\{s\}}\boldsymbol{c*} = (x_c, y_c, z_c)$ in the sensor frame
2) transform to the global frame ${}_{\{g\}}\boldsymbol{c*} = {}_{\{g\}}\boldsymbol{T}^{\{s\}}\ {}_{\{s\}}\boldsymbol{c*}$
3) concatenate ${}_{\{g\}}\boldsymbol{c}$ with ${}_{\{g\}}\boldsymbol{c*}$
4) move the robot and go back to step 1 until the end of the exploration
5) fit a curve (2D line or 3D surface) to the ${}_{\{g\}}\boldsymbol{c}$
6) calculate stiffness of the explored object using eq. (4.4)

II Deformable object recognition (Section 4.4):
0) initialize empty vectors of the area of contact $\boldsymbol{A}$ (measured with a tactile sensor) and the ground-truth force $\boldsymbol{f}$ (measured by a force/torque sensor) at the beginning of the experiment t_0
1) squeeze-release an object. The desired force tactile feature is given by the square-wave function:

$$f_d(t) = |f_d| sgn[sin(2\pi(t - t_0)/T)] \tag{4.2}$$

where T is the period in sec.
2) get at each time step the area of contact A^* and concatenate with $\boldsymbol{A}$
3) get at each time step the ground-truth force f^* and concatenate with $\boldsymbol{f}$
4) $t = t + \Delta t$ and go to the step 1 until the end of the experiment
5) apply Dynamic Time Warping algorithm to $\boldsymbol{A}$ and $\boldsymbol{f}$ at the end of the experiment

Task	$\boldsymbol{P}$	$\boldsymbol{J^{-1}}$	$\boldsymbol{s}^{ref}$	On-line algorithm	Off-line algorithm
Obj. explor. by edge tracking	$\begin{pmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&0&0\\0&0&0&0&0&1\end{pmatrix}$	$\begin{pmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&1&0&0&-1&0\\1&0&0&-1&0&0\\0&0&0&0&0&1\end{pmatrix}$	CoC_x, f = const; CoC_y, α_z, $ZMP_x = 0$	update ${}_{\{g\}}\boldsymbol{c}$	Fit a curve (line or surface)
Deform. obj. rec.	$diag(0,0,1,0,0,0)$	$diag(1,1,1,0,0,0)$	f as in eq. (4.2)	update $\boldsymbol{A}, \boldsymbol{f}$	DTW
Obj. manip.	$diag(1,1,1,0,0,0)$	$diag(1,1,1,0,0,0)$	CoC' as in eq. (4.5); f = const.	update $\boldsymbol{z_e}$, ${}_{\{g\}}\boldsymbol{c}$	Shape reconstruction with eq. (4.10)
Co-manip.	$\begin{pmatrix}0&0&0&0&1&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&1\end{pmatrix}$	$\begin{pmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&1&0&0&-1&0\\1&0&0&-1&0&0\\0&0&0&0&0&1\end{pmatrix}$	ZMP_x, α_z, $CoC_y = 0$; f = const.	update ZMP_x	$\times$

Table 4.1: Task description parameters. Abbreviations: const. - constant, obj. - object, deform. - deformable, rec. - recognition, manip. - manipulation, explor. - exploration.

III Manipulation of known and unknown objects by rolling them (Section 4.5):
0) if the shape is known, calculate the desired CoC' as in eq. (4.5) otherwise set any CoC' within the sensing surface, and initialize the empty vectors of the end-effector poses along z-axis $\boldsymbol{z_e}$ and the vector of the contact points ${}_{\{g\}}\boldsymbol{c}$
1) roll the object by moving the robot's end-effector so that the actual CoC converges to the desired CoC' in the sensing frame
2) if the shape was not known, record the end-effector pose along z-axis $\boldsymbol{z_e}$ and contact points ${}_{\{g\}}\boldsymbol{c}$ and then apply eq. (4.10).

IV Object co-manipulation for sharing the object's load while keeping it in balance (Section 4.6):
0) orient the sensing surface in the opposite direction to the vector of gravity
1) put one side (edge) of an object on the sensing surface and hold the opposite side
2) keep the object at its equilibrium by moving the robot's end-effector so that any disturbance of the actual ZMP converges to zero.

These task descriptions determine affordances [Gibson 77], which can be described as all actions on an object or environment that are physically possible (confined by the natural and artificial constraints). These actions are described by the parameters given in Table 4.1 and will be explained in detail in the next sections.

4.3 Tactile object exploration

4.3.1 Review of experimental scenarios in exploration by edge following task

The main advantage of the robots with tactile arrays is their ability to follow the edge of an object continuously. Thus, the related tactile servoing works [Berger 91, Chen 95, Zhang 00, Li 13a] have been validated with this tactile servoing problem. Their proposed experimental scenarios are listed in the Table 4.2. Even though the approaches in [Berger 91, Chen 95, Zhang 00, Li 13a] were different, the experimental scenarios were the same and included the edge of an object that was straight or bended in one plane. The sensing surface of the sensor was always already aligned with the object's edge. As mentioned in Section 3.3.2, the number of controllable degrees of freedom was limited to 4 (3 translational motions along x-, y-, z- axes and 1 rotational motion around the z axis of the sensing frame). Moreover, all of the experimental scenarios included objects whose stiffness coefficients were constant, because they were either lying on a stiff table or rigidly fixed to a stiff environment. The proposed tactile servoing framework allows us to perform new experimental scenarios, shown at the bottom of the table 4.2, that overcome these limitations.

4.3.2 General description of the task

In order to explore an object, as for example a metallic bar, the location of which being unknown, a robot can move its end-effector with a tactile array until the appearance of a contact. Then, for the exploration of the bar, the motion of following the edge is performed by servoing the force, orientation of the edge, center of contact along the axis perpendicular to the edge (keeping the edge in the middle of the sensor), and the moment of force around the axis perpendicular to the edge (keeping a uniform line force distribution), thus setting $\gamma_z = 1$, $\gamma_y = 1$, $\gamma_{wx} = 1$, and $\gamma_{wz} = 1$ in $\boldsymbol{P}$. It is assumed that the edge lies along the x-axis of the sensor's frame and the edge is kept at the center of the sensor. The tangential forces are assumed to be rather small and not affecting sliding motions.

Approach	Object	Bended in:	Stiffness	dofs	Sequential execution
[Berger 91]	1) A metal box and 2) S-shape object	2D: 1) Straight and 2) bended	Constant	2	×
[Chen 95]	Objects with: 1) straight, 2) S-shape edges.	2D: 1)straight, 2)bended	Constant	3	×
[Zhang 00]	1) A rubbermaid container, 2) a part of a rim of a bucket	2D:1) straight, 2) bended	Constant	4	×
[Li 13a]	Cable (rubber cable on a table)	2D	Constant	4	Possible
New.	Straight, bended, floating metal bars	2D and 3D	Constant and variable	5	✓

Table 4.2: Summary of the experimental scenarios in edge following tasks.

The hierarchical structure of the proposed tactile controller allows switching between the mappings of the tactile features and the Cartesian components of motions. In the following, these tactile features' mappings for edge tracking are used in a set of different experiments listed in Table 4.3 and described in the following sections.

Exp.	Object	Contact type	Features	Inv. Tact. Jac. J^{-1}	Proj. matrix P	Description. Application
1	Bar (rigid)	Partial edge	CoC	$\alpha_1 = 0\ \alpha_2 = 0$	$diag$ $(0,0,1,0,1,0)$	Alignment with a partial edge using the state-of-the-art approach.
2	Bar (rigid)	Partial edge	CoC then ZMP	$\alpha_1 = 0\ \alpha_2 = 0$ then $\alpha_1 = 1$	$diag$ $(0,0,1,0,1,0)$	Alignment with a partial edge using the proposed approach.
3	Bar (rigid)	Edge	Force, orientation, CoC_y, CoC_x	$\alpha_1 = \alpha_2 = 0$	$diag$ $(1,1,1,0,0,1)$	Follow a bar that was in contact and aligned
4	Bar (rigid)	Point then Edge	CoC_x then ZMP_x, force, orientation, CoC_y then CoC_x	$\alpha_1 = 0\ \alpha_2 = 0$ then $\alpha_1 = 1$	$diag$ $(0,0,1,0,1,0)$ then $diag$ $(1,1,1,0,1,1)$	Follow a bar that was not aligned
5	Bended bar (rigid)	Edge	ZMP_x, force, orientation, CoC_y, CoP_x	$\alpha_1 = 1\ \alpha_2 = 0$	$diag$ $(1,1,1,0,1,1)$	Follow a bended bar; bend is in normal direction to the sensing plane
6	Bended & curved bar (rigid)	Edge	ZMP_x, force, orientation, CoC_y, CoP_x	$\alpha_1 = 1\ \alpha_2 = 0$	$diag$ $(1,1,1,0,1,1)$	Follow a bended and curved bar; bends are in normal and planar directions to the sensing plane
7	Floating bar (flexible)	Edge	ZMP_x, force, orientation, CoC_y, CoP_x	$\alpha_1 = 1\ \alpha_2 = 0$	$diag$ $(1,1,1,0,1,1)$	Follow a bar with variable stiffness. One of the ends of a bar is floating

Table 4.3: List of experimental results on real robots. Exploration.

4.3.3 Experiments 1 and 2: increasing the contact area with "partial" edge-contact

In this subsection, we describe the switching between the proposed ZMP feature and the conventional CoC feature in the experiment with a partial edge-contact. In the previous Chapter (Section 3.5.2), the convergence of the ZMP tactile feature during the partial contact was demonstrated and was sufficient to control an estimated moment of force.

In order to explore an object (as for example, a metallic bar), whose location is unknown and the vision information is occluded or not available, a robot can move its end-effector with a tactile array until the appearance of a contact (Fig. 4.3, a). At the presence of a contact (Fig. 4.3, b), the end-effector can rotate (around x- or y- axis) in order to align with the edge of the bar. The Euclidean distance between the center of contact CoC and the center of the sensing array can be mapped to the rotational motions around the x- and y- axes of the end-effector's frame by setting $\alpha_1 = \alpha_2 = 0$ in J^{-1} and $P = diag(001110)$ as described in the following subsection – Alignment by CoC. At the moment when the edge is detected, J^{-1} switches to control a moment of force (the proposed ZMP feature) by setting $\alpha_1 = 1$ without changing P as described in the subsection – Alignment by CoC and ZMP.

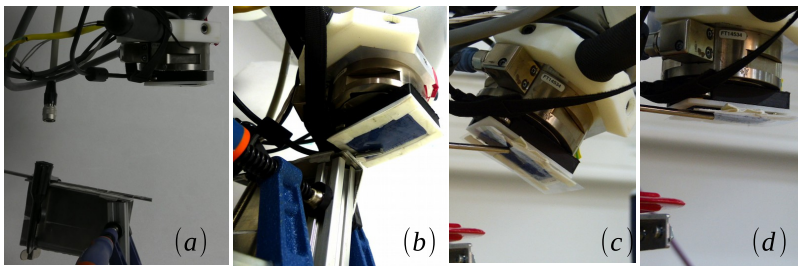


Figure 4.3: Exploration of a metal bar: (a) state "no contact", (b) "point"-type contact, (c) state-of-the-art approach fails to align with an edge that is partially covering the sensor's surface, (d) alignment with the proposed ZMP feature.

Alignment by CoC: previous approach (experiment 1)

The goal of this task is to align the orientation between the bar and the sensing array so that their surfaces are parallel as shown in Fig. 4.3d. When an edge is not covering the sensing array from one side to another (not infinite), the previously proposed approach [Li 13a] to control the rotation around the x- and y- axes of the sensor's frame fails to do

this alignment (Fig. 4.3, c). The error between the center of contact coordinate CoC_x and the center of the sensor does not converge to zero (Fig. 4.4, a). The variable elements of the matrices are equal to $\alpha_1 = 0$ and $P = diag(001010)$ in this experiment. Hence, CoC_x is mapped to the rotational motion around the y-axis of the end-effector, which corresponds to the inverse Jacobian used in the state-of-the-art approach [Li 13a]. So far, $\mathbf{J^{-1}} = \begin{pmatrix} 100000 \\ 010000 \\ 001000 \\ 000010 \\ 000100 \\ 000001 \end{pmatrix}$. In Fig. 4.4a, the first marker X indicates the start of the rotation of the end-effector when the new set point is established $CoC_x = 0$ (i.e. the center of the sensing frame). This rotation continues during phase II in order to align the sensor with the bar by rolling the contact along the bar. However, at some point (second marker), the contact reaches the end of the bar and the end-effector starts rotating around the end of the bar, i.e. the feedback of CoC_x stops converging to the desired values as shown in phase III ($CoC_x = -5mm$). Fig. 4.4, b shows the history of the orientation of the sensor frame: the end-effector keeps rotating around the end of the bar (around y-axis) until it reaches the joint limits.

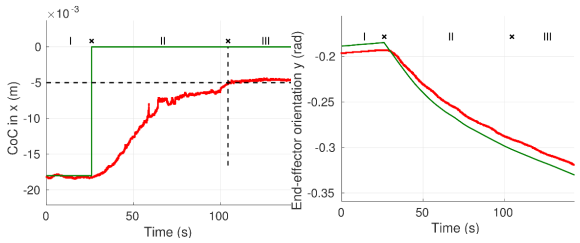


Figure 4.4: Alignment with a partial edge using the state-of-the-art approach [Li 13a]. (a) The evolution of the center of contact in x-axis. The feature error does not converge to the desired value (the center of the sensor) when an edge is in partial contact with the sensor surface. (b) Rotation around y-axis.

Alignment by CoC and ZMP: our approach (experiment 2)

The rotational motion around y-axis can be linked to the error in CoC and then to the error in ZMP_x at the moment of the detection of an edge in order to correct the infinite rotation of the previous approach. For doing this , the variable in the inverse tactile

Jacobian matrix must to be set to $\alpha_1 = 1$. When the robot touches the bar, it will firstly detect a point-contact (Fig. 4.3b) and the rotational error is given by CoC_x by the Jacobian $\mathbf{J^{-1}} = \begin{pmatrix} 100000 \\ 010000 \\ 001000 \\ 000010 \\ 000100 \\ 000001 \end{pmatrix}$. After some rotation, an edge appears in the tactile image (following the algorithm given in the Section 3.4.2) and the rotational error is given by ZMP_x when the Jacobian becomes $\mathbf{J^{-1}} = \begin{pmatrix} 100\ 0\ 00 \\ 010\ 0\ 00 \\ 001\ 0\ 00 \\ 000\ 0\ 10 \\ 100-100 \\ 000\ 0\ 01 \end{pmatrix}$. Fig. 4.5 illustrates the tactile images and features during this transition from the point-contact (Fig. 4.5 a, b) to the edge-contact (Fig. 4.5 c, d).

(a) (b) (c) (d)

Figure 4.5: Tactile images and CoC: (a) at initial state of rotation by CoC_x, (b) moment when the edge is detected, (c) moment of switching to control the ZMP_x feature, and (d) the final state of the contact. Green and red colors represent desired and feedback features.

In Fig. 4.6a, the first marker x indicates the switching time from rotation by CoC_x (phase I) to control the estimated moment of force with the ZMP_x feature (phase II): the type of contact switches from point to edge type. The second marker is the end of the transient state of the control of the estimated moment of force (phase III). In addition to ZMP_x (Fig. 4.6c), the normal force (Fig. 4.6 d) is controlled simultaneously to keep the contact. Thus, this method can align with the partial without infinite rotation as in the previous approach.

4.3.4 Experiment 3: straight bar following, initially in contact and aligned

In order to follow a line during the exploration of an edge of an object (for example, a bar), the external tangential motion is artificially imposed onto the sensor along the edge

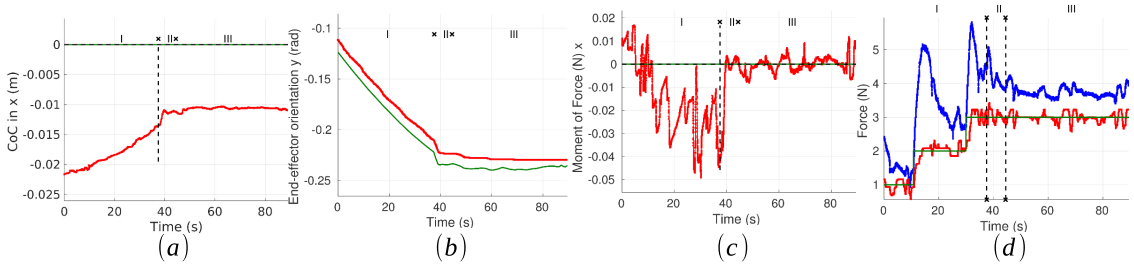


Figure 4.6: Alignment with a partial edge using the proposed approach. (a) The evolution of the center of contact along x-axis. The feature error converges to the desired value by sequentially controlling CoC_x and then ZMP_x. (b) The rotation around y-axis. (c) The ZMP_x feature, and (d) force. The desired, feedback, and observer's values are in green, red, and blue colors, respectively.

by setting $\gamma_x = 1$ in $\boldsymbol{P}$ and $\alpha_1 = 0$ in $\boldsymbol{J^{-1}}$. Thus, the desired value of CoC_x defines the velocity and direction of a sliding motion along the edge. The error of the end-effector's pose during this sliding motion is given by:

$$\boldsymbol{\Delta x} = [(CoC'_x - CoC_x), \Delta CoC_y, \Delta f, -, -, \Delta\alpha_z]^T \tag{4.3}$$

where CoC'_x can be set to any value higher or lower than the actual CoC_x and defines the velocity and direction of the sliding motion with respect to the sensor's frame. The features and the configuration of the controller are shown as the third experiment in Table 4.3.

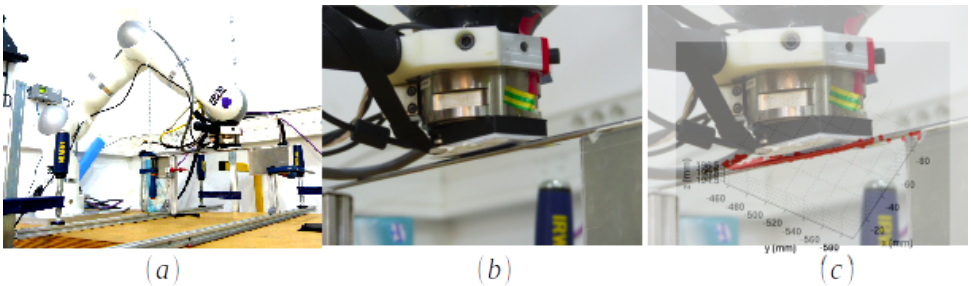


Figure 4.7: (a) The experimental setup for following a straight bar, (b) the KUKA arm is in contact with the metal bar fixed from its both sides, (c) the results of exploration: detected contact points depicted over the real metal bar (depicted with red dots).

For the exploration of a bar (Fig. 4.7, a, b), the motion for following its edge (contact

alignment with the bar) is realized by servoing the force (to keep the contact with the bar), orientation of the edge (the edge is controlled to be parallel with respect to the x axis of the sensor attached frame) and center of contact along y-axis (to keep the contact in the middle of the sensor in y-axis), thus setting $P = diag(111001)$. The moment of force is not controlled during this experiment since the bar is parallel with respect to the x axis of the sensing frame.

The evolutions of the feature errors, including CoC_x, CoC_y, force, orientation, are given in Fig. 4.8 a - d, respectively. During phase I the sensing surface is in contact with the bar. The actual CoC_x is at the center of the sensing frame, since the bar is infinite in the frame and parallel to x-axis. The exploration begins at phase II; the velocity of the exploration depends on the error $\Delta = CoC'_x - CoC_x$, where $CoC'_x = -20mm$ (for the opposite direction, the desired $CoC'_x = +20mm$). The result of the exploration (registered at the end of phase III) is depicted in Fig. 4.8e and in Fig. 4.7c: the red dots is a cloud of contact points representing the surface of the bar. A thin and flexible protecting plastic was added to decrease the tangential forces to protect the sensor, since it is not robust to them.

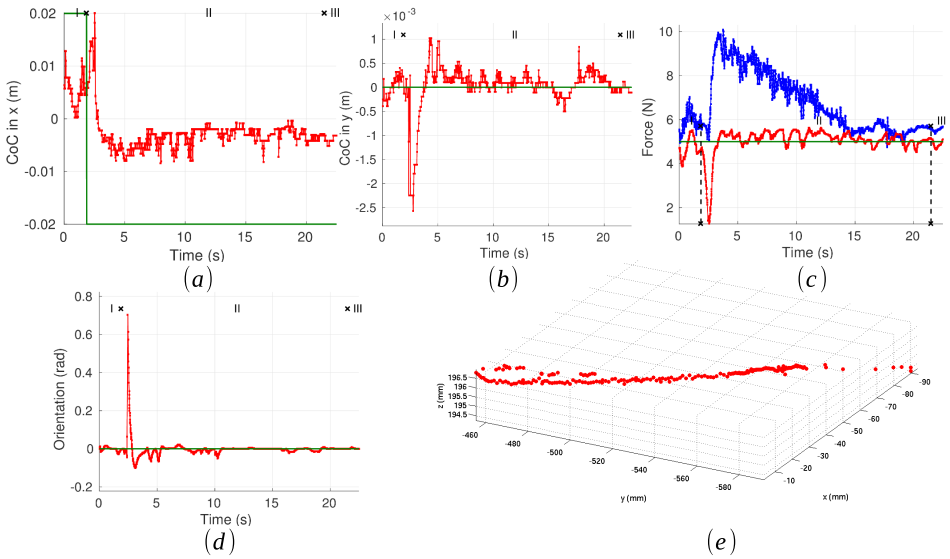


Figure 4.8: (a) Plots of the evolution of CoC_x, (b) CoC_y, (c) force, (d) orientation during the sliding motion over the edge of the straight and aligned metal bar. (e) The results of the exploration are the points of contacts.

4.3.5 Experiment 4: straight bar following, initially not in contact nor aligned

In the previous experiment (Section 4.3.4), the bar was already in contact with the sensor and also aligned with the sensing surface. Similar experiments were shown in the robotics literature several times [Berger 88] [Sikka 94a] [Li 13a]. Since our tactile framework is reconfigurable, it is able to implement a more general exploration strategy[1], where the bar is not in contact nor parallel with respect to the sensor frame at the initial state (Fig. 4.9, a).

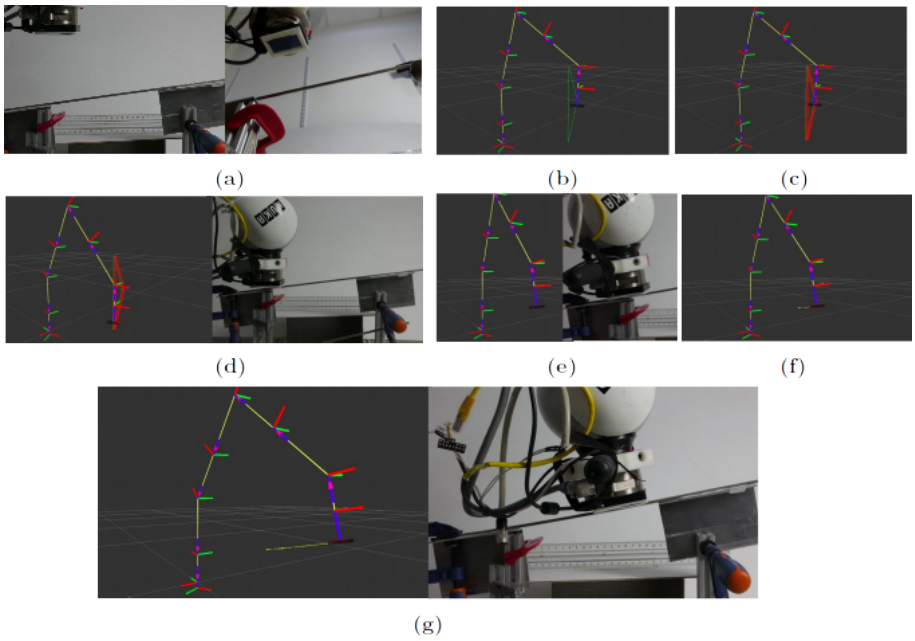


Figure 4.9: Exploring an unknown shape of a metal bar: align with an edge and move along the detected edge. (a) The initial state, (b) trajectory in green line and (c) generated poses in red line that the end-effector follows until (d) the sensor detects a contact, (e) alignment with an edge, (f) following the edge, (g) reconstructed bar at the end of the exploration.

This experiment can be divided into 4 different phases according to the different configurations of $\boldsymbol{P}$ and $\boldsymbol{J}^{-1}$:

- **Phase *I*: free motion until contact with the bar**

[1] https://www.youtube.com/watch?v=kydq-brZnYU&index=9&list=UUQEGPPSY-gQjDm9TGokCDuA

During phase I, the robot follows an arbitrary generated trajectory (Fig. 4.9, b, c) in order to find the contact with the bar. In our work, it is a triangle, starting from the actual pose of the end-effector and moving down in z direction (Fig. 4.10 a) until the contact. Tactile servoing is disabled by $\mathbf{P} = [0]$ while the contact is not detected.

- **Phase II: aligning (increasing the area of contact) with the bar**

  At the instant of contact (Fig. 4.9, d), the sensor frame starts aligning with the bar by rotating around y-axis (Fig. 4.9,e), which is linked to CoC_x similarly to Experiment 1 (Section 4.3.3). It is assumed that the edge lies along the x-axis of the sensor's frame. At this phase II, the desired value of the feature was set to zero (center of the sensor) as shown in Fig. 4.10 b. During the increase of the area of contact via a rotation around y-axis, CoC_x moves towards the center of the sensor until the instant when the point-contact becomes the edge-contact (as in Fig. 4.5).

- **Phase III: switch to edge control**

  When the edge is detected in the tactile image, the robot switches the mapping of the rotational motion around y-axis linked to the error in CoC_x to the error in ZMP_x (similar to Experiment 2) to avoid the infinite rotation of Experiment 1. In addition, CoC_y, α_z, and force features (Fig. 4.10 d, e and g) are controlled from this instant until the end of exploration, similar to Experiment 3. Therefore, $\mathbf{J}^{-1} = \begin{pmatrix} 1 & 00000 \\ 0 & 10000 \\ 0 & 01000 \\ 0 & 00010 \\ -1 & 00100 \\ 0 & 00001 \end{pmatrix}$ and $\mathbf{P} = \begin{pmatrix} 000000 \\ 010000 \\ 001000 \\ 000000 \\ 000010 \\ 000001 \end{pmatrix}$. This phase is active as long as the orientation of the edge in the tactile image is not perfectly aligned with x-axis (the errors between the desired and feedback features: α'_z and α_z, ZMP'_x and ZMP_x, CoC'_y and CoC_y are all zero).

- **Phase IV: exploration**

  After the complete alignment of the bar with the sensing surface in phase III, the robot starts the exploration driven by the error in CoC_x (Fig. 4.10 b), linked to the translational motion along the bar as in Experiment 3. Variable elements in

the projection matrix are set as follows: $\mathbf{P} = \begin{pmatrix} 100000 \\ 010000 \\ 001000 \\ 000000 \\ 000010 \\ 000001 \end{pmatrix}$. Thereby, the interaction matrix activates the translational motion along the x axis of the sensing frame in order to start the sliding motion along the bar (Fig. 4.10f, evolution of the end-effector position) while simultaneously control the moment of force (Fig. 4.10c), force (Fig. 4.10g), the orientation (Fig. 4.10e), and CoC_y (Fig. 4.10d). The tactile Jacobian is the same as in phase *III*. As a result, we recorded the cloud of contact points (the yellow pixels in Fig. 4.9g) at the end of the exploration.

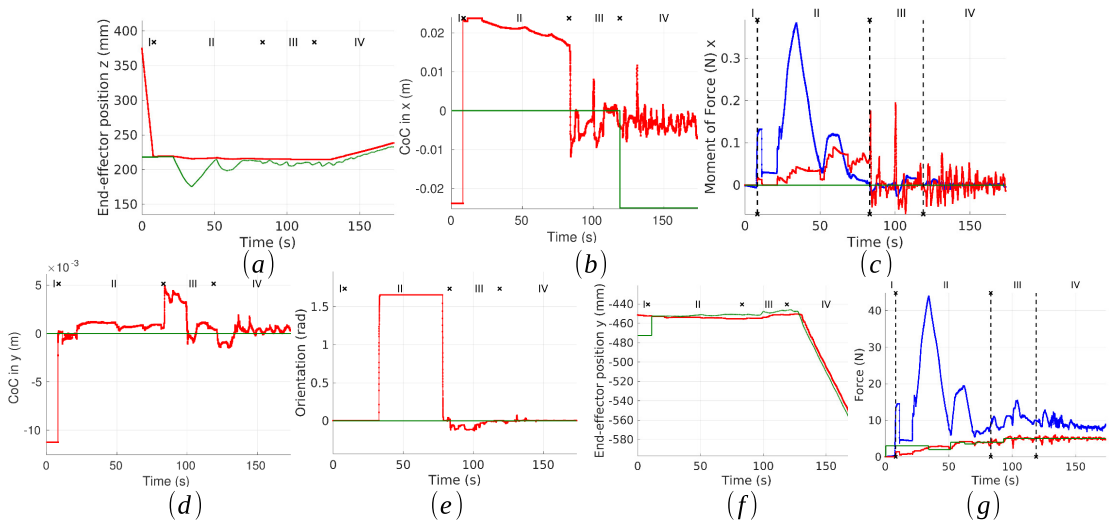


Figure 4.10: Plots of the evolution of (a) the position of the end-effector in z-axis, the evolution of (b) CoC_x during the alignment with an edge, (c) the moment of force around y-axis when an edge is detected, the history of (d) CoC_y, (e) orientation, (f) end-effector's y-coordinate, and (g) the force during the edge following task.

4.3.6 Experiment 5: exploration of a bended bar

In the previous experiments, the experimental setups included straight bars. However, in the real world, they can be bended. With the proposed ZMP features, the rotations of the robot end-effector around x- and y- axes (i.e. on the sensing surface) are controlled. Hence, the robot is able to tactile servo a bar that is bended in the plane normal to the sensing surface (Fig. 4.11) and the radius of curvature of this bar can be later estimated.

The features and the configuration of the controller are shown as the fifth experiment in Table 4.3. At this time, the tactile projection matrix is: $\boldsymbol{P} = diag\ (1,1,1,0,1,1)$. The inverse tactile Jacobian matrix's variable elements are: $\alpha_1 = 1$ to compute ZMP_x and $\alpha_2 = 0$.

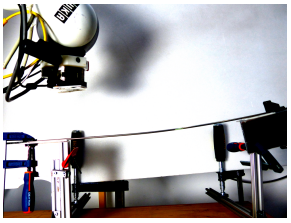


Figure 4.11: The setup for the exploration of a metal bar bended towards the sensing surface.

Fig. 4.12a-g illustrate the evolution of the position of the end-effector in z-axis, the history of CoP_x, the moment of force around y-axis, the history of CoC_y, α_z, force, and the detected points of contact during the edge following task with the bended bar. Since the bar is bended upwards in the planar surface, that is perpendicularly to the sensing surface, the z-coordinate of the end-effector increases (Fig. 4.12a). The recorded point cloud illustrates this bend (Fig. 4.12g). At time $t = 18s$, there are deviations of the actual values of CoC_y and orientation (Fig. 4.12d and e, respectively), because the sensor lost the contact with the bar for a moment at the middle of the sensing frame. This can also be observed by the force values that decreased at the same time (Fig. 4.12f). The source of the noise of the ZMP feature values (Fig. 4.12c) is the sensor's imprecisions that have been discussed in the Section E.2. The radius of curvature of the bended bar can be estimated from the history of the points of contact by projecting the coordinates into the plane in which the bar is bended and then by matching a circle on the detected curve (Fig. 4.12, h).

We also compared the baseline approach [Li 13a] with our approach that adds ZMP. Fig. 4.13 a - c illustrate this comparison by showing the moment of force, ZMP_x, and force during the same experiment of following the bended bar without ZMP feature – baseline approach– (blue color) and with ZMP feature – our approach – (red color). Without the estimate of the moment of force (i.e. ZMP), the edge exploration task of a bended bar

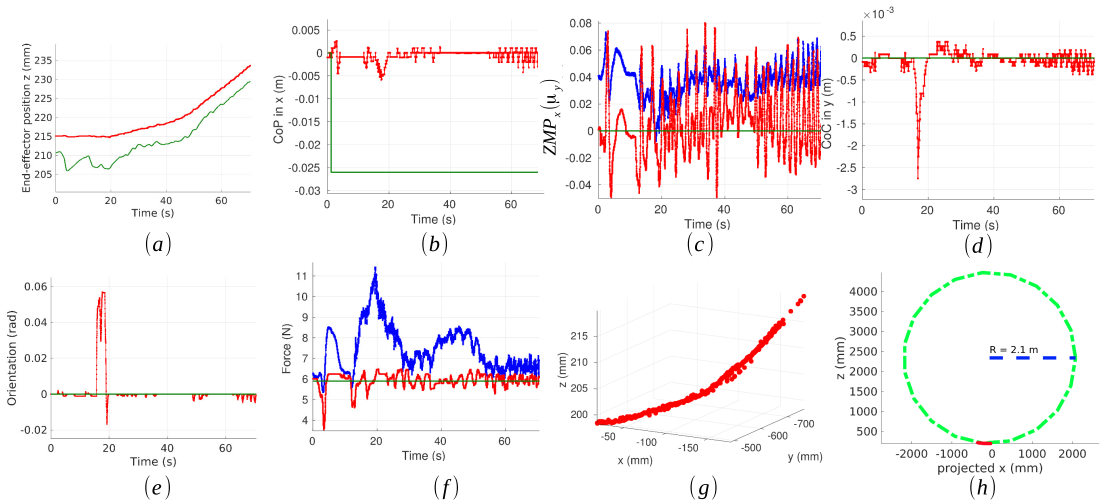


Figure 4.12: Plots of the evolution of (a) the position of the end-effector along z-axis (the assumed (eq. (3.22)) and real (at 1 Hz) values are in green and red, respectively), (b) the evolution of CoP_x, (c) the moment of force around y-axis, (d) the history of CoC_y, (e) orientation, (f) the force (the desired, feedback, and ground-truth observer's values are in green, red, and blue colors, respectively), (g) the history of the points of contact during the exploration, and (h) the reconstruction of the bended bar by fitting a radius of its curvature.

fails: the end-effector did not rotate with the inclination of the bar, which causes the abnormal increase of force (50 N) and moments of force (blue lines in Fig. 4.13 a and c).

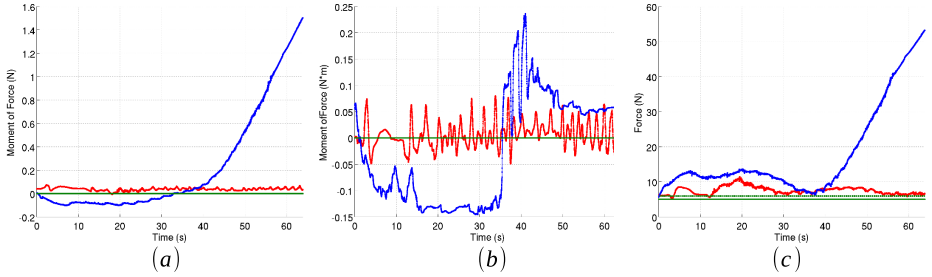


Figure 4.13: The comparison of (a) the moment of force, (b) ZMP_x, and (c) force during the edge servoing without (blue color) and with (red color) use of ZMP feature. The desired values are in green color. The moment of force and the force are measured with the ATi Gamma force and torque sensor.

4.3.7 Experiment 6: exploration of a bar bended in two planes

In the previous experiment, the bar was bended in xz plane only (i.e. towards the sensing surface). In this experiment, the bar is bended in two planes, including xy and xz planes

(Fig. 4.14). Both tactile inverse Jacobian and projection matrices are the same as in the above experiment.

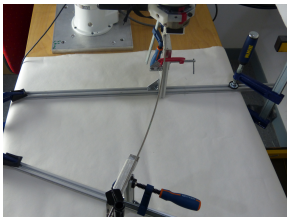


Figure 4.14: The setup for the exploration of a bar bended upwards (plane xz) and curved in the plane parallel to the sensing surface (plane xy).

Fig. 4.15 illustrates the evolution of the position of the end-effector, the tactile feature errors, and the resulting points of contact during the exploration of the bended and curved bar.

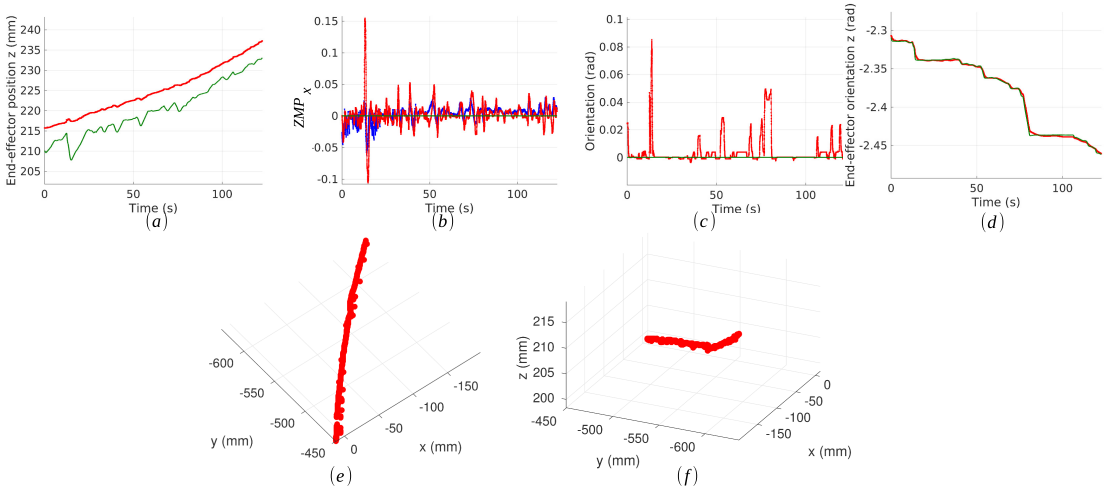


Figure 4.15: Plots of the evolution of the position of (a) the end-effector's position along z-axis; (b) history ZMP_x, (c) orientation of the edge around z-axis, (d) evolution of the orientation of the end-effector around z-axis, (e) cloud of the points of contact from the top view and (f) side view during the edge following task over a bended and curved bar.

Results validate that the tactile servoing controller can be used in applications of environment exploration by following edges with complex shapes, that could not be performed with the previous approaches.

4.3.8 Experiment 7: following a floating bar

The environment, in which a robot operates, can be variant. As for example, a metal bar that is fixed from one side only (Fig. 4.16a). Thereby, when the robot end-effector applies forces during the exploration, this bar can be more bended (Fig. 4.16b). The end-effector can follow this bended bar as long as the radius of the curvature of the bended bar does not decrease so much that the area of contact decreases and the edge in the tactile image becomes a point (Fig. 4.16c).

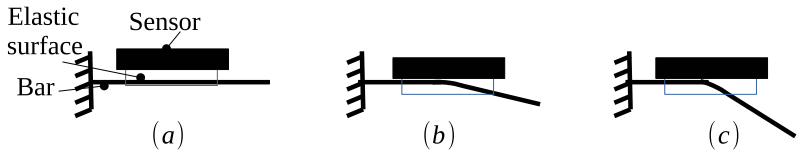


Figure 4.16: Assumption on the radius of curvature during sliding motions along floating and bended bars: (a) the bar is not bended, the bar is bended and (b) the area of contact is kept due to the compliance of the sensing surface, (c) the bar is bended so that the contact area decreases too much and the edge in the corresponding tactile image transforms to a point-contact.

The features and the configuration of the controller are shown as the seventh experiment in Table 4.3, which are the same as in the previous two experiments. Although the stiffness of the bars in the previous experiments was constant, the robot should be able to adapt to a variant environment and explore objects that have non-constant stiffness. With the proposed contact frame control, the robot's end-effector is able to adapt to floating bars with variable stiffness (their stiffness decreases with the distance from their fixed end). Fig. 4.17a shows the experimental setup for the edge following task with the bar, whose one of the ends is not fixed (blue ball). Since the second end is not fixed, the bar deforms under the load applied at remote locations from the fixed end (Fig. 4.17, b).

During the exploration of the flexible bar[2], the end-effector traveled along the bar (300 mm at a constant velocity $v = 300mm/110s = 2.7mm/s$). The farther the end-effector from the initial location the more the bar is bended towards the ground, and, therefore, the position of the end-effector slips down along z-axis (Fig. 4.18 a) of the base attached frame (frame $\{g\}$ in Fig. 4.17b). The histories of the five controlled features, including CoP_x, ZMP_x, CoC_y, orientation of the edge around z axis of the sensor frame, and force

[2]https://www.youtube.com/watch?v=-0JCF1vt5A8&list=UUQEGPPSY-gQjDm9TGokCDuA&index=5

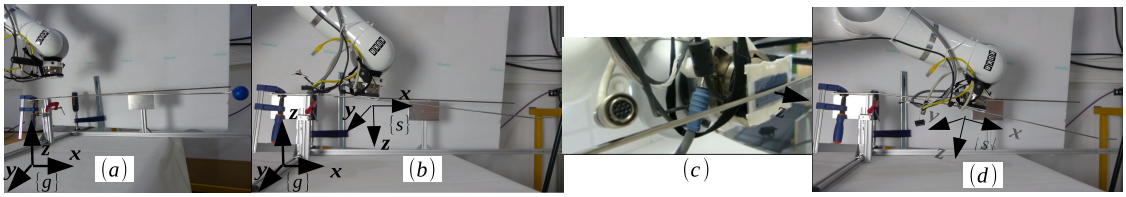


Figure 4.17: The setup for exploration of a bar by the line following task: the metal bar is floating from one of the ends. The stiffness is changing. (a) Initial setup, (b) the sensor frame and the global frame while initial bending, (c) the contact between the sensing frame and the bar, (d) and the rotation of the sensing frame due to the bend of the bar.

are shown in Fig. 4.18 b, c, d, e, f, respectively. The progress of the end-effector along the bar is driven by the error between the desired CoP'_x (green line) and the actual CoP_x (red line). The desired ZMP'_x is zero, and the feedback ZMP_x varies as the bar bends. When the bar bends, the moment of force applied to the sensing surface changes, which is compensated by making the ZMP_x feature error converge to zero. As in the previous experiments, force and CoC_y are controlled to keep the contact with the bar in the middle of the sensing surface.

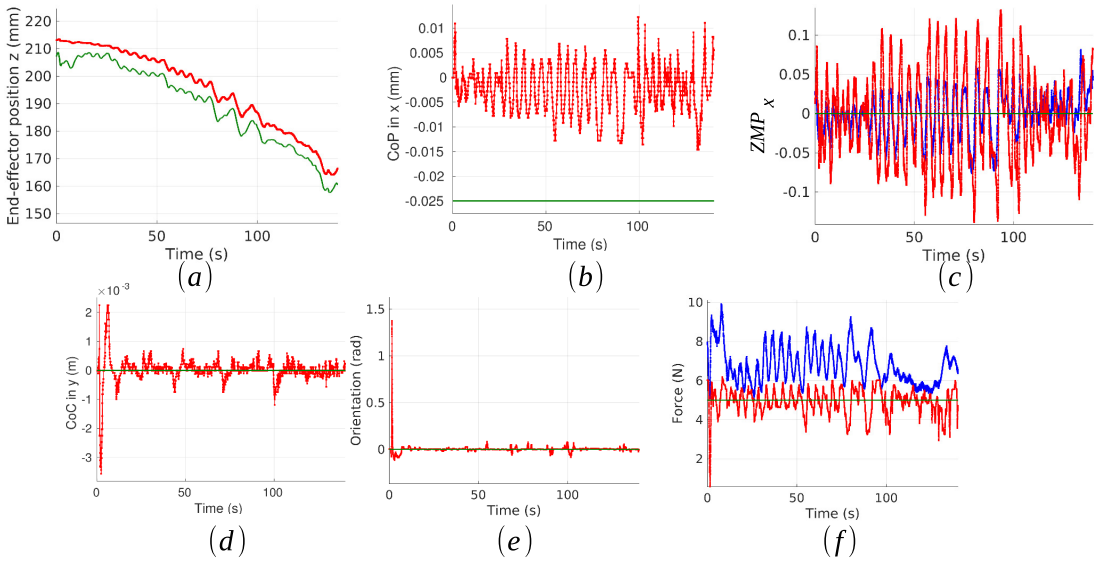


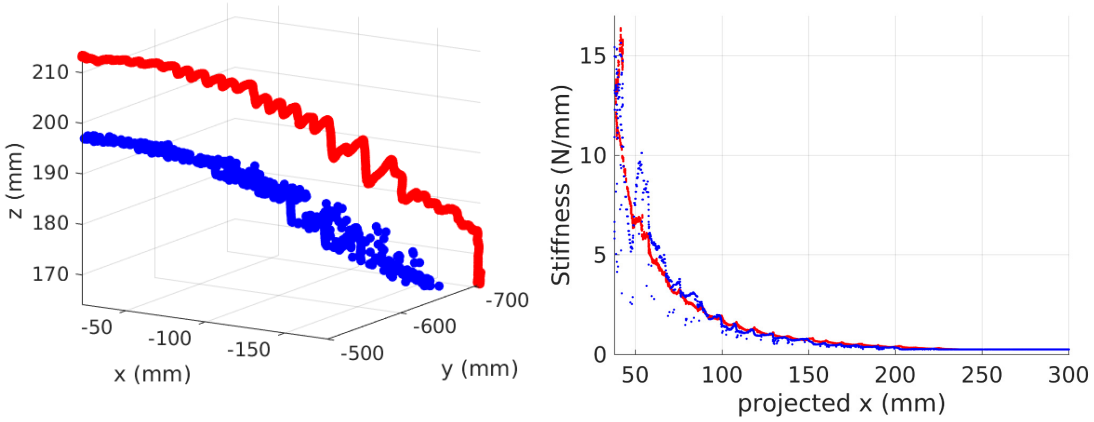
Figure 4.18: Plots of the evolution of the position of the end-effector in z-axis (a), the evolution of CoC_x (b), the moment of force around y-axis (c), the history of CoC_y (d), orientation (e), and force (f) during the edge following task with a floating bar.

As a result of this exploration, the stiffness of the bar at distance l_t corresponding to

the time t from the beginning of the exploration can be calculated:

$$k_{bar}(t) = f(t)/(z(0) - z(t)) \tag{4.4}$$

where $z(0)$ and $z(t)$ are the z-coordinates of the end-effector (contact point) at the beginning and at the time t of the exploration, and $f(t)$ is the force at the time t. In our experiment, $z(t)$ can be given by the real positions of the end-effector (acquired at 1 Hz) or the assumed positions (eq. (3.22)). Fig. 4.19a shows the history of real end-effector position along z-axis (red) and the estimated points of contacts (blue). From them, we can plot the stiffness values along the bar $k_{bar}(t)$ versus l_t (Fig. 4.19 b). The stiffness decreases down to $k_{bar} = 150N/m$. In the same way, we could use this exploration procedure to compute the stiffness of any flexible object similar to the metal bar.



(a) (b)

Figure 4.19: Estimation of the stiffness of the bar along its length. (a) Estimated locations of the points of contact (in red) and history of the end-effector's real poses (blue). (b) Plot of the length l_t of the bar versus its estimated stiffness $k_{bar}(t)$ at each point of contact (red) and real stiffness of the bar calculated from the force (blue) measured with the ATI Gamma force sensor and the displacement of the end-effector in z-axis.

4.4 Deformable object recognition

4.4.1 Review in tactile based deformable object recognition

The experimental scenarios for deformable object recognition considered in the thesis include elastic objects that can be squeezed (fruits, toys, balls, etc.). Other non-elastic objects, as for example ropes, papers, clothes, etc., are not considered. Thus, the objects can be squeezed in order to recognize them based on their tactile responses. Fig. 4.20a outlines the most common steps used in [Schneider 09, Liu 12a, Göger 09, Pezzementi 11, Drimus 14, Liu 12c] in tactile-based object recognition. These steps include: preprocessing, feature extraction, and classification. For the preprocessing steps, the following operations are applied: spatial filtering, thresholding, and normalization of sensor output values to the highest one. The feature extraction is realized using the image moment analysis, PCA and Fast Fourier Transforms (FFT). Finally, the classification step includes neural networks, regression models, etc. in order to obtain the type of object according to its tactile data.

As tactile contact images change with the time when a robot squeezes a deformable object, features can be extracted from series of images from a tactile array. Drimus et al. [Drimus 14] proposed to use the Dynamic Time Warping algorithm and the k-NN classifier to distinguish a spoilt fruit from a fresh fruit by squeezing them with a two-fingered gripper equipped with a 8×8 sensing array (Fig. 4.20b). They applied the algorithm to sequences of their tactile features represented by an average of a tactile image. In contrast to this approach, we propose to use absolute force measurements and tactile images to classify deformable and rigid objects. This method will be described in the following.

4.4.2 General description of the task

Deformable objects can be classified based on their stiffnesses. By activating the force feature in the tactile servoing controller, a tactile array can press the deformable object with desired forces, for example, square-wave signals. Then the response of the object can be recorded and classified based on the euclidean distances provided by the Dynamic

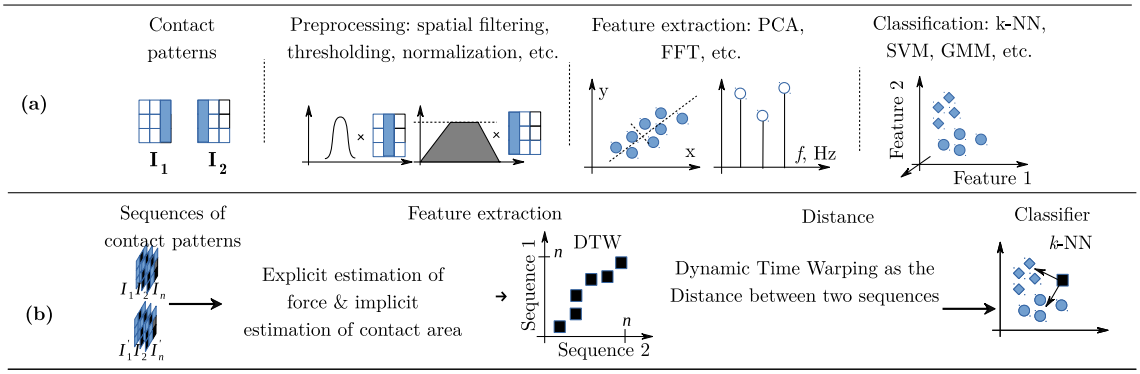


Figure 4.20: Object recognition with tactile feedback. Computational techniques applied in tactile-based object recognition for (a) any objects [Schneider 09, Liu 12a, Göger 09, Pezzementi 11, Liu 12c] and (b) for deformable objects [Drimus 14].

Time Warping (DTW), as implemented in [Drimus 14]. The DTW algorithm is used for measuring the similarity between time series by minimizing the effects of distortion and shifts in time. It can be used to detect similarity between signals with different phases. Given two time series, $\boldsymbol{a} = a_1, a_2, ..., a_N$ and $\boldsymbol{b} = b_1, b_2, ..., b_N$, the DTW algorithm returns the distance, $d = DTW(\boldsymbol{a}, \boldsymbol{b})$, between the two time series, where $d \in R$, $d > 0$ and being closer to inf for completely different time series and zero for equal series. In this thesis, the area of contact (estimated force from a tactile array) and real force (measured by a ground-truth force sensor) served as inputs to the DTW algorithm.

In fact, the sequences of these tactile features, which are the input of the DTW algorithm, are obtained from the tactile controller responses. The next section aims to compare these controller responses over rigid and soft objects in order to show the efficacy of the proposed controller in deformable object recognition tasks. The proposed experiments for benchmarking these tasks are listed in Table 4.4.

Exp.	Object	Contact type	Features	Property	Description
1	Ball (soft)	Point	Force	Polypropylene	The applied forces vary in a lower range (1.5 N - 3 N)
2	Ball (soft)	Point	Force	Polypropylene	The applied forces vary in a higher range (1.5 N - 5 N). The sequences of force responses are compared with those of Experiment 1.

3	Ball (rigid)	Point	Force	Ceramic	The sequences of force responses applied to a rigid ball are compared with those that were applied to a flexible ball.
4	Foam (soft)	Edge	Force	Polyethylene	The sequences of force responses applied to a soft foam are compared with those that were applied to a flexible ball.

Table 4.4: List of experimental results with real robots comparing the controller responses for soft and rigid objects.

Three objects were chosen for the experiments: flexible and rigid balls and a flexible foam (Fig. 4.21, a, c). The KUKA robot was pushing the object against a stiff tabletop (Fig. 4.21, b, d). The flexible ball had a hollow volume inside and it is made of Polypropylene. The rigid ball was very stiff (similar to concrete) and was made of ceramic. Both balls are commonly used as toys. The third object was a foam.

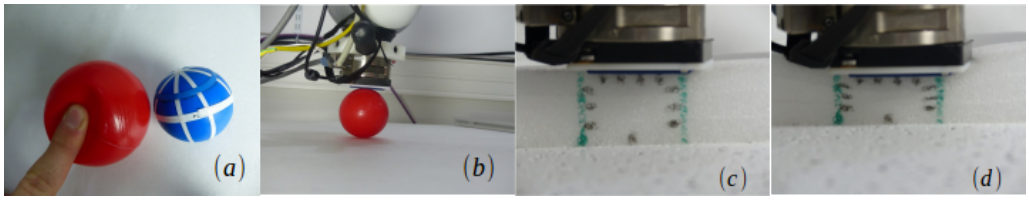
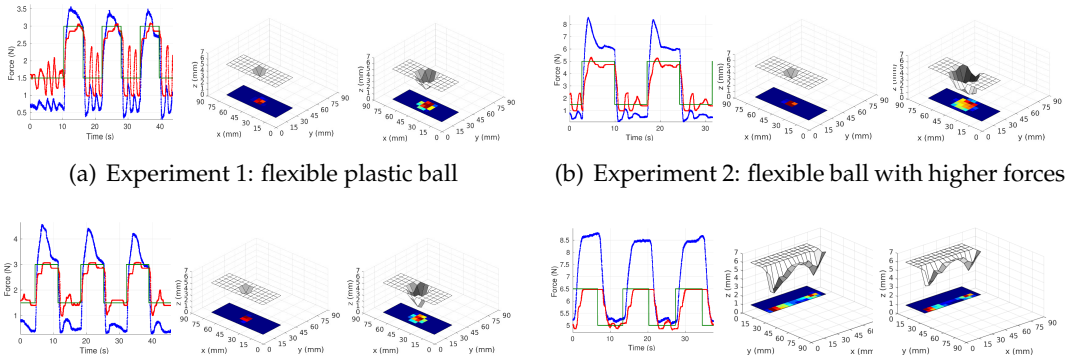


Figure 4.21: The flexible and rigid objects: (a) flexible ball (red ball) and rigid ball (blue ball with white strikes), (b) configuration of the robot and locations of the objects during force control, (c) flexible foam, (d) compressed foam.

4.4.3 Experiments 1-4: tactile responses of soft and rigid objects

Figures 4.22 show the force responses to a square-wave function (left-hand side), deformations of the sensing surface at the minimum (middle) and maximum (right-hand side) applied forces to the soft ball with lower forces (Experiment 1, subfigure 4.22a), soft ball with higher forces (Experiment 2, subfigure 4.22b), rigid ball (Experiment 3, subfigure 4.22c), and flexible foam (Experiment 4, subfigure 4.22d), respectively. The tactile

controller's responses for the soft ball have oscillations at the force level of ∼1.5 N and no oscillations at the level of ∼3 N. This is due to the fact, that at the lower force range, the sensing surface is slightly touching the ball so that the number of tactels in contact varies.



(a) Experiment 1: flexible plastic ball

(b) Experiment 2: flexible ball with higher forces

(c) Experiment 3: rigid plastic ball

(d) Experiment 4: soft foam

Figure 4.22: Within each subfigure for each experiment: the force responses to the square-wave function (left-hand side); the desired, feedback, and observer's value are in green, red, and blue colors, respectively. The deformations of the sensing surface at minimum (middle) and maximum applied force (right-hand side).

The experiments with the flexible plastic ball revealed that at different force ranges the responses have their own particular behavior (Fig. 4.23): after a higher level of applied normal force (∼5 N instead 3 N), there are overshoots in force responses (the estimated from the contact area [Fig. 4.23a] and the ground-truth [Fig. 4.23b] values). Since the flexible ball has a hollow volume, its stress-strain curve may have a step (discontinuous) form that results in these overshoots of the force response at higher ranges of applied forces. The presence of these overshoots means that they can be used to distinguish objects that are deformable but have different internal structures and materials. Fig. 4.24 illustrates the plots of the force responses for the flexible foam and flexible plastic ball. For distinguishing the two different objects, the Dynamic Time Warping [Drimus 14] can be applied to the history of the tactile images (Fig. 4.25) at different force ranges.

Fig. 4.27 shows the results of DTW for two pairs of time series of the estimated force and the ground-truth force for the rigid ball and the soft ball, and for the soft foam and the soft ball: original signals (top) and results of DTW (bottom). The Euclidean distance

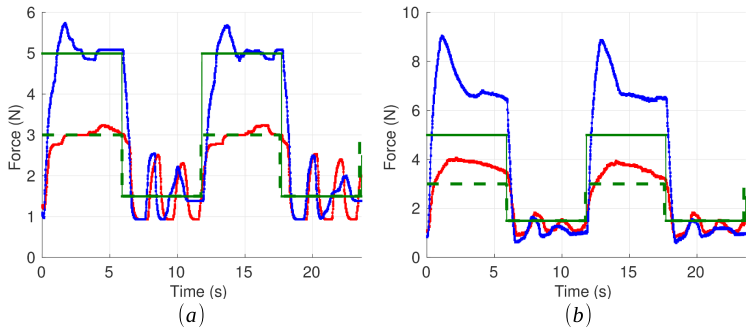


Figure 4.23: Comparison of the force responses for the flexible ball at low and high applied force ranges: (a) the forces estimated with the tactile sensing array, (b) the observer's (ATi gamma force) measurements. The desired values, the responses at a lower and at a higher amplitudes are depicted in green, red, and blue colors, respectively.

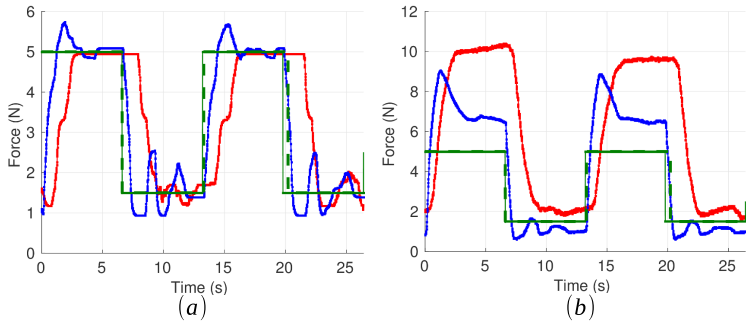


Figure 4.24: Comparison of the force responses for the soft foam and flexible plastic ball: (a) forces estimated with the tactile sensing array, (b) the observer's (ATi gamma force) measurements. The desired values, the responses of the soft foam and of the flexible ball are depicted in green, red, and blue colors, respectively.

between estimated forces (area of contact or contact pattern) for the rigid and soft balls is 495, whereas the same parameter between the soft foam and flexible ball is 822. In addition, the Euclidean distances between the ground-truth forces for the same pairs of objects are 1632 and 1088, respectively, as are summarized in Table 4.5. For humans, as was shown by Hayward et al. [Robles-De-La-Torre 01], perception of force is more significant than a spatial information about an object. Therefore, the force responses over the foam and the flexible ball should be more similar than the responses over the rigid ball and flexible ball. However, the contact images during the interactions with the soft and the rigid balls are more similar than the contact images during the interactions with

the soft ball and foam.

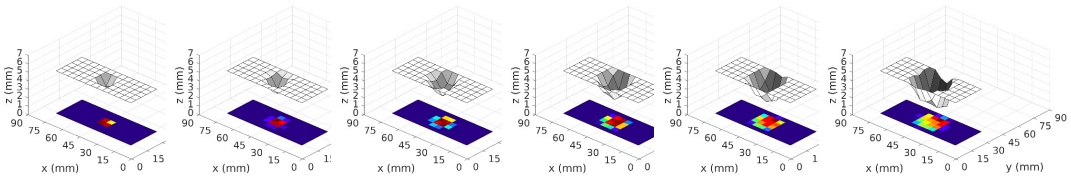


Figure 4.25: History of tactile contact frames while pushing a deformable plastic ball.

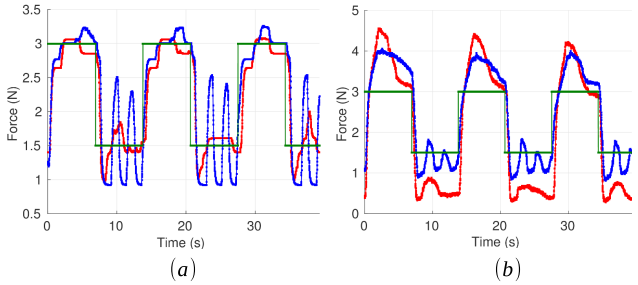


Figure 4.26: Comparison of the force responses for the rigid and flexible balls: (a) forces estimated with the tactile sensing array, (b) the observer's (ATi gamma force) measurements . The desired value, the responses of the soft and of the rigid balls are depicted in green, red, and blue colors, respectively.

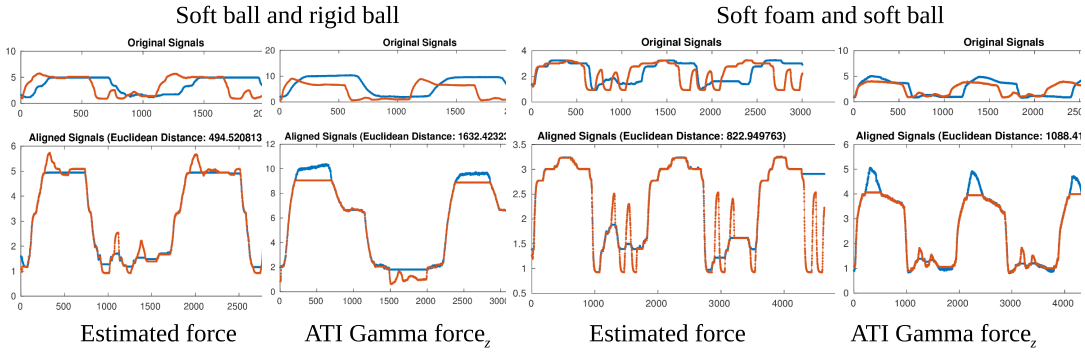


Figure 4.27: Dynamic time warping results for tactile image sequences

Pair of objects	DTW Euclidean distance (estimated force = contact area)	DTW Euclidean distance (observer's force)
Rigid & soft balls	495	1632
Soft foam & soft ball	822	1088

Table 4.5: Euclidean distances between the DTW of two pairs of sequences of tactile images: for the rigid and soft balls and for the soft ball and the soft foam.

4.5 Manipulation by rolling

4.5.1 Review of experimental scenarios in manipulation by rolling

The experimental scenarios described so far did not involve manipulating an object. However, the tactile feedback provided by the tactile sensing arrays allows robots to manipulate objects without visual information. Moreover, an object can be explored by manipulating it.

In [Sikka 94a] and [Chen 97], one tactile array attached onto a robot's end-effector was used in order to roll a rolling pin[3] on a flat surface. In [Bicchi 99] and [Li 15], two tactile arrays attached onto a two-fingered robot were used to roll an unknown object with cylindrical and spherical shapes, respectively. The main parts and attributes of the experimental scenarios of these works are summarized in Table 4.

In our work we repeated the tasks of rolling cylindrical and spherical objects. In addition, we performed rolling a cylindrical object over a rugged terrain (non-flat surface). Finally, we will show (Section 4.5.6) how the shape of an unknown object can be reconstructed by rolling it.

4.5.2 General description of the task

The proposed tactile servoing controller can be used to manipulate objects (by considering the natural and artificial constraints as shown in Section 3.4.4). In the current work,

[3]that is normally used for the dough

Approach	Object	Surface
[Sikka 94a]	Cylindrical object	Flat
[Chen 97]	Rolling pin (for dough).	Flat
[Bicchi 99]	Ball (unknown shape)	Flat
[Li 15]	Cylinder	Flat
New.	Rigid/Soft balls and cylinders, ellipsoids	Flat and rugged terrain

Table 4.6: Summary of the experimental scenarios in manipulation by rolling of spherical, cylindrical and ellipsoid objects.

the manipulation by rolling of objects over a static surface is considered. The tangential contact forces are assumed to be rather high, so that slips do not occur (the normal forces during the manipulations are within the Coulomb friction cone).

The surface or an object may have some irregularities: the terrain of rolling can be rough or the object can have projections, for example, a marker with a cap that has a holder (bump). Then the static plane and the axis of rolling the object are not parallel at the time the end-effector rolls over projections, thus causing an inclination. This inclination can be incorporated within the controller by the ZMP feature.

In the following, the use of tactile features in a number of tasks of rolling different objects is explained with illustrations of final results, the bandwidth of the tactile controller to roll objects is evaluated and a comparison of rolling object tasks with and without the proposed features is detailed. These experiments are listed in Table 4.7.

Exp.	Object	Contact type	Features	Jac. matrix $\mathbf{J}^{-1}$	Task matrix $\mathbf{P}$	Description/Application
1	Cylinder (rigid)	Edge	CoC_x, f	$\alpha_1 = 0$ $\alpha_2 = 0$	$diag$ $(1,0,1,0,0,0)$	Rolling a rigid cylinder. Evaluation of the tactile controller's bandwidth
2	Cylinders (soft & rigid)	Edge (point)	CoC_x, f	$\alpha_1 = 0$ $\alpha_2 = 0$	$diag$ $(1,0,1,0,0,0)$	Rolling soft/rigid cylindrical objects

3	Cylinders	Edge	CoC_x, ZMP_y, f	$\alpha_1 = 0$ $\alpha_2 = 1$	$diag$ $(1,0,1,1,0,0)$	Rolling over rugged terrain. Comparison of force responses
4	Balls	Point	CoC_x, CoC_y, f	$\alpha_1 = 0$ $\alpha_2 = 0$	$diag$ $(1,1,1,0,0,0)$	Rolling soft/rigid spherical objects
5	Ellipse	Edge	CoC_x, f	$\alpha_1 = 0$ $\alpha_2 = 0$	$diag$ $(1,0,1,0,0,0)$	Recognition by manipulation: ellipse and cylinder

Table 4.7: List of experimental results on real robots. Manipulation by rolling.

4.5.3 Experiments 1 and 2: rolling cylindrical objects

When the manipulated object has a cylindrical shape, the angle of rotation of the object and the traveled distance of the center of contact in the tactile image are correlated by the equation of the arc length of a circle given an angle of rotation:

$$l = \Delta\theta\pi R \tag{4.5}$$

where R is the radius of the cylinder, $\Delta\theta$ is the angle of rotation, and the length of the arc $l = \Delta(CoC_x)$. Since $\Delta(CoC_x) = CoC'_x - CoC_x$, the angle of rotation can be artificially constrained. It is assumed that the rolling motion is along x axis of the sensor frame.

Hence, when CoC_x and force features are activated ($\gamma_z = \gamma_x = 1$), the orientation of the object can be controlled by setting the desired CoC'_x, which is related to the angle of rotation by eq. (4.5). Then the contact frame error is given by:

$$\Delta\boldsymbol{x} = [(CoC'_x - CoC_x), -, \Delta f, -, -, -]^T \tag{4.6}$$

Experiment 1: rigid cylinder

Fig. 4.28 shows the setup for Experiment 1 – rolling a rigid cylinder: the cylindrical object with spokes (deepening lines) on the side part is pushed against a static table by the tactile

array attached on the KUKA arm's end-effector. The spokes indicate the angles and serve as a protractor: there is an angle of 15 degrees between two neighbor deepening lines.

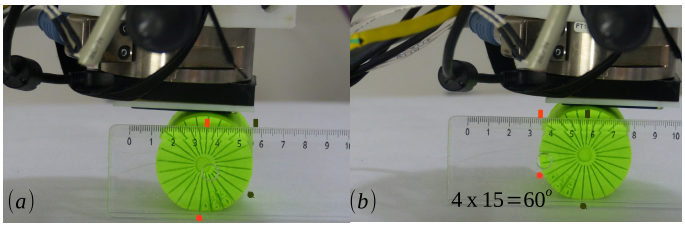
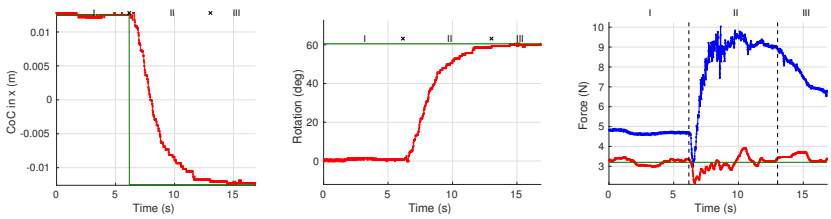


Figure 4.28: Rolling a stiff cylindrical object: (a) before rotation and (b) after. The red and green dots depict current and desired angles of rotation, respectively. The red and green rectangles depict the points of contact with the sensing array that correspond to these angles of rotation.

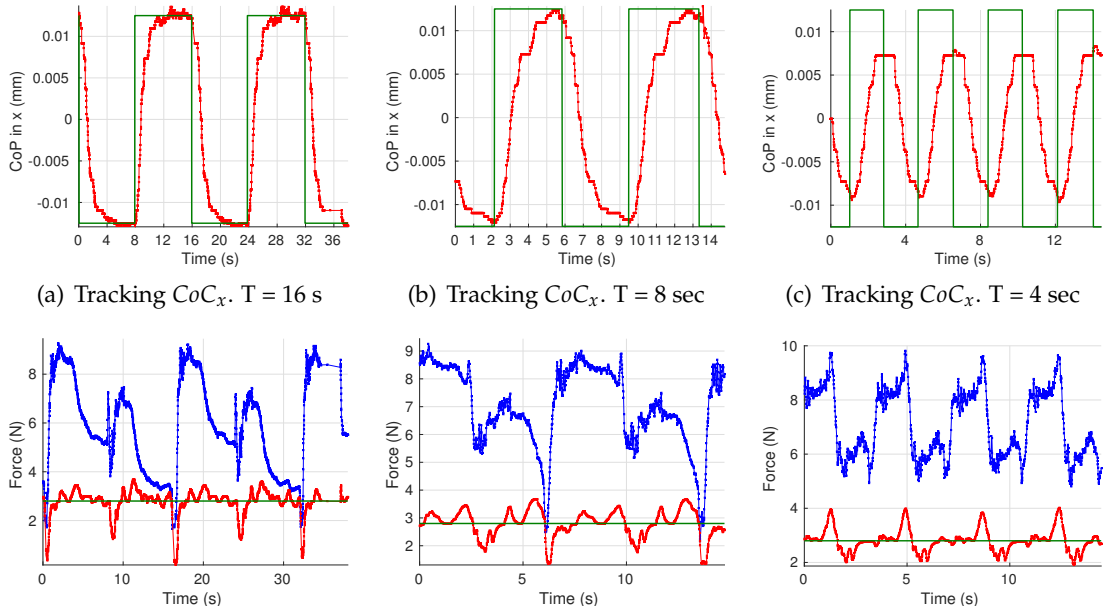
At initial state, the location of the center of contact along x-axis (CoC_x) is ∼12 mm (Fig. 4.29, a). The maximum size of the sensor along x-axis is 50 mm and the origin of the contact sensing frame is at the geometrical center of the sensor. The end-effector travels ∼24 mm (ruler in Fig. 4.28) in order to rotate the object by 60^o (Fig. 4.29, b): the radius of the cylinder is 22.5 mm. In addition, the contact force (estimated by the tactile array) is kept at 3.2 N (Fig. 4.29c).

In industrial applications requiring assembling of a rather high amount of production units or in some specific tasks, as for example preparing a pizza dough, the velocity of manipulation is of importance. This can be evaluated by testing the performance of tracking the position of contact. In order to test this position tracking capability, the robot manipulated the cylinder by tracking square-wave position signals whose amplitude was changing within ±50% of the tactile sensor's length centered at the center of the sensor (Fig. 4.30). The responses of the tactile controller have been evaluated at three different periods (16 sec, 8 sec and 4 sec as shown in Fig. 4.30 a - c). Object motion was recorded with the tactile sensing array and a video camera. At the times when CoC'_x changes its sign, the end-effector changes the direction of the motion, and there are overshoots in the force response (Fig. 4.30d-f). The overshoots can be explained by changes in the area of contact between the object and the elastic surface of the sensor during these transitions. Since the contact force is estimated by the contact area only, the observer's force increases



(a) Evolution of CoC_x (b) Evolution of angle of rotation (c) Evolution of the force

Figure 4.29: Evolution of tactile features while rolling a rigid object. The desired, feedback, observer's values are depicted in green, red, and blue colors, respectively.



(a) Tracking CoC_x. T = 16 s (b) Tracking CoC_x. T = 8 sec (c) Tracking CoC_x. T = 4 sec

(d) Force responses. T = 16 sec (e) Force responses. T = 8 sec (f) Force responses. T = 4 sec

Figure 4.30: Object manipulation capability. The desired, feedback, observer's values are depicted in green, red, and blue colors, respectively.

with the velocity of manipulation, whereas the estimated force (red curve) is controlled at the same level. The position tracking bandwidth during the manipulation by rolling was ∼0.1 Hz (Fig. 4.31). The position tracking bandwidth represents the boundary for the manipulation abilities of the proposed tactile servoing controller.

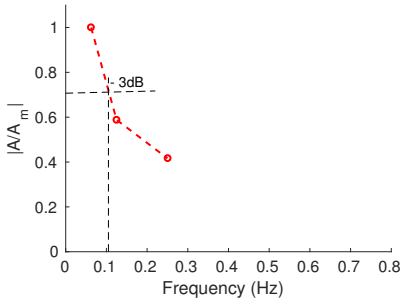


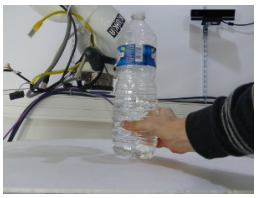
Figure 4.31: Object position gains vs. frequency for $\pm 50\%$ tactile sensor's length about the center point for the KUKA arm.

Experiment 2: rolling rigid and soft cylindrical objects

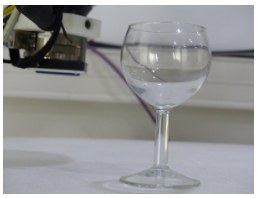
In the above experiment, the manipulated object was 3D printed using ABS plastic. Thus, the surface of the object was without irregularities. In reality, there are many objects that have rough surface and different stiffnesses. In order to validate the performance of the tactile controller in tasks of manipulation by rolling objects with different stiffnesses, the robot was commanded to manipulate a bottle of glue, a bottle of water, a wineglass, and a screw-driver (Fig. 4.32 a - d). The robot end-effector pushed these objects against a stiff table to roll them.

It was assumed that the axis of rotation of the objects was perpendicular to the x axis of the sensor attached frame, because a point-contact (Fig. 4.32 k, l) was detected instead of an edge-contact type for the wineglass and screwdriver due to their shapes. Regardless the softness of the objects, there is a decrease of the contact force at the beginning of the rotation (beginning of phase II in Fig. 4.32 e - h). Phases I, II, III indicate the initial, transient (i.e. rotation), and final states of the manipulations. At the initial state, a new desired set point CoC'_x is fed to the tactile controller and the force is controlled simultaneously as in Experiment 1. At the transient state, the error in CoC converges to zero, and at the final phase the actual CoC_x converges to the desired CoC'_x. The real contact force while rolling the bottle of glue is higher than the one while rolling the bottle of water, which can be explained by the viscosity of the glue inside the bottle of glue (Fig. 4.32 e, f).

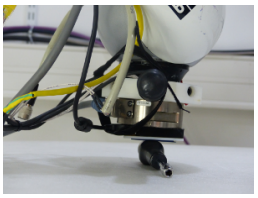
(a) A bottle of glue



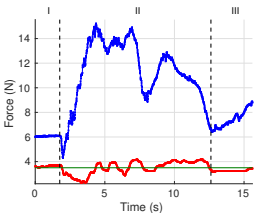
(b) A bottle of water



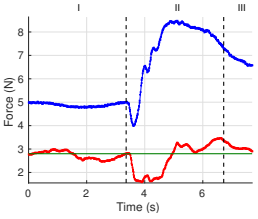
(c) A wineglass



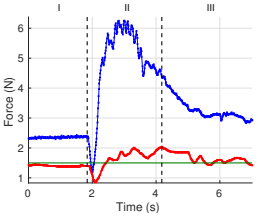
(d) A screw-driver



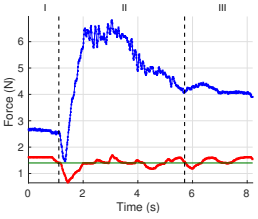
(e) Force response for the bottle of glue



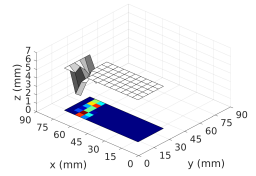
(f) Force response for the bottle of water



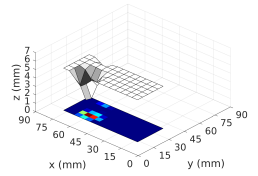
(g) Force response for the wineglass



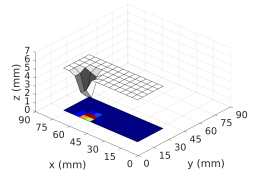
(h) Force response for the screw-driver



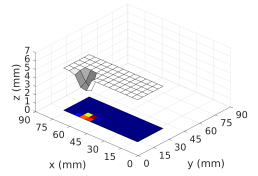
(i) Contact profile for the bottle of glue



(j) Contact profile for the bottle of water



(k) Contact profile for the wineglass



(l) Contact profile for the screw-driver

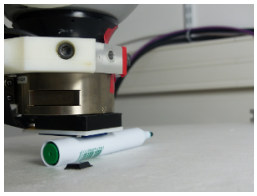
Figure 4.32: Force responses during rolling soft and rigid objects with cylindrical shapes. The desired, feedback, observer's values are depicted in green, red, and blue colors, respectively.

4.5.4 Experiment 3: rolling cylindrical objects with ZMP

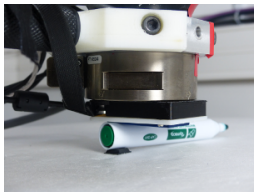
When the surface of the rolling plane is not flat or object has projections (Fig. 4.33 a or c), the moments of force that appear during the rolling over the projections may lead to a damage of the object. This can be avoided by the inclusion of the ZMP feature into the controller, since the ZMP feature serves as an estimation of the moments of force. In addition to the activated variables in the above experiments, ZMP_y is activated by $\gamma_{wx} = 1$ in the projection matrix P, so that contact is evenly maintained across the sensor surface. Thereby, the contact frame error is given by:

$$\Delta \boldsymbol{x} = [(CoC'_x - CoC_x), -, \Delta f, \Delta ZMP_y, -, -]^T \tag{4.7}$$

The comparison in rolling a rigid object (Fig. 4.33b) between the use of the ZMP_y feature ($P = diag\ (1,0,1,1,0,0)$) and without the use of the ZMP_y feature ($P = diag\ (1,0,1,0,0,0)$) is shown in Fig. 4.33e (tactile array) and f (observer). When ZMP_y is incorporated within the controller (red line in Fig. 4.33 e), the end-effector adapts to the new orientation of the axis of rotation of the object[4]. Thus, the applied moment of force decreases (red line in Fig. 4.33 f), whereas it increases (blue line in Fig. 4.33 f) when the feature is not controlled. It is worth noting that the applied moment of force is partially absorbed by the Cartesian impedance of the KUKA arm and the compliant surface of the sensor.

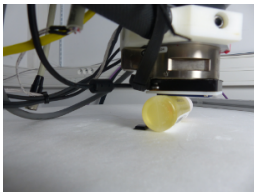


(a) A rugged terrain

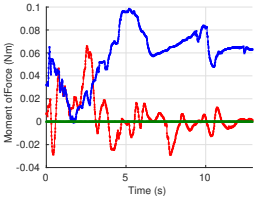


(b) Rolling over a bump

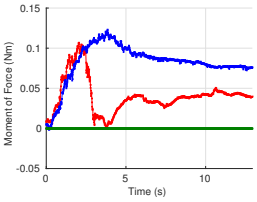
(c) An object with a bump



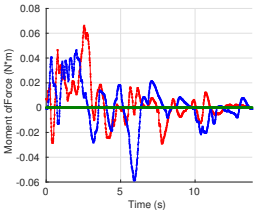
(d) Soft object and a bump



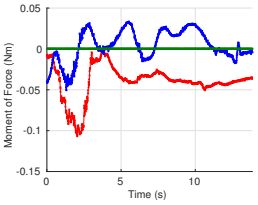
(e) Evolution of ZMP_y while rolling a rigid object over a bump: when ZMP_y is controlled (red color) and without control (in blue) of the ZMP feature



(f) Observer measurements of μ_x while rolling a rigid object over a bump: when ZMP_y is controlled (red color) and without control (in blue) of the ZMP feature



(g) Evolution of ZMP_y while rolling a rigid (in red) and soft (in blue) objects over a bump



(h) Observer measurements of μ_x while rolling a rigid (in red) and soft (in blue) objects over a bump

Figure 4.33: Moment of force responses during rolling a rigid and soft objects with and without control of ZMP_y (estimation of the moment of force). The desired value of ZMP_y is depicted with the green color

If the object to be rolled is deformable, e.g. the bottle of glue (Fig. 4.33 d), the applied moment of force is also absorbed by the object itself; then the applied moment of force decreases with the compliance of the object. Fig. 4.33g shows the controller response for

[4]https://www.youtube.com/watch?v=6h0u3hXufMk

ZMP_y while rolling the rigid marker (in red color) shown in Fig. 4.33b and the soft bottle with glue (in the blue color) shown in Fig. 4.33d. As it was expected, the real applied moment of force measured by the ground-truth ATi Gamma force sensor is smaller for the soft object (blue line in Fig. 4.33 h).

4.5.5 Experiment 4: rolling spherical objects

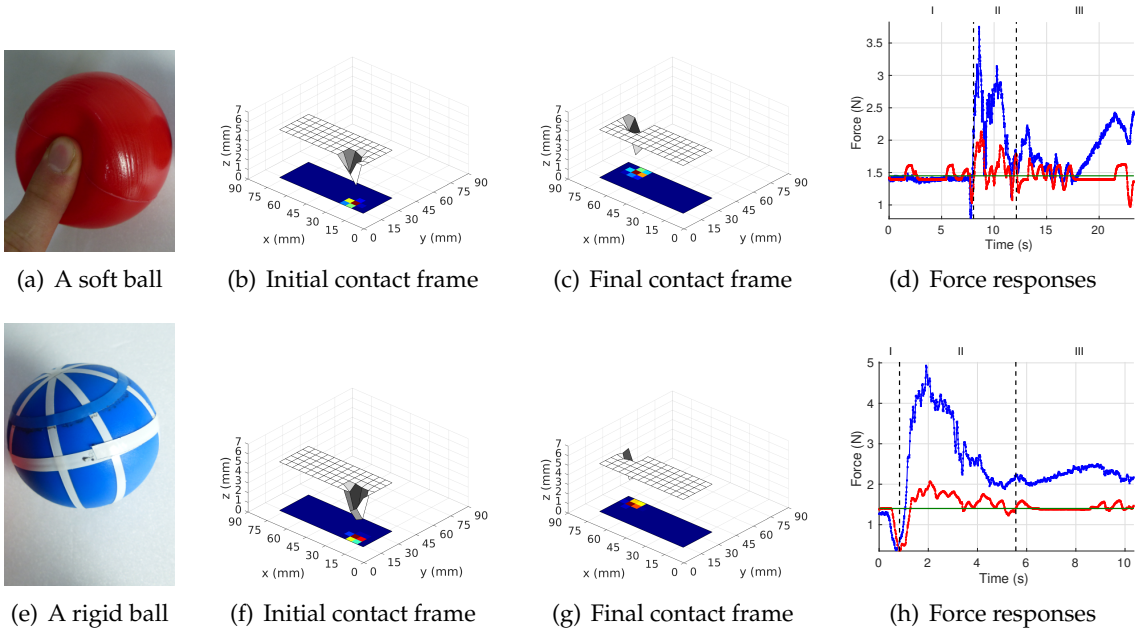
Spherical objects can be rolled in the two directions of a planar surface. Hence, the pose error of the end-effector can also be constrained along y-axis in addition to the x axis of the sensing frame ($\gamma_z = \gamma_x = \gamma_y = 1$). Then, the contact frame error is given by:

$$\Delta \boldsymbol{x} = [(CoC'_x - CoC_x), (CoC'_y - CoC_y), \Delta f, -, -, -]^T \tag{4.8}$$

In the following experiments, the robot was commanded to roll a soft ball (Fig. 4.34 a) and a rigid ball (Fig.4.34 e) from one corner (Fig.4.34 b and f for soft and rigid balls, respectively) to the diagonally opposite corner (Fig.4.34 c and g for soft and rigid balls, respectively) of the sensor's surface. The oscillations in the force responses for the soft ball are higher than for the rigid ball (Fig.4.34 d and h for soft and rigid balls, respectively) due to the hollow structure of the soft plastic ball (Fig. 4.34 a), which causes variations of the area of contact. If there was a viscous liquid inside the soft ball, the force response would have less oscillations as for the bottle of glue in the experiment 3. The phases $I - III$ represent the initial, transient (rolling), and final states of rotation.

4.5.6 Experiment 5: rolling objects with ellipsoid shapes

When the manipulated object has the shape of an ellipse, it can still be rolled as a cylinder (eq. (4.6)) as the force is controlled and, therefore, the end-effector moves along z-axis as well. It is worth noting that the cylindrical and elliptical objects can then be classified based on this force responses and the position of the end-effector along z axis during the manipulation, which is known as recognition by manipulation. The parameters of the ellipse can be computed using the arc length and the difference between the maximum



(a) A soft ball (b) Initial contact frame (c) Final contact frame (d) Force responses

(e) A rigid ball (f) Initial contact frame (g) Final contact frame (h) Force responses

Figure 4.34: Force responses during rolling soft and rigid balls. The desired, feedback, observer's values are depicted in green, red, and blue colors, respectively. The first and second vertical dashed lines depict the time of the beginning and end of roll, respectively.

and minimum coordinates of the end-effector along z axis. The total arc length is given by:

$$C = \pi\big(3(a+b) - \sqrt{10ab + 3(a^2+b^2)}\big) \tag{4.9}$$

where a and b are the major and minor axes of the ellipse, respectively, and C is the total length of the ellipsoid. From the minimum to the maximum position along z-axis, the end-effector travels $\frac{1}{4}$ of the total length of the ellipsoid. Hence, a and b can be derived using the above equation and knowing the difference of the end-effector positions ΔZ and ΔCoC_x that are acquired during the manipulation.

Fig. 4.35 illustrates the process of rolling the object with ellipsoid shape: the contact point coordinate z_c is maximum when the major axis (the longer axis) of the ellipse is perpendicular to the ground (left-hand side of Fig. 4.35a), and z_c is minimum when the minor axis of the ellipse is perpendicular to the ground (the major axis is parallel, middle of Fig. 4.35a). Due to the symmetry of the ellipse with respect to the both major and

minor axes, the length from point 1 to 3 is therefore $\frac{1}{4}C$. The point of contact at any time instant ($t = [t_1 t_2 t_3]$) is represented by the feature CoC_x as the object is assumed to be aligned with the y-axis of the sensor attached frame.

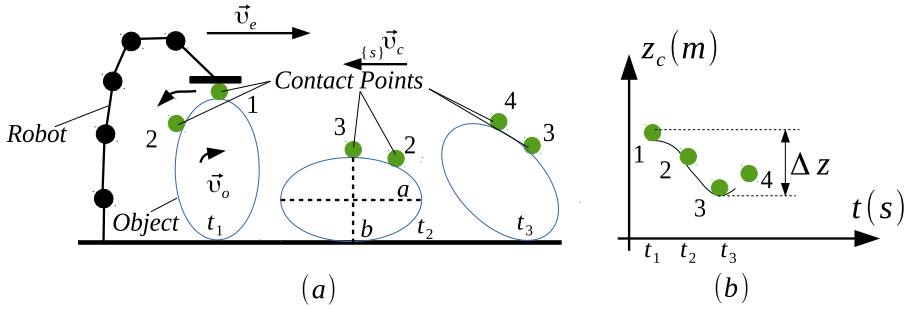


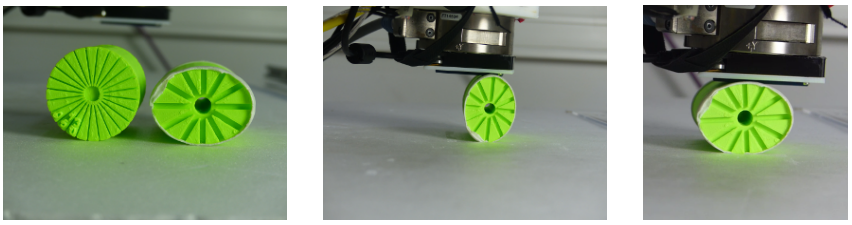
Figure 4.35: Rolling a stiff object with ellipsoid shape: (a) rotation and (b) history of the end-effector position coordinates along z-axis $z(t)$. The object with ellipsoid shape rolls over a flat surface. a and b are the major and minor axes of the ellipse. The contact points with the end-effector move along the surface of the ellipse from 1 to 4. These points move with the velocity ${}_{\{s\}}v_c$ with respect to the sensor attached frame $\{s\}$.

Thus, the length of the minor and major axes can be derived as follows:

$$b = -\big(48\Delta CoC_x + 36\pi\Delta Z\big) + \frac{\sqrt{(48\Delta CoC_x + 36\pi\Delta Z)^2 - 144\pi\big(\Delta CoC_x(24\Delta Z - 14\Delta COC_x) + \Delta Z\pi(2 - 9\Delta Z)\big)}}{36\pi} \tag{4.10}$$

$$a = b + \Delta Z/2 \tag{4.11}$$

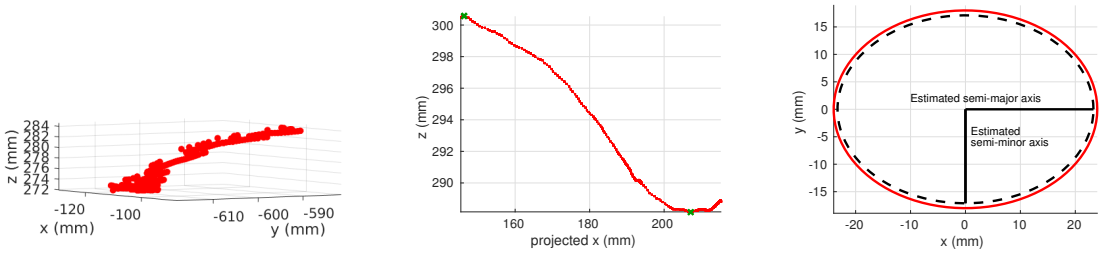
One of the advantages of the tactile servoing is that it can be used in tasks of object recognition by manipulation. This becomes important when the shape of an object cannot be recognized from a single touch, i.e. from one contact pattern. For example, two objects, one of which is cylindrical and the second is elliptical (Fig. 4.36 a), can be distinguished by rolling them. The results of rolling the cylinder are shown in the experiment 1. The object with ellipsoid shape is rolled using the tactile servo controller at the same configuration, in which the error in CoC_x is mapped to the translational motion of the end-effector. Fig. 4.36b and c show the position/orientation of the object before and after rotation, respectively.



(a) Cylinder and ellipse (b) Ellipse before rotation (c) Rotated

Figure 4.36: Rolling an object with ellipsoid shape.

The shape of the object can be reconstructed from the history of points of contacts (Fig. 4.37a) by computation of the semi-minor and semi-major axes of an ellipse using eq. 4.10. Nevertheless, since this cloud of contact points is noisy, we will use the position of the end-effector along z-axis (given by the robot's forward kinematics) to estimate the parameters of the ellipse (Fig. 4.38b). The coordinates along x axis are projected to the axis of motion. The green crosses in the picture depict the minimum and maximum z-coordinate, which are, in fact, feature points. Using the euclidean distances along the both axes of coordinates between these feature points, the ellipsoid was reconstructed according to the (4.10) and (4.11) with accuracy of 98% (Fig. 4.37 c). These feature points can be found by analyzing the history of the positions of the end-effector along z axis of the base attached frame and the x- (or y-) axis of the base attached frame projected to the axis of the direction of motion of the end-effector in xy plane.

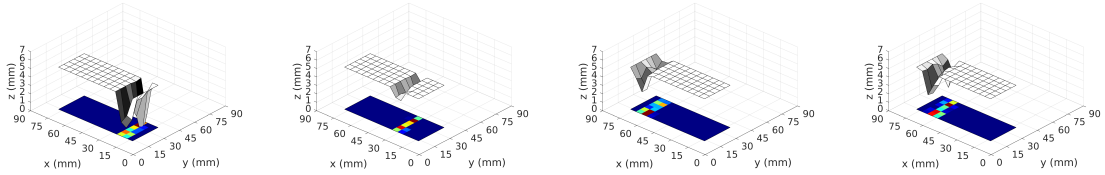


(a) Cloud of points of contact (b) Contacts vs. axis of motion (c) Reconstructed (dashed line) and real (red line) ellipse

Figure 4.37: Results of rolling an object with ellipsoid shape.

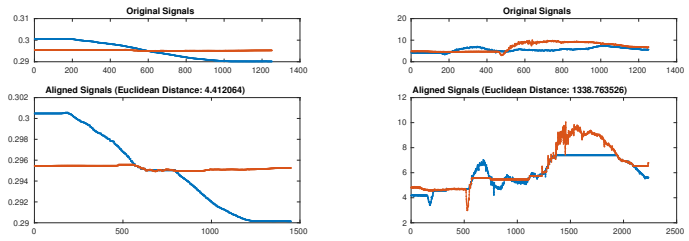
In addition to the changes of the end-effector pose along its z axis, there are differences in the contact profile during the manipulation (Fig. 4.38). It means that the force

response (estimated by the area of contact) is also different for the objects with the elliptical and cylindrical shapes. Since the force responses and the manipulator poses are different during the manipulation of the objects with different shapes, the Euclidean distance computed by the DTW (Section 4.4) can be applied to distinguish the two objects. The results of the DTW applied to the pose and force histories are shown in Fig. 4.39 a and b, respectively. The resulting distances preserve the units of the input variables; the units of the output of DTW applied to positions are therefore in meters and to force are in Newtons. Although the distance in position seems to be more important than in force, for the classification algorithms these distances could be scaled and their relative importance even changes (for instance, in [Robles-De-La-Torre 01], the force overcomes the position perception in the shape recognition tasks).



(a) Before rolling (b) At minimum force (c) After rolling (d) Maximum force

Figure 4.38: Contact profiles during rolling an object with ellipsoid shape.



(a) DTW of position (b) DTW of force

Figure 4.39: Compare results of rolling cylinder and ellipse. Dynamic Time Warping applied to the end-effector pose along z-axis and force: for the ellipsoid and cylinder results are in blue and red colors, respectively.

4.6 Co-manipulation

4.6.1 Introduction to human-robot co-manipulation

Going beyond the traditional applications of robotic systems, the human-robot co - manipulation paradigm has emerged in recent years. The robots that were developed with the support of this paradigm are called collaborative robots or in short cobots. Up to now, the robots and operators did not share the same working space, since the robots have always been not safe for the human operators. However, the cobots are in fact the opposite of the current industrial robots: they are compact, light-weight, and dexterous; they have a rather rich sensory system and safety features. Before their emergence, industrial systems had minimal levels of cooperation with humans. Current research trends in these research line aim to increase the level of cooperation between humans and robots by developing safe [Meguenani 15, Maiolino 13] and collaborative algorithms [Robotiq 14].

The sense of touch is essential in these human-robot cooperative tasks, especially, when there is a physical interaction. For instance, one of the current directions in these direct physical human-robot interactions aims at developing algorithms for heavy load sharing tasks between a robot and a human [Lawitzky 10, Stckler 11, Agravante 14]. Force-torque sensors and vision cameras were used for these heavy load sharing tasks (normally carrying a table together), but not tactile sensing arrays. The tactile servoing framework proposed in this thesis would improve the cobots performing these types of tasks. For instance, we will apply our tactile servoing framework to make a robot carry a tray cooperatively with a human (Fig. 4.40).

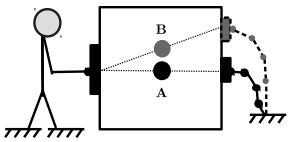


Figure 4.40: Co-manipulation for keeping a tray at equilibrium.

Experimental results

In the following experiment, the tactile sensing array is used so that a robot holds a flat tray from one of the sides of the plate. The tactile array is now attached as shown

in Fig. 4.41 a. The opposite side of the tray is hold by a rotating bearing (Fig. 4.41 b). Thus, the tray can freely rotate (Fig. 4.41 c). The experimental setup is shown in Fig. 4.41 d: a copper bar is glued to the bottom side of the tray in order to enhance the extraction of tactile features; a paper sheet on top of the sensing array is used to eliminate friction forces.

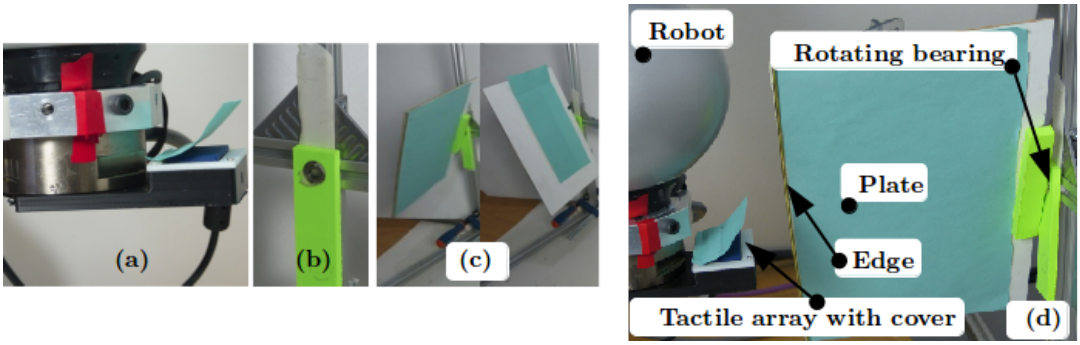


Figure 4.41: Setup for co-manipulation: (a) the tactile sensor on the end-effector, (b) structure with bearing, (c) free rotation of the plate, (d) description of the setup for co-manipulation.

The experimental scenario is as follows. The KUKA robot arm holds the tray from the bottom (Fig. 4.42 a). Initially the center of mass of the tray is at the geometrical center. When the center of mass of the plate moves from the initial location by adding a weight, the edge of the tray applies a moment of force to the end-effector of the robot. Then the robot should compensate this emerged moment of force, thus keeping the tray in balance, i.e. in horizontal orientation wrt ground (perpendicular to the Gravity vector). Fig. 4.42 b illustrates the scenario. The dashed line represents the new configuration of the arm when the center of mass CoM moves from point A to point B. When the applied moment of force is compensated, the CoM lies again on the line connecting the two points that hold the tray at the equilibrium. In the given experiment, the CoM is changed by moving objects on the tray as shown in the same figure: the scale weight with mass of 200 g is moved from point A (Fig. 4.42c) to point B (Fig. 4.42d). In the reality, instead of a fixed structure with a rotary bearing, the opposite side can be hold by a human operator.

The variable elements of the inverse tactile Jacobian matrix α_1 and α_2 are set so that

Figure 4.42: Co-manipulation for keeping the balance: a - the robot arm holding a plate that rotates around its center, b - illustration of the changes of the center of mass of the plate from point A to B, c - a scale-weight is placed on the plate at point A, d - the scale-weight is place at point B.

$\mathbf{J^{-1}} = \begin{pmatrix} 1\ 0\ 0\ 0\ 0 \\ 0\ 1\ 0\ 0\ 0 \\ 0\ 0\ 1\ 0\ 0 \\ 0\ -1\ 0\ 0\ 1 \\ -1\ 0\ 0\ 1\ 0 \\ 0\ 0\ 0\ 0\ 1 \end{pmatrix}$. The only variable of importance is $\alpha_1 = 1$, other variable elements do not contribute to the final error of the contact frame since only four features are needed for the given co-manipulation task: ZMP_x (to keep the equilibrium wrt CoM), CoC_y (to keep the contact at the center of the sensing surface), force (to keep the contact), orientation of an edge (to align the contact along the x axis of the sensing frame). These tactile feature errors are mapped to the error in contact frame by the tactile projection matrix $\mathbf{P} = \begin{pmatrix} 0\,0\,0\,1\,0\,0 \\ 0\,1\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1 \end{pmatrix}$. Hence, the contact frame error is given by:

$$\Delta \boldsymbol{x} = [\Delta ZMP_x, \Delta COC_y, \Delta f, -, -, \Delta \alpha_z]^T \tag{4.12}$$

that drives the end-effector to the new position (dashed lines in Fig. 4.40).

Fig. 4.43a-i show the evolution of features and pose of the end-effector during the task of keeping the CoM in the center of the tray[5]. An object on the tray was moved once from the center of the tray, which would result in turning the tray up-side down if the robot would not react to this change from point A to B. The first marker x indicates

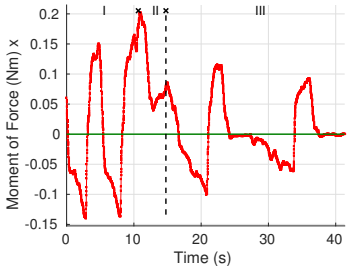
[5]https://www.youtube.com/watch?v=n2A4sMYmIQE&list=UUQEGPPSY-gQjDm9TGokCDuA&index=7

the when the object is moved from the center, which causes the increase of ZMP_x (Fig. 4.43a). The contact frame at this moment is as shown in Fig. 4.43b. The force is desired to be constant around 6 N (Fig. 4.43 c). The error of the ZMP_x feature is mapped to the translational motion of the end-effector (Fig. 4.43d in x-axis and Fig. 4.43g in the y axis of the global frame since the tray was not aligned with neither of the axes) in order to compensate the appeared moment of force. When the robot moves enough its end-effector that holds the tray so that the CoM is again on the line connecting the holding points, the contact frame has a uniform distribution of pressure again (Fig. 4.43e). It is worth noting that the orientation of the line in the tactile image caused by the edge of the tray is also controlled (Fig. 4.43h), and the end-effector rotates around the z-axis of the sensing frame (Fig. 4.43i) in order to keep the orientation of the edge parallel to the x axis of the sensor attached frame. The amount of rotation is equal to 1 deg (0.02 rad in Fig. 4.43h).

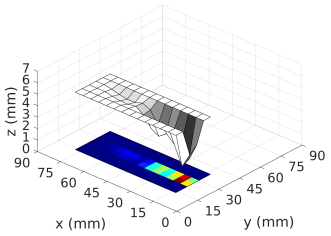
4.7 Conclusions

In this chapter, we have succeeded in elaborating applications of the tactile servoing controller for a robot arm equipped with a pressure sensing tactile array. These applications include object exploration by following an edge of an object, tactile object recognition, manipulation by rolling, exploration by rolling, and co-manipulation tasks. The same controller is used thanks to a special touch dependent inverse tactile Jacobian and projection matrices that map tactile feature errors into the robot's end-effector pose error in a desired way. In fact, a task description layer has been added in order to define the evolution of the tactile feature errors. Thus, the use of our tactile servoing framework (Fig. 3.1) is demonstrated and generalized to deal with a rather high amount of applications that require physical interactions.

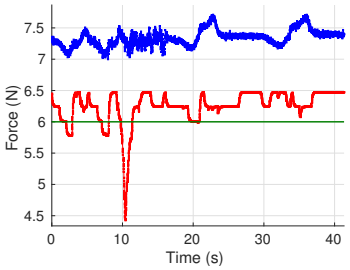
The developed experiments with the 7 dof KUKA arm prove that the framework performs well in different types of manipulation and exploration tasks. The framework can be used for human-robot co-manipulation tasks. The success of the proposed framework will enable many other advances in the domain of dexterous manipulation. In fact, this



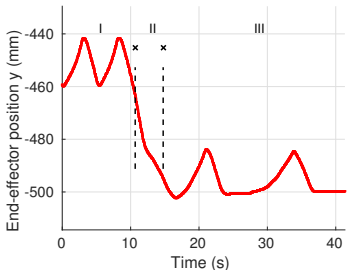
(a) ZMP_x



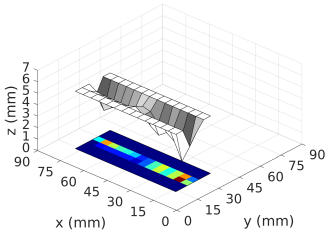
(b) Contact surface at the moment between phase *I* and *II*



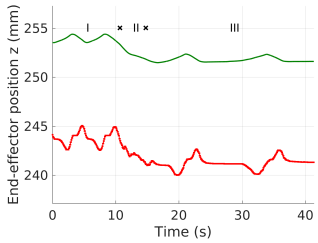
(c) History of force



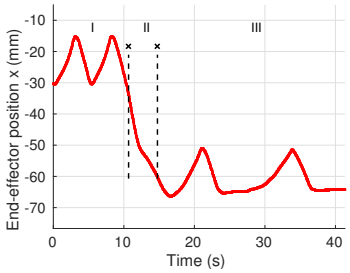
(d) History of y coordinate of the end-effector's pose



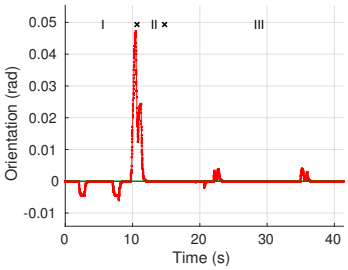
(e) Contact surface at the moment between phase *II* and *III*



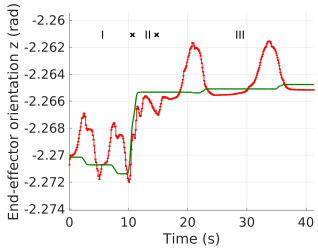
(f) History of z coordinate of the end-effector's pose (green) and the estimated z coordinate of the contact frame (red)



(g) History of x coordinate of the end-effector's pose



(h) Orientation of an edge in the tactile image



(i) History of the angle of the rotation of the end-effector around z axis of the global frame: real (green) and estimated (red)

Figure 4.43: Evolution of features and end-effector pose coordinates during the co-manipulation task.

framework can be used to control a robot hand interacting with the environment using several fingers equipped with pressure sensing arrays. A preliminary work will be shown in the next Chapter dedicated to control the contact with multiple tactile sensors.

Chapter 5
Towards control of multiple contacts

In this chapter, a preliminary attempt is proposed to extend the previous tactile servoing framework for using several tactile sensors simultaneously. The proposed technique is referred as Tactile Localization and Mapping (TaLaM) and will be used to control a dexterous robot hand equipped with a tactile skin that consists of multiple pressure sensing arrays attached onto different links of the hand. Previous works about the integration of these sensing arrays with robot hands are reviewed in Section 5.2. Section 5.3 describes a custom-made tactile skin and its integration process with the fingers of an anthropomorphic robot hand. The prototype of the skin is designed for developing the TaLaM technique (Section 5.4), which is drawn on the Bayesian probabilistic theory. Only a general description of this technique will be presented and its further implementation is left for future work.

5.1 Introduction

A possible extension for the research described in the previous chapters is to apply the developed tactile servoing framework to control a multifingered robot hand performing dexterous in-hand object manipulation. However, in contrast to a robot manipulator that interacts with an object by using its end-effector and having a contact at a single location, contacts between the object and the hand can occur at multiple locations: the hand makes contact with an object using its finger surfaces and palm. This makes the ability to sense these multiple contacts be indispensable. Nevertheless, as will be shown in the following section, most of current robot hands incorporate tactile sensors at their fingertips only. Therefore, in order to provide a robot hand with complete sensing capabilities, its finger surfaces and palm should be covered with tactile sensing arrays.

For this reason, new tactile sensing arrays developed at the Laboratory of System Integration and Technologies (LIST) of the French Alternative Energies and Atomic Energy

Commission (CEA) were integrated all over the surface of a robot hand. In particular, a first prototype of these sensing arrays were tested with the multifingered Shadow Robot Hand. The finger surfaces of the robot hand were replaced by custom-made 3D-printed parts, that were fused with the sensing arrays. A simulation model was also developed by extending the approach described in the Sec. 2.4.1.

Our experiments described in the previous chapter were limited to single-contact cases. Applying the proposed tactile servoing framework in order to perform dexterous manipulation will require control of multiple contacts through finger movements. The final goal of this chapter is to formulate the concept of multipoint contact control (by a new technique called TalaM, Tactile Localization and Mapping) based on the use of multifingered robot hands equipped with tactile sensing arrays (Section 5.4). This should fill the gap that currently exists between tactile servoing algorithms and dexterous manipulation planning.

5.2 Survey of robot hands with tactile skin

Research in design of multi-fingered dexterous robot hands, being previously focused on prosthetic hands only, has surged in last decades. Various robot hands were developed in research laboratories and became commercially available [Company 04], [Twendy-one 14], [Simlab 14]. In recent years, these hands were equipped with more sensors in order to make them more dexterous. In this section we review the existing robot hands equipped with tactile sensors (Appendix B for the types of sensing technologies and appendix F for additional information about system integration issues). These hands are listed in Table 5.1. The most common configurations of multi-fingered hands incorporate three fingers (Fig. 5.1) or five fingers (Fig. 5.2). In the following, the most common ones equipped with tactile sensors will be described.

Tactile sensor	Robot Hand	No. of tactels	Res./Sens./Range	Rate
Piezoresistive sensors				
FSR [Electronics 13]	Robonaut data glove[Diftler 03]	19	5mm/0.1N/20N	1 kHz

Fabric sensor [Eeonyx 14]	Sensor Glove [Büscher 15]	56	$34mm^2$/(0.130N)	-
Rubber-based [Drimus 14]	Schunk gripper [Schunk 14a]	8x8	$6.25mm^2$/-/250kPa	100 fps
Rubber-based [Teshigawara 11]	High-speed 3-fingered hand [Namiki 03]	17x19	3mm/-/-	10 kHz
Weiss Robotics [Weiss 05]	Schunk sDH [Schunk 14b]	(14x6) and (14x7)	3.5mm/-/250kPa	800 fps
3D-shaped sensor [Koiva 13]	Shadow Hand	12	5.5mm/ 0.03$\frac{N}{cm^2}$/ 10N	$\sim$ 1 kHz
Rubber-based [Fukui 11]	Universal robot hand [Fukui 11]	102 on tip	3.6 mm/ 1N/-	50 Hz
Gifu hand sensor	Gifu Hand III [Mouri 02]	624	$\sim$4mm/-/22$\frac{N}{cm^2}$	10 Hz
Tekscan [Tekscan 13]	Shadow Hand [Liu 12a]	349	4mm/-/345kPa	200 Hz
FSR [Kyberd 92]	Southampton hand [Kyberd 92]	15	-	-
ATi Nano17 sensors [ATi 14]	Shadow hand [Shadowrobot 14]	5 per finger	-/ 3.26mN/12N	833 Hz
Weiss Robotics [Robotics 14]	Fluidic FRH-4 hand [Gaiser 08]	14x6	3.5mm/-/250kPa	230 fps
Capacitive sensors				
Icub sensor [Schmitz 10], [Schmitz 11]	iCub Humanoid robot	12 per tip, 48-palm	$7mm$/2.5$\frac{fF}{kPa}$/150kPa	25-250Hz
PPS sensors [PPS 14]	PR2 robot grippers [Romano 11]	22	$4mm$/6.25mN/7kPa	24.4 Hz
PPS RoboTouch [PPS 14]	Allegro robotic hand [Jara 14]	24	$25mm^2$/7kPa	30-100Hz
Dynamic sensor [Cutkosky 14]	Robotiq Gripper [Robotiq 14]	132	-/-12N	300 Hz
Combined sensor [Schmidt 06]	Parallel jaw gripper [Schmidt 06]	16	10 mN	up to 35 kHz
PPS RoboTouch	Barrett Hand [Barret 14]	120 per finger	$5mm$/6.25mN/7kPa	30-100Hz
Piezoelectric sensors				
PRes. [Robotics 14] + PVDF [Göger 09]	8 DoF Fluid Hand [Schulz 04]	4x7	3.5mm/-/250kPa	$\geq$1kHz
PRes. ink + PVDF [Choi 06]	SKKU Hand II [Choi 06]	24 on fingertip	0.5mm/-/-	-

Tactile skin [Strohmayr 13]	DLR Hand [Grebenstein 12]	in	process of	development
Barometric measurements based sensor				
Takktile (silicon) [TakkTile 13]	iHY Robot Hand [Odhner 14]	24+48	5mm/10mN/4.9N	50 Hz
BioTac (liquid) [Fishel 08]	Shadow Hand [Shadowrobot 14]	1 per finger	-/0.1N/3N	1 kHz
QTC tactile sensors				
Robonoaut sensors	Robonaut Hand [Martin 04]	33	-/0.1N/10N	-
Piratech [Peratech 14]	Shadow Hand [Company 04]	36	$3\frac{kN}{m^2}/400\frac{kN}{m^2}$	-
Optical tactile sensors				
Sensor for MRT [Xie 14]	Robot Manipulator	9	-/0.5N/5N	25 fps
3DoF sensor [Yussof 09]	Robot gripper	41	3mm/0.08N/1.8N	10 Hz
Optoforce [LTD. 15]	Barret Hand	1 per finger	10mm/-/10N	-
Multi-modal tactile sensors				
Proximity Sensor [Hasegawa 10]	A three-fingered hand [Hasegawa 10]	Palm: 5x6	10cm and 2cm	1 kHz
BioTac sensor [SynTouch 13]	Shadow Hand [Xu 13]	19 + fluid barometer + thermistor	-/∼0.01N/1:1000	50Hz; 2kHz; 50 Hz
Optical+ PVDF+ Force	3-fingered gripper [Kampmann 14]	324 fibers, 120 PVDF, 3 F/T	-/-/4N	30fps; 10kHz; 100Hz
"Structure-borne sound" tactile sensors				
Microphone	Oxford prosthesis [Kyberd 98]	1	-	-
Accelerometer	PR2 robot grippers [Romano 11]	1	$0.15m/s^2$	3 kHz
SeaShell effect sensor [Jiang 12]	PR2 robot grippers	1	-/Non/Non	44 kHz

Table 5.1: List of tactile sensors that have been integrated with robot hands. Number of tactels (No.), spatial resolution (Res.), sensitivity (Sens.), dynamic range (Range) and data acquisition rate (Rate) are provided where available.

In [Drimus 14], an 8x8 tactile array based on piezoresistive rubber has been attached onto the grippers of the 3-fingered Schunk SDH hand for classifying deformable objects. Outstanding in speed performance, the Lightweight High-Speed Multifingered Hand System [Namiki 03] integrates a Center-of-Pressure (CoP) sensor for the force measurements and a PVDF[1] based high sensitive tactile sensor for slip detection, as shown in Fig. 5.1(a) [Teshigawara 11]. The commercial 3-finger Schunk SDH hand [Schunk 14b] with integrated Weiss Robotics piezoresistive tactile sensors [Robotics 14] incorporates a 14x6 array on each distal link and a 14x7 array on each middle link. The Universal robot hand [Fukui 11] has 102 tactels on fingertips and 70 tactels on the rest of the links. In contrast to the serial connection of sensors present in the iCub skin [Schmitz 11], each tactile array has its own connection with the acquisition board (Fig. 5.1(b)). The Barret hand [Bar-

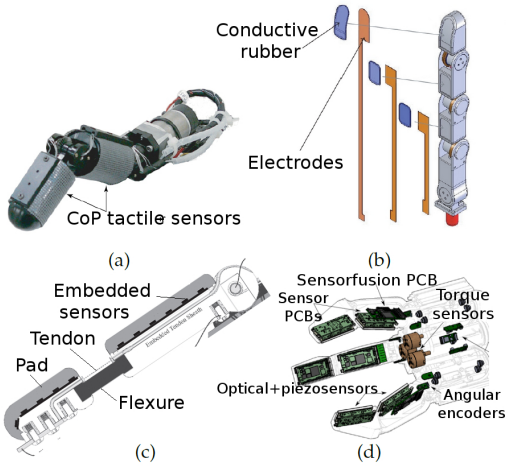


Figure 5.1: Three-fingered robot hands with tactile sensors: (a) a finger with tactile sensor of the 3-fingered high-speed robot hand [Namiki 03], (b) assembly of tactile sensing arrays with a robot finger of the Universal robot hand with 3 movable and 2 immovable fingers [Fukui 11], (c) schematic illustration of a finger of the iHY robot hand with embedded array of pressure sensors based on digital barometers placed inside the soft paddings of the fingers [Odhner 14]; (d) schematic illustration of the integration of a multimodal sensing system with a three-fingered robot hand [Kampmann 14].

ret 14] has capacitive tactile sensors on the tips, distal links and palm. "Takktile" arrays [TakkTile 13] based on barometric measurements have been integrated with the iRobot-Harvard-Yale (iHY) Hand [Odhner 14]. The hand is covered by an array of 48 tactels on

[1] polyvinylidene fluoride

the palm, 2x6 arrays on proximal links, and 2x5 arrays on distal links with two of the tactels on the tip (Fig. 5.1(c)). Fig. 5.1(d) illustrates a robot hand with a multimodal sensing system, including optical tactile sensors and force/torque sensors [Kampmann 14]. These force torque sensors are placed at the base of each finger, not at the tips as in the Shadow hand [Liu 15]. The three-axis opto-force sensors [LTD. 15] can be installed on the tips of the Barret Hand [Barret 14]. In [Koyama 13], photo-reflectors have been attached to the three-fingered robot hand to provide proximity information for preshaping the fingers during grasping. The Seashell effect sensors [Jiang 12], which also provide proximity information, can be installed on robot grippers for the same purpose.

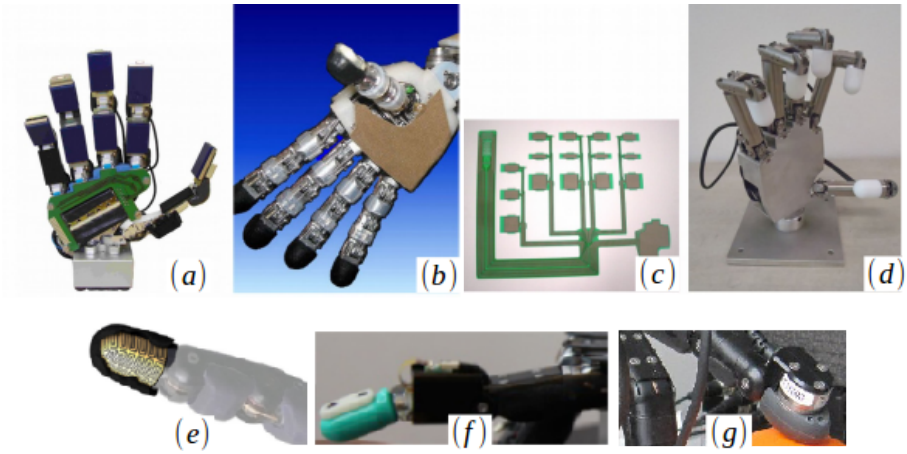


Figure 5.2: Five-fingered robot hands with tactile sensors: (a) the fluidic robot hand with combined piezoelectric and piezoresistive tactile sensors that can sense high-frequency vibrations due to the absence of electric motors [Göger 09], (b) the robot hand of the iCub humanoid robot with tactile sensors on the fingertips and the palm [Schmitz 11], (c) flexible tactile sensing arrays of the SKKU robot hand [Choi 06], (d) the SKKU robot hand [Choi 06], a 3D-shaped rigid tactile sensing array with 12 sensing elements attached to the fingertip of the Shadow robot hand [Koiva 13] (e),(f) the BioTac multimodal tactile sensor installed on the Shadow robot hand by replacing the last two links of the finger [Xu 13], (g) ATi nano 17 force/torque sensor on the fingertip of the Shadow robot hand [Corrales 13].

The tactile sensing system for the five-fingered DLR robot hand-arm system [Grebenstein 12] is designed as large scale tactile skin using the column - row net structure [Strohmayr 13]. The robonaut hand has tactile feedback through a tactile data glove

incorporating piezoresistive technology [Diftler 03] and QTC[2] technology [Martin 04]. Fig. 5.2(a) shows the Fluidic hand [Schulz 04] with a modified version of the Weiss [Robotics 14] sensors. The robot hand of the iCub humanoid robot [Metta 08] has sensors on the fingertips and palm, but not on the middle and proximal phalanges (Fig. 5.2(b)). The dexterous Gifu III robot hand (Fig. 5.2(d)) has a sensing array of 859 tactels (Fig. 5.2(c)) based on piezoresistive conductive ink [Mouri 02]. An array of 24 conductive ink in combination with piezoelectric PVDF material has been used in the SKKU II robot hand [Choi 06]. The Shadow Hand [Shadowrobot 14] has different integrated tactile sensors: force/torque sensors (Fig. 5.2(g)) [Corrales 13], multimodal Biotac tactile sensors (Fig. 5.2(f)) [Xu 13], 3D-shaped fingertip tactile sensors (Fig. 5.2(e)) [Koiva 13], and QTC sensors [Company 04].

An analysis of advantages and disadvantages of these hand/sensor combinations is given in Table 5.2.

Hand/Sensor Combination	Advantages	Disadvantages
3D-shaped array [Koiva 13] & Shadow Hand; iCub robot fingertip sensor [Maiolino 13]	Multiple point of contact, covers spherical shapes, wires - within fingers	Normal force measurements only
Ellipsoid f/t sensor [Liu 15] & Shadow Hand; OptoForce [LTD. 15] & Barret Hand	Covers spherical shapes, high sensitivity, shear forces	Single point of contact only, wires - outside of fingers
BioTac [SynTouch 13] & Shadow Hand	Multiple point of contact, high bandwidth, wires - inside	Last joint static (20 degrees)
Robonaut glove and Hand [Diftler 03]	Ease of replacement, low cost	Not reliable compared to rigidly attached sensors
Fabric sensor [Büscher 15]	Ease of replacement, stretchable	Wear and tear off
Tactile sensing array (PPS [PPS 14], Tekscan [Tekscan 13], and etc.) & any robot hand	Can be easily attached to any flat and cylindrical surfaces	Cannot cover spherical shape, wiring issues
Weiss Robotics [Robotics 14] & any robot hand; Takktile [TakkTile 13] & iHY hand	Robust	Flat surface only
SeaShell effect sensor (Cavity with microphone & PR2) [Jiang 12]	Pre-touch sense	Direct contact of the cavity with an object limits forces

[2]quantum cascade

Proximity Sensor [Hasegawa 10]	Pre-grasp sense	Cannot measure very close proximities
Accelerometer at the base of robot grippers [Romano 11]	Vibration detection	Interference with electric motor noise
Microphone at the tips of the Oxford Hand prosthesis [Kyberd 98]	Vibration detection, No interference with motor noises	Can not measure forces
A fluidic robot hand with piezoresistive tactile sensors [Göger 09]	Vibration detection	Simple surface of phalanges

Table 5.2: Sensors integrated with robot hands: advantages and disadvantages of major approaches.

In this analysis, we can see two main solutions to integrate tactile sensors with robot hands: sensors directly integrated inside the structure of phalanges or sensors attached on the phalanges by some flexible elements. In the first solution, very simple shapes (i.e. planar, spherical, cylindrical and ellipsoids) for phalanges are developed in order to simplify the integration. In addition, in order to reduce wiring connections, most approaches only install sensors on the fingertips. These two limitations will not permit us to implement the TaLaM technique as we need to sense a maximum number of contact points while manipulating an object inside the hand. The second solution can be applied to complex shaped phalanges capable of performing in-hand rolling and sliding movements. Nevertheless, the approaches based on this solution suffer from a lack of reliability in their measurements (for instance, in glove-based systems, the sensors can be moved from their initial location while manipulation) and a lack of durability (for instance, fabric-based sensors suffer from tearing-off).

We can combine the advantages of both solutions by defining a new integration procedure: designing a specific surface for latter solution to wrap around complex shapes and the advantage of the second solution by integrating flexible tactile pressure sensing arrays on the finger surfaces and palm of a robot hand.

5.3 CEA tactile skin on the Shadow hand

In the following, the ongoing work on the integration of an artificial skin onto the Shadow robot hand is described. The skin was developed at the CEA. This CEA tactile robot skin is made of tactile pressure sensing arrays. Unlike the WTS 0614 WeissRobotics sensors, these CEA arrays can be bended so that they can wrap the finger paddings of the robot hand.

The integration of the tactile skin with the hand, which was not initially designed to be used with other sensors, requires reverse engineering for replacing its finger surfaces when the robustness of the system is of importance.However, when the robustness is not an issue, these arrays can be attached directly on top of the original finger surfaces as shown in Fig. 5.3a. It illustrates a pilot version of the integrated tactile sensing arrays, used to verify the electro-magnetic compatibility: the outputs of the sensors (Fig. 5.3, c) were recorded when a current was supplied to the motors of the hand during grasping an object (Fig. 5.3, b). These tactile sensing arrays are described in the following.

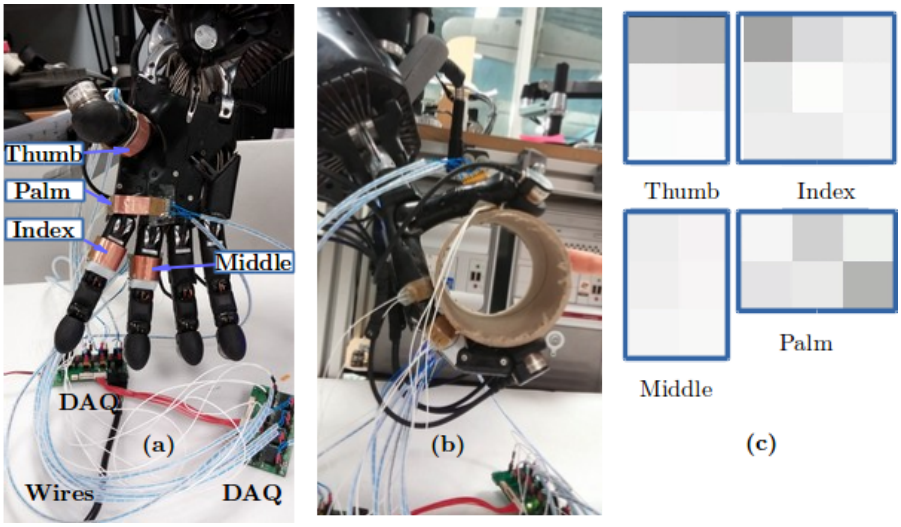


Figure 5.3: The first prototype of the integration of the CEA sensing skin: (a) Shadow robot hand with the CEA tactile sensing arrays and wiring issues; (b) hand grasping an object; (c) tactile contact images acquired by the CEA sensing skin during the grasp , where the black color stands for the highest pressure value and the white color for the lowest.

5.3.1 Capacitive tactile array

The tactile sensing arrays developed by CEA are based on the capacitive technology (Section B.1.2). Each array consists of two thin layers of flexible PCB[3], which makes the structure bendable, and a dielectric material that is deformable. Fig. 5.4a and b show these layers of a pilot version of the CEA tactile sensor, which consists of a 3×3 array of capacitors with the size of $1.1 \times 1.1\ mm^2$ and a total size of $21 \times 57\ mm^2$. In between these two layers, there are pieces of Polyethylene material (deformable dielectric). The assembled sensing array is shown in Fig. 5.4 c. The copper layer on the top is the shield that protects from electro-magnetic interferences. The thickness of the Polyethylene material is a compromise between the dynamic range and the sensitivity. A thicker layer of the dielectric material enables larger mechanical deformations, and, therefore, increases the dynamic range of measurable forces. However, it reduces the sensitivity of the sensor.

For the 3×3 array of the tactile skin, twelve wires are required (Fig. 5.4 d). Wires soldered in this way decrease the robustness of the system in total: they can be easily worn off. This can be solved by soldering a signal conditioning board directly on the flexible copper layers, as for example in the PPS [PPS 14] tactile sensors (Fig. 5.4 e).

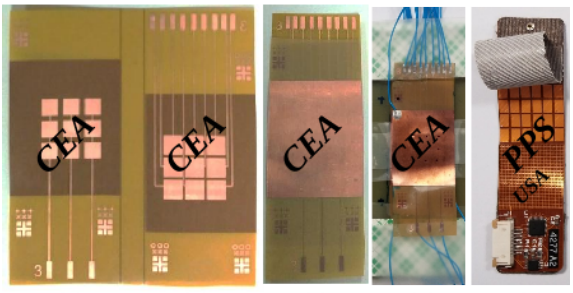


Figure 5.4: The structure of the pilot version of the CEA tactile sensing array and the 4×6 Pressure Profile System sensor with Chip on Board technology. (a) Bottom and (b) top layers of the CEA sensor; (c) assembled CEA sensor; (d) CEA sensor soldered with wires; (e) PPS sensor with COB (chip on board) technology [PPS 14].

[3]Printed Circuit Board

Data acquisition block diagram

The measurement of the capacitances of the tactels of the CEA skin was realized by the Smartec Universal Transducer Interface (UTI) [Smartec 16]. The UTI transforms analog signals from capacitive sensors into digital values (13 bits) that can be read directly by a microcontroller unit. The maximum capacitance that the UTI can sense is the following: 2 pF, 12 pF, and 300pF depending on the model of the UTI.

The maximum number of tactels (capacitors) depends on the time that is required to poll all of the capacitors. These tactels are connected with the UTI through a coaxial cable or directly via route in its PCB. Since the tactile servoing controller operates at 100 Hz, the sampling rate of the CEA sensor skin is desired to be 100 Hz. Therefore, there are 84 tactels in the CEA tactile skin. The block diagram of the tactile skin and the robot hand is shown in Fig. 5.5. Besides the CEA sensors, the Shadow Robot hand incorporates ATi Nano 17 force sensors connected with a Data Acquisition board (DAQ). The motion controller of the hand communicates with the computer (PC) via EtherNet at the rate of 1 kHz and controls the joints of the hand, including joint angles and tensions of tendons.

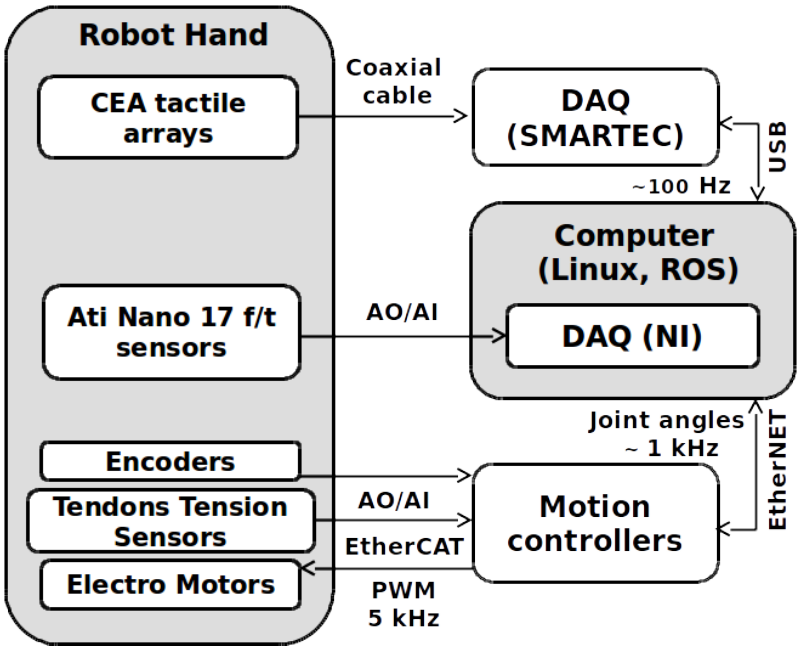


Figure 5.5: The CEA tactile skin and the Shadow Robot Hand Block Diagram of the integrated system.

5.3.2 Mechanical integration

The constraints set by the Shadow robot hand were the following:

- the fingertips are already equipped with ATi Nano 17 force and torque sensors
- the output wires of the CEA skin cannot be neither routed within the structure of the hand nor connected to the Printed Circuit Boards (PCB) of the hand's motion controllers, since the number of input pins in Shadow hand (version of 2009 year) is limited to 5;
- the claddings of the fingers have rubber paddings that do not allow attaching the sensing arrays fixedly (rigidly).

The maximum number of tactile sensing cells was limited by the hardware. Therefore, the selection of the locations of the sensing elements was purely based on technical aspects without taking into account the studies on human hands [Johansson 09, Gonzalez 14]. The number of the tactels on each of the phalanges of four fingers (index, middle, ring, little) is the following: on the intermediate link, there are three tactels with one tactel on each of the faces except the back side (Fig. 5.6, middle) and on the proximal link, there are eight tactels with four of them on the front side and two tactels on each of the sides (Fig. 5.6, on the right-hand side). The palm will be allocated with fourteen tactels, the proximal and middle phalanges of the thumb will be equipped with six and four tactels, respectively. In contrast to the pilot version, the UTI chips were mounted directly on the flexible PCBs in the first prototype of the integrated tactile skin. Thus, the number of wires was significantly decreased.

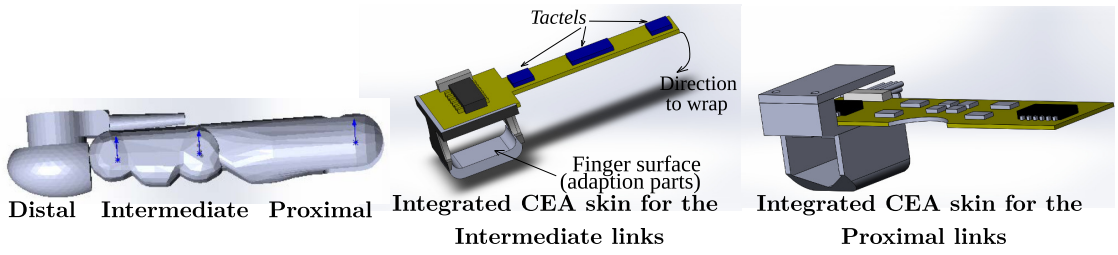


Figure 5.6: CAD models of the Shadow Robot hand finger and the integrated tactile sensing arrays on the intermediate and proximal links.

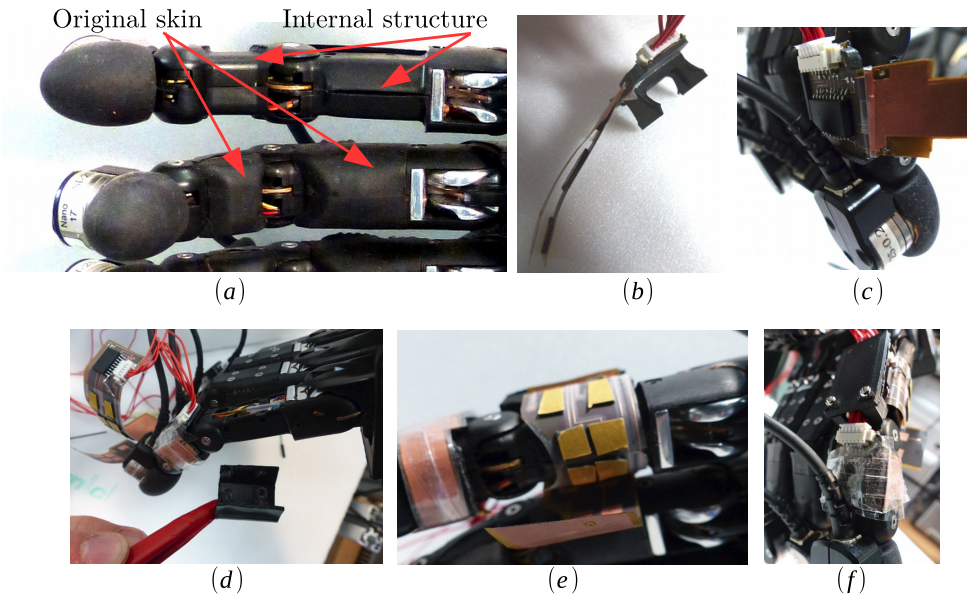


Figure 5.7: Integration of the tactile skin with the robot hand. (a) Original skin of the fingers and internal structure . (b) Integrated tactile skin for the intermediate link. (c) Intermediate link with the CEA skin. (d) Integration of the CEA skin on the proximal link by preheating a 3-D printed structure that allows the plastic piece to bend. (f) Proximal link with the CEA skin (e). The wire routing of the CEA skin on the back sides of the links.

The original skin of the hand fingers was removed (Fig. 5.7 a) and reversed engineered. The tactile skin was then glued on the new skin (i.e. the adaptation parts) manufactured using a three dimensional rapid prototyping machine (Fig. 5.7b). The data acquisition integrated circuit and wires were placed on the back side of the phalanges (Fig. 5.7c). In order to attach the new skin on the finger structure, the printed parts were preheated to make them bendable (thus, they did not break) and were then attached to the structure of the finger (Fig. 5.7 d). The skin with its internal components is shown in Fig. 5.7e. The wires of the middle phalanx skin go through the back side of the proximal phalanx (Fig. 5.7f).

We have also developed a simulation model of this integration of the CEA sensors with the Shadow robot hand. Following the same approach as was described in Section 2.4 the simulated sensors were added to the hand model in the Gazebo simulation environment (Fig. 5.8, a). The number of tactels and the arrangement of the tactels in the simulated tactile sensing arrays can be adjusted to be the same as the real CEA sens-

ing arrays[4]. The outputs (tactile images) of the simulated tactile arrays attached onto the proximal and intermediate links of four fingers are shown in Fig. 5.8c. These tactile images were acquired when the hand grasped an object (Fig. 5.8b).

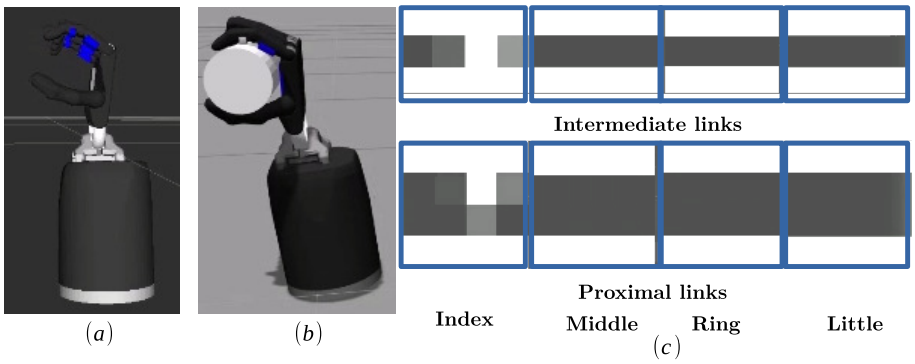


Figure 5.8: Tactile sensors on a robot hand in the Gazebo simulation environment. (a) Robot hand model. (b) Robot hand grasping an object . (c) Tactile images during this grasp.

5.4 TaLaM

The finger surfaces of the robot hand were covered with the sensing arrays in order to be able to sense and then control multiple points of contact between the hand and an object. Simultaneous Localization and Mapping (SLAM), normally used in the context of localizing mobile robots in unknown environments, is extended to solve this problem. In the case of mobile robots, this technique fuses signals from several sensors (for instance, an inertial sensor and range scanners) to localize a robot wrt landmarks (red dots in Fig. 5.9a) and also to build a map of the environment simultaneously. The contact points that occur during dexterous manipulation can be considered in a similar way to these landmarks. Drawing on this localization and mapping technique in SLAM, we propose an approach of controlling robot hands with multiple tactile arrays and refer to as Tactile Localization and Mapping (TaLaM).

Let us assume that vision is not available or the object is occluded. In order to manipulate is within a robot hand, the object must be localized first: the position and orientation

[4]the number of tactels and the arrangement are defined in the URDF (Unified Robot Description File) file.

of the object wrt the palm of the hand must be determined. Let us also assume that the shape of the object is known in advance. Then the points of contact can be defined (i.e. Cartesian mapping) wrt the object's frame (M in Fig. 5.9 b). The position of the frame can be set at the Center of Mass (M) and the orientation can be defined from the object axis of interest (AoI), which is a characteristic direction (for example, the longest dimension or the laser axis for a laser pointer as shown in Fig. 5.9 c).

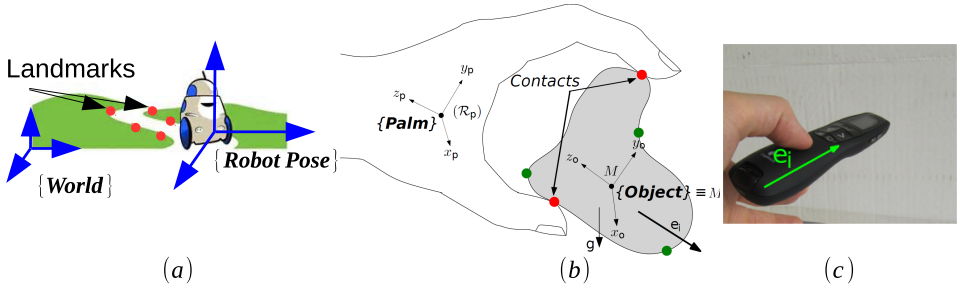


Figure 5.9: SLAM and TaLaM. (a) The illustration of SLAM. (b) An object within a hand with two contact points depicted with red dots [adopted from [Nguyen 13a]], (c) AoI (axis of interest) e_i of a laser pointer [adopted from [Nguyen 13a]]. In order to localize the object, a characteristic axis e_i is defined and referred to as object axis of interest. M is the center of mass of the object.

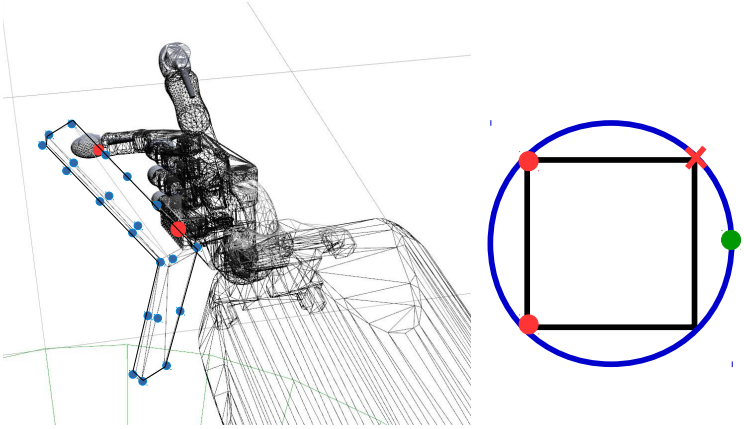
5.4.1 Localization at a single grasp

It was assumed that the shape of an object was known, then its shape can be represented by a set of k points $\{p_k\}^{\mathcal{O}}$, where $\mathcal{O}$ is the objects' frame and p_k are the coordinates of a point k on the surface (the blue points in Fig. 5.10a). Given the geometry of the hand and the signals from tactile sensors, there are n measurements $y_1, y_2, ..., y_n$ coming from n tactile sensors at a single grasp (the red points in Fig. 5.10a). The Cartesian transformation from the sensor frame n to the palm frame P is:

$$A_n y_n + b_n = z_n, \tag{5.1}$$

where A_n and b_n are the rotation matrix and translational vector, respectively, and z_n is the measurement n in the palm frame. In order to localize the object we need to map the cloud of points of contact $\{y_n\}^P$ to the cloud of points in the model $\{p_k\}^{\mathcal{O}}$ by find-

ing an optimal transformation ${}_P T^{\mathcal{O}}$. The Iterative Closest Point (ICP) algorithm can be employed for this purpose.



(a) (b)

Figure 5.10: The Gazebo simulation environment: (a) the Shadow Robot Hand in contact (the contact points are depicted with red dots) with an object. (b) Illustration of the maximum distinctiveness (discrimination) for the next action of exploration.

5.4.2 Localization by manipulation

If there are i solutions $[{}_P T^{\mathcal{O}}_{(1)}, {}_P T^{\mathcal{O}}_{(2)}, ... {}_P T^{\mathcal{O}}_{(i)}]$ that make match the two point clouds (i.e. the mapping between the contact points and the model) at a given grasp of an object (for instance, as shown in Fig. 5.10a, we will get the same point cloud if we would be touching the bottom side of the L-shape object) the robot hand will need to explore the object. Hence, a policy to choose which is the next manipulation action is needed. Bicchi [Bicchi 00] identified several mechanisms for manipulation actions:

- regrasping
- finger gaiting
- sliding
- rolling

The next manipulation action should be chosen so that it will be the most informative for localizing the object. This problem can be solved by probabilistic **Bayes filters** with further **Sequential Importance Resampling** at each new manipulation action [Petrovskaya 06]. For the sake of clarity, let us assume that the manipulation is realized via

regrasping. Therefore, there can be multiple grasps denoted as $G_1, G_2, ..., G_{N_g}$ and Z_{G_1} represents the measurements $z_1, z_2, ..., z_n$ at the grasp G_1. For a given grasp G_j (where $j = 1, 2, ..., N_g$), each possible transformation ${}_P T^{\mathcal{O}}_{(i)}$ (where $i = 1, 2, ...N_s$; N_s being the number of believes/samples for the grasp G_j) will represent an hypothesis of the state $x^{[i]}_{G_j}$ of the object inside the hand. Then the posterior believe $Bel(x^{[i]}_{G_j})$ given the grasp G_j of this state will be:

$$Bel(x^{[i]}_{G_j}) = P(x^{[i]}_{G_j} | Z_{G_j}) \tag{5.2}$$

A possible way to calculate this posterior believe is to obtain the sum of all Euclidean distances $d^{[i]}_{G_j}$ between each measurement (or each contact point z_n) and its corresponding object model point p_k for each state i:

$$d^{[i]}_{G_j} = \sum_{\forall n \in G_j} \| z_n - {}_P T^{\mathcal{O}}_{(i)} \cdot p_k \| \tag{5.3}$$

and later obtain the probability of this state (i.e. sample) as the ratio of its inverse distance to the sum of all the inverse distances in all the states (samples) of a given grasp G_j:

$$Bel(x^{[i]}_{G_j}) = \frac{1/(d^{[i]}_{G_j})}{\sum_{i=1}^{N_s} 1/(d^{[i]}_{G_j})} \tag{5.4}$$

A more complex – presumably, more reliable – way would include the mapping between tactile features (for instance, the orientation of an edge in the tactile image) and not contact points only. At any next grasp G_{j+1}, probabilities of some of the states will increase or decrease. Then those states that will have higher probabilities will be **resampled** and the ones with lower probabilities will be dropped out.

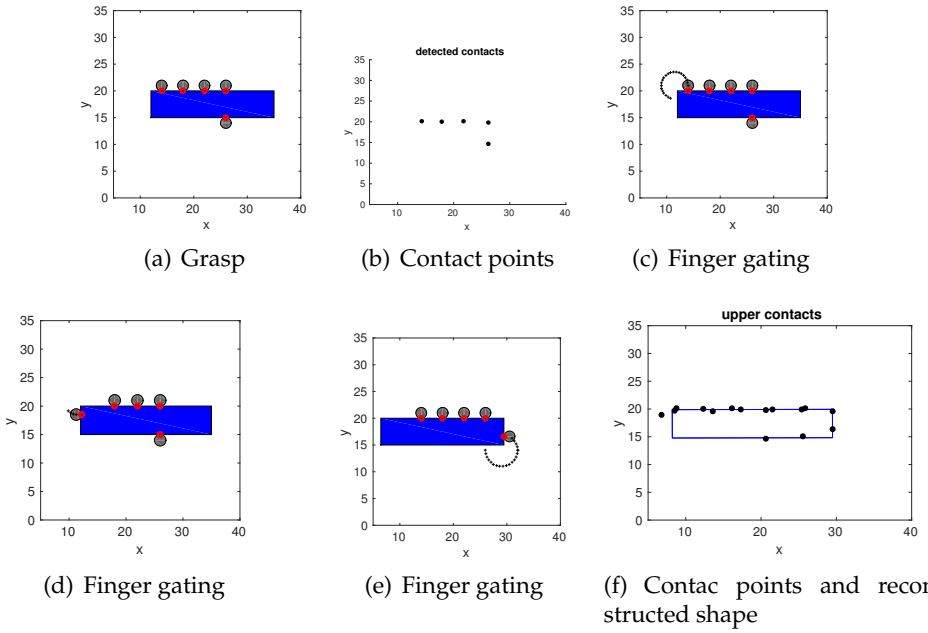
Defining the policy for the next grasp is a rather challenging task. One of the simplest solution is to find for each finger such a new contact point on the surface that would give the maximum differentiation between all the considered states of the object. An example is shown in Fig. 5.10 b: the current points of contact are depicted as the red dots; in order to classify the shape whether it is a circle or square, the next contact point to test should be at the location of the green dot rather than at the location of the red cross, which would provide a higher distinctiveness (i.e. discrimination).

5.4.3 Advancing the TaLaM technique

So far, we have assumed that the shape of the object was known in advance and that the robot hand regrasped the object to obtain new contact points (i.e. tactile contacts appeared and disappeared). In order to make the TaLaM technique more general, we can consider cases when the hand explores an unknown object and does not loose contact during this exploration.

If the shape of the manipulated object is not given in advance, the points of contacts should be ***simultaneously* localized and mapped** (i.e., on-line reconstruction of the surface of the object) during the manipulation of the object. Since the proposed tactile servoing does not require a model of the object, it can be applied to perform a *simultaneous* tactile localization and mapping (sTaLaM). A simple example of an 1D object exploration and localization algorithm is illustrated in Fig. 5.11: gray and red circles represent fingertips and their contact points with a rectangular object, respectively; the cloud of the registered points of contacts (with added noise) is depicted with black dots. In this example, the trajectories for motions (circular sequences of crosses) of the fingertips were predefined. The hand performed finger gaiting in order to find contacts with the object: at each finger gating (action), the robot gathered a new point of contact that is used to reconstruct the shape of the object. Thus, each new action can be chosen thanks to the previously proposed TaLaM technique by extending the state of the object with all the points of its surface and not only its global location.

As already mentioned above, we have considered regrasping and finger gaiting manipulation mechanisms, in which transitions between two grasps are discontinuous. However, in order to increase the robustness and dexterity of the manipulation, these transitions should be continuous. For doing this, the proposed tactile servoing controller (Chapter 3) can be used to keep the contacts with the object, and, thus should be extended to tasks based on multiple-contact control. These tasks could be formulated, for example as an optimization problem (i.e., by a quadratic objective function with some constraints, including joint angle limits, motors' torque limits, etc.). Then the solution of this optimization problem would drive the hand towards a next hand-object state that will improve the knowledge about the object (localization and reconstruction).



(a) Grasp (b) Contact points (c) Finger gating

(d) Finger gating (e) Finger gating (f) Contac points and reconstructed shape

Figure 5.11: 1D example of the exploration and Localization of an object within a hand.

5.5 Conclusions

In this chapter, we have succeeded to integrate a prototype of a tactile skin with a multifingered dexterous robot hand. This tactile skin was developed at CEA, Paris. It uses capacitive technology and consists of pressure sensing arrays attached onto the links of the hand. With this tactile skin, the robot hand will be able to control contacts that occur at multiple locations. In order to control these contacts, a more general framework was proposed, that is drawn on the existing Simultaneous Localization and Mapping (SLAM) approaches. The framework was referred to as Tactile Localization and Mapping (TaLaM). It consists of Bayesian filters, linear quadratic optimization algorithms, and the tactile servoing framework developed in this thesis. The main goal of this framework is to localize a known object within a robot hand or even to reconstruct the shape of an unknown object (sTaLaM, simultaneous TaLaM) by manipulating them. Only the general bases of this approach were described; it will be implemented and tested in future work.

Chapter 6
Conclusions and future work

6.1 Tactile sensing and touch-driven motion

Among the available exteroceptive sensors, tactile sensing arrays offer a greater opportunity to sense the physical properties of the environment to be explored, of objects being manipulated, and of contact during human-robot co-manipulation tasks. An advantage of tactile sensing over vision is that touch can deal with occlusion and provide a more complete information about the contact. Based on planar pressure sensing arrays, the goal of this work was to design, implement, and validate a new tactile servoing framework to control physical interactions in order to perform manipulation by rolling, exploration, deformable object recognition, and co-manipulation tasks.

The first part of this work was to study the existing computational techniques to extract the information from the signals provided by a pressure sensing array – array of pressure or force measuring elements – with an elastic layer. When an object presses the elastic layer of the sensor, a contact frame can be reconstructed using the pressure/force values at each sensing element. The reconstruction of this contact frame can be computed by tactile features extracted from the tactile images of the sensor by using machine vision techniques. The tactile feature space proposed in this thesis (Chapter 2) has a higher dimensionality than in previous works related to touch-driven control. In fact, a new tactile feature ZMP was defined to represent non-uniform line force distributions in edge contact types that were not considered previously.

The proposed tactile features were later incorporated in a new tactile controller (Chapter 3) that would drive a robot to steer the contact frame in a desired manner. Moreover,

touch-driven directions of motion (or tactile affordances that obey natural and artificial constraints) available at the contacts between a robot end-effector and the environment depend on the type of contact: edge- and point-contact types. This was realized via introducing touch-dependent inverse tactile Jacobian (reconstruction of a set of tactile feature errors depending on the point- or edge-contact types) and task projection matrix (mapping tactile feature errors from the tactile image space to the Cartesian space). The dynamic modification of these matrices enabled a reconfigurable tactile servoing scheme that can be applied in a high number of physical interaction tasks.

In order to make this tactile control strategy more general, a task description layer has been added to the proposed framework (Chapter 4). Using the defined affordances (i.e. task descriptions), it was shown that a robot could switch its behaviors. For instance, the robot can increase the surface of contact until the appearance of an edge of an object in the tactile image and later slide over the edge of the object in order to explore it. Thus, we achieved a reactive touch control strategy. The proposed tasks have been evaluated in a rather high amount of experiments to show that the tactile-based approach succeeds in all of them. In the following, the precise achievements of each of these tasks are described.

6.2 Applications of touch-driven robots

Each robotic application conducted within this thesis required its own task description that was specified by a set of desired tactile features. The tactile features were incorporated into a feedback control loop by activating corresponding elements in the projection matrix accompanied with resetting PI gains of the feature errors. Experiments for the following tasks have been conducted throughout this manuscript:

Co-manipulation

In industrial applications with co-shared workplaces of human operators and robots, robotic assistance and interactions are of importance. A co-manipulation strategy using tactile pressure sensing arrays has been proposed for a human-robot cooperative task for load sharing: controlling the equilibrium of a tray that is hold by a robot end-effector from one side and a human operator from the opposite side. The robot was able to adapt

its movement based on the resulting changes in the corresponding tactile features in order to achieve this task. Previous works for doing this type of cooperation task were limited by the use of conventional force sensors. Our system was able to improve the contact alignment and the equilibrium of the tray by incorporating the new ZMP feature obtained from a tactile sensing array.

Deformable object recognition

Object recognition is a good benchmarking task to drive the development of techniques for controlling the physical interactions with an object in order to estimate and represent the object's physical characteristics, including their deformabilities (i.e. stiffness constants). While evaluating the tactile force control over objects with different stiffness constants, it was observed that the tactile responses, including the area of contact and force, can vary for different amplitudes of applied normal forces. Therefore, the Euclidean distance given by the Dynamic Time Warping (DTW) algorithm has shown to serve as a good signature to distinguish deformable objects from rigid objects. In contrast to previous approaches, where only the tactile information was used, we have applied the DTW to the absolute force measurements (given by a ground-truth force sensor) in addition to the area of contact obtained by a pressure array in order to get better signatures to differentiate objects based on their stiffnesses.

Object recognition by manipulation

Since the tactile controller can be applied to manipulate an object without knowing its shape a priori, it can be reconstructed during the manipulation as shown in the experiment of rolling a 3D printed object with ellipsoid shape. It has been also shown that the objects can be recognized not only by their geometrical information, but also by the combination of the tactile contact area of the sensor and the proprioceptive information of the robot. Thereby, the DTW algorithm has been applied to the history of the position of the KUKA arm's end-effector and to the sequence of contact areas extracted from the corresponding tactile images. This permits us to differentiate objects with cylindrical shape from objects with ellipsoid shape by rolling them, while previous works considered objects with spherical shape only.

Exploration

The nuclear plant disaster in Fukushima, Japan on 11th of March, 2011 showed that autonomous robots are not fully ready to work in unstructured and variant environments. The tactile perception increases the reliability of the robots in such environments, since the vision information can be not available for many different reasons, including smoke, occlusions, etc. We have shown that a robot driven only by the tactile perception can explore a variant environment, as for example, a floating metal bar whose stiffness and radius of curvature were not constant. Our approach succeeded to perform this task thanks to the use of a combination of tactile features, including ZMP, meanwhile previous approaches would fail as shown in our experiments.

6.3 Further work in touch-driven robot control

In general, there are two essential milestones for future investigations:

1) there is the demand for a more general approach to manipulation, where a task, a robot hand, and grasping abilities are jointly considered by including information from different sensing modalities and multiple tactile sensors as discussed in Chapter 5. Therefore, a global framework based on multi-sensor information for real-time and real-life applications is needed.

2) there is the demand for "tactile intelligence". It is worth mentioning that all of the works cited earlier and the approach outlined in this thesis have one basic idea in common: the sensed information is static. The sensors are employed to take a "*snapshot*" of the environment. This snapshot is then processed to yield the contact frame. However, the concept of touch extends beyond this: it includes the sensing and analyzing of the dynamic characteristics of the tactile information, which is referred as "*active*" touch. Let us consider, for example, the case when a robot reaches an object from behind, for example a cup, and the robot should immediately search for the handle. While searching for the handle, the area of the robot's skin that could be nearer to the handle should have higher spatial resolution than the rest of the skin. Biological evidence of this dynamic spatial resolution comes with foveation in vision: a human eye can focus on a specific area in an image to have high resolution and less on the rest.

These demands can be fulfilled by considering the following research lines.

6.3.1 Multimodal and multi-contact control

One promising direction to take, in order to overcome the difficulty of controlling robots using tactile sensors, is a multisensory approach. Rather than considering tactile sensors in isolation, it is possible to take advantage of other types of sensors that may exist in robots such as joint torque sensors and external vision. This is not a new idea since Mario Prats [Prats 13] combined signals of tactile sensing arrays, forces, and vision in robot arm controllers in various tasks, including the opening of doors by rotating and pushing the door knobs. However, there are still a lot of white spots in the domain of multimodal control. If we take, for example, the tactile features derived from the distribution of the points of contact and the image features, there are no common features that can be used by robots for smooth transition from the non-touch to touch based motion control. Therefore, finding these **common features** in vision and tactile contact images is a promising direction in human-robot interactions and co-manipulation tasks. Not only different sensing modalities should be combined, but also several tactile sensors should be considered together by further developing and implementing the TaLaM approach described in Chapter 5.

6.3.2 Vibrotactile control

The sensor noise, which was caused by the pressure variations of each element in the tactile sensing array, was filtered with a low pass filter. However, vibrations can be very useful in two aspects. The first one is well known and was implemented previously: the vibrations were measured in order to prevent a slip of a grasped object or in order to recognize a texture on its surface. The second aspect was proposed in the current thesis but is not developed yet: the velocity of the end-effector and the traveled distance during the sliding motion can be controlled based purely on vibrations. There should be a correlation between the rate of change of the velocity and of the frequencies that occur during the slipping motion. Fig. 6.1a shows the experimental setup for validating the

pressure sensor measurements with a source of vibrations with known frequency: the Shadow Robot hand was holding a vibrator with a two-fingertip grasp. The motor was fed with a pure sine-wave signal with a frequency of 25 Hz. The raw pressure values and their spectrum are given in Fig. 6.1b and c, respectively. The extracted harmonics show that the vibration sensors on the fingertips of the robot (the BioTac sensors) can be used to detect the oscillations that occur at the points of contact.

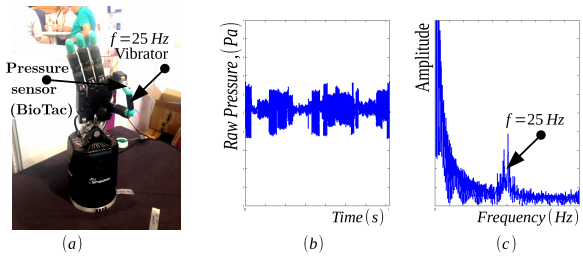


Figure 6.1: Experiments with vibrations: (a) the Shadow Robot hand with BioTac sensors holding a vibrating motor. (b) Raw pressure values , and (c) the spectrum of the sensed vibrations.

The index finger of the hand was then sliding over a textured surface (Fig. 6.2 a) at two different velocities v_1 and $v_2 > v_1$. The frequency responses of the two sliding motions were different as depicted in the Fig. 6.2b and c for the v_1 and v_2, respectively. Therefore, the finger could be set in motion and its velocity modified afterwards based on the caused vibrations in order perform touch-based motion control.

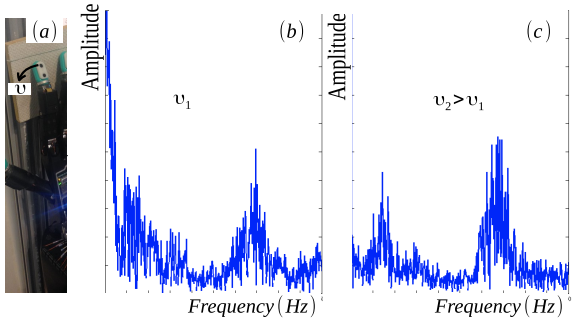


Figure 6.2: Vibrations at two different velocities of sliding over a textured surface. (a) The experimental setup, the spectrum (b) at the lower velocity and (c) at the higher velocity.

6.3.3 Extension to other human-robot interaction applications: hand-overs

One of the applications that has not been considered in the human-robot co-manipulation applications of this thesis is human-robot handovers. Figure 6.3 shows how the developed tactile servoing framework can be used for this application: by aligning the robot palm with an object while grasping it[1].

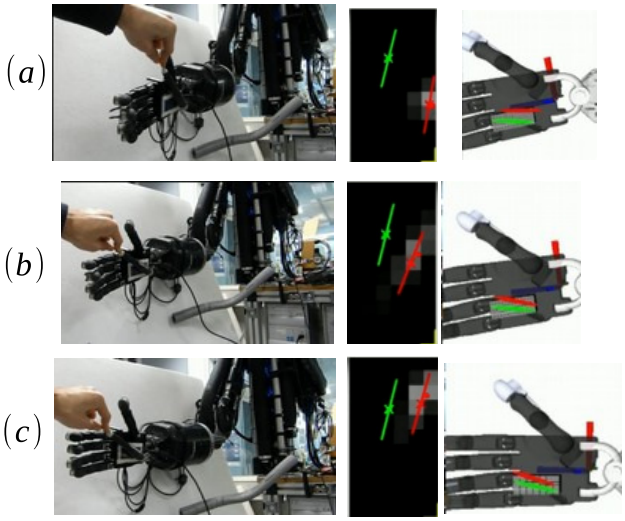


Figure 6.3: Hand-over task. a - contact detected. b - begin alignment. c - ready to grasp.

6.4 Summary

This research has attempted to improve robotic performance in the domain of physical interaction control using tactile sensing. We have shown that during physical interactions between the environment and a robot end-effector equipped with a planar pressure sensing array, the number of degrees-of-freedom, which can be artificially constrained in order to drive a robot manipulator in a desired manner, increased in the presence of a mechanically compliant contact surface. Thus, a compliant surface of tactile sensing arrays has been shown to be a useful property not only to compensate imprecisions but also to control contact frames during the physical interactions. By the use of an active touch-dependent Tactile Inverse Jacobian, the same tactile servoing controller has been applied

[1] https://www.youtube.com/watch?v=7vGKX-QqYec&list=UUQEGPPSY-gQjDm9TGokCDuA

to perform different tasks, including exploration, manipulation, co-manipulation, in a continuous sequence regardless a type of a contact. The developed tactile servoing framework can be implemented in robots to allow them to perform new tasks in industrial applications, as for example, cleaning of the internal chamber of big metal 3D printers from toxic metal powder, which is nowadays performed by human operators.

Bibliography

[AD 14] **AD**. *AD7147 technical datasheet*. http://www.analog.com/static/imported-files/Data_Sheets/AD7147.pdf, 2014. Accessed May 9, 2014. (Cited on page 186.)

[Agravante 14] **D. J. Agravante**, **A. Cherubini**, **A. Bussy**, **P. Gergondet et A. Kheddar**. *Collaborative human-humanoid carrying using vision and haptic sensing*. In 2014 IEEE International Conference on Robotics and Automation (ICRA), pages 607–612, May 2014. (Cited on page 129.)

[AirMuscle 16] **AirMuscle**. *AirMuscle*. https://www.shadowrobot.com/products/air-muscles/, 2016. Accessed May 9, 2016. (Cited on page 70.)

[Allen 88] **Peter K. Allen**. *Integrating vision and touch for object recognition tasks*. The International Journal of Robotics Research, vol. 7, n° 6, pages 15–33, 1988. (Cited on page 53.)

[ATi 14] **ATi**. *F/T Sensor: Nano17*. http://www.ati-ia.com/products/ft/ft_models.aspx?id=Nano17, 2014. Accessed May 9, 2014. (Cited on pages 137 and 185.)

[Barret 14] **Barret**. *Barret Hand*. http://www.barrett.com/robot/products-hand.htm, 2014. Accessed May 12, 2014. (Cited on pages 137, 139, and 140.)

[Bekiroglu 11] **Yasemin Bekiroglu**, **J. Laaksonen**, **Jimmy Alison Jorgensen**, **V. Kyrki et D. Kragic**. *Assessing Grasp Stability Based on Learning and Haptic Data*. Robotics, IEEE Transactions on, vol. 27, n° 3, pages 616–629, June 2011. (Cited on pages 89 and 194.)

[Berger 88] **Alan D. Berger**. *On Using a Tactile Sensor for Real-Time Feature Extraction*. Thèse de doctorat, Carnegie-Mellon University, 1988. (Cited on page 101.)

[Berger 91] **Alan D. Berger et Pradeep K. Khosla**. *Using Tactile Data for Real-Time Feedback*. The International Journal of Robotics Research, vol. 10, n° 2, pages 88–102, 1991. (Cited on pages xiii, 15, 16, 19, 26, 48, 53, 54, 93, and 94.)

[Bicchi 99] **Antonio Bicchi**, **Alessia Marigo et Domenico Prattichizzo**. *Dexterity through rolling: Manipulation of unknown objects*. In Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on, volume 2, pages 1583–1588. IEEE, 1999. (Cited on pages 116 and 117.)

[Bicchi 00] **A. Bicchi**. *Hands for dexterous manipulation and robust grasping: a difficult road toward simplicity*. Robotics and Automation, IEEE Transactions on, vol. 16, n° 6, pages 652–662, Dec 2000. (Cited on page 150.)

[Briot 79] **Maurice Briot et others**. *The utilization of an artificial skin sensor for the identification of solid objects*. In 9th Int. Symp. on Industrial Robots, pages 13–15, 1979. (Cited on page 15.)

[Bruyninckx 03] **Herman Bruyninckx**, **Peter Soetens et Bob Koninckx**. *The Real-Time Motion Control Core of the Orocos Project*. In IEEE International Conference on Robotics and Automation, pages 2766–2771, 2003. (Cited on page 196.)

[Büscher 15] **Gereon H. Büscher**, **Risto Kiva**, **Carsten Schürmann**, **Robert Haschke et Helge J. Ritter**. *Flexible and stretchable fabric-based tactile sensor*. Robotics and Autonomous Systems, vol. 63, pages 244–252, janvier 2015. (Cited on pages xxi, 22, 137, 141, and 184.)

[Chen 95] **Ning Chen**, **Hong Zhang et R. Rink**. *Edge tracking using tactile servo*. In Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on, volume 2, pages 84–89. IEEE, 1995. (Cited on pages 16, 48, 53, 55, and 93.)

[Chen 97] **N. N. Chen**, **Hong Zhang et Raymond E. Rink**. *Touch-driven robot control using a tactile Jacobian*. In Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on, volume 2, pages 1737–1742. IEEE, 1997. (Cited on pages 116 and 117.)

[Choi 06] **Byungjune Choi**, **Sanghun Lee**, **Hyouk Ryeol Choi et Sungchul Kang**. *Development of Anthropomorphic Robot Hand with Tactile Sensor : SKKU Hand II*. In Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, pages 3779–3784, Oct 2006. (Cited on pages xix, 137, 140, 141, and 192.)

[Chuang 13] **Cheng-Hsin Chuang**, **Ming-Shyan Wang**, **Yuan-Chu Yu**, **Chien-Liang Mu**, **Kuen-Feng Lu et Chin-Ting Lin**. *Flexible tactile sensor for the grasping control of robot fingers*. In Advanced Robotics and Intelligent Systems (ARIS), 2013 International Conference on, pages 141–146, May 2013. (Cited on pages xxi and 187.)

[Company 04] **Shadow Robot Company**. *Developments in dextrous hands for advanced robotic applications*. In Automation Congress, 2004. Proceedings. World, volume 15, pages 123–128, June 2004. (Cited on pages 136, 138, 141, and 188.)

[Corrales 13] **Juan Antonio Corrales**, **Véronique Perdereau et Fernando Torres Medina**. *Multi-fingered robotic hand planner for object reconfiguration through a rolling contact evolution model*. In 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, May 6-10, 2013, pages 625–630, 2013. (Cited on pages xix, xxii, 140, 141, 180, 196, and 207.)

[Cutkosky 08] **MarkR. Cutkosky**, **RobertD. Howe et WilliamR. Provancher**. *Force and Tactile Sensors*. In **Bruno Siciliano et Oussama Khatib**, éditeurs, Springer Handbook of Robotics, pages 455–476. Springer Berlin Heidelberg, 2008. (Cited on pages 180, 185, and 187.)

[Cutkosky 14] **Mark R. Cutkosky et John Ulmen**. *Dynamic Tactile Sensing*. In **Ravi Balasubramanian et Veronica J. Santos**, éditeurs, The Human Hand as an Inspiration for Robot Hand Development, volume 95 of *Springer Tracts in Advanced Robotics*, pages 219–246. Springer-Verlag, 2014. (Cited on pages xxi, xxii, 9, 13, 48, 137, 181, 182, 184, 186, and 195.)

[Dahiya 08] **Ravinder S Dahiya et Maurizio Valle**. *Tactile sensing for robotic applications*. Sensors, Focus on Tactile, Force and Stress Sensors, pages 298–304, 2008. (Cited on pages 184 and 187.)

[Dahiya 10] **R.S. Dahiya, G. Metta, M. Valle et G. Sandini**. *Tactile Sensing — From Humans to Humanoids*. Robotics, IEEE Transactions on, vol. 26, n° 1, pages 1–20, Feb 2010. (Cited on pages 9, 179, 180, 181, 186, and 187.)

[Dahiya 13a] **RavinderS. Dahiya et Maurizio Valle**. *Tactile Sensing Technologies*. In Robotic Tactile Sensing, pages 79–136. Springer Netherlands, 2013. (Cited on pages 179, 183, and 189.)

[Dahiya 13b] **R.S. Dahiya, P. Mittendorfer, M. Valle, G. Cheng et V.J. Lumelsky**. *Directions Toward Effective Utilization of Tactile Skin: A Review*. Sensors Journal, IEEE, vol. 13, n° 11, pages 4121–4138, Nov 2013. (Cited on pages 179, 182, and 196.)

[Dang 14] **Hao Dang et PeterK. Allen**. *Stable grasping under pose uncertainty using tactile feedback*. Autonomous Robots, vol. 36, n° 4, pages 309–330, 2014. (Cited on page 179.)

[Diftler 03] **M.A. Diftler, C.J. Culbert, R.O. Ambrose, Jr. Platt R. et W.J. Bluethmann**. *Evolution of the NASA/DARPA Robonaut control system*. In Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on, volume 2, pages 2543–2548 vol.2, Sept 2003. (Cited on pages 136 and 141.)

[Drimus 14] **Alin Drimus, Gert Kootstra, Arne Bilberg et Danica Kragic**. *Design of a flexible tactile sensor for classification of rigid and deformable objects*. Robotics and Autonomous Systems, vol. 62, n° 1, pages 3 – 15, 2014. New Boundaries of Robotics. (Cited on pages xvii, xxi, xxii, 110, 111, 113, 137, 139, 184, 194, 195, and 207.)

[Eeonyx 14] **Eeonyx**. *Piezoresistive Fabric Sensors*. http://www.eeonyx.com/eeontex.php, 2014. Accessed April 29, 2014. (Cited on pages 137 and 184.)

[Electronics 13] **Interlink Electronics**. *FSR*. http://www.interlinkelectronics.com/fsrtech.php, 2013. Accessed April 29, 2014. (Cited on pages 136 and 184.)

[Ellekilde 10] **Lars-Peter Ellekilde et Jimmy A. Jorgensen**. *RobWork: A Flexible Toolbox for Robotics Research and Education*. Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK), pages 1 –7, june 2010. (Cited on page 37.)

[Fearing 85] **Ronald S. Fearing et John M. Hollerbach**. *Basic solid mechanics for tactile sensing*. The International journal of robotics research, vol. 4, n° 3, pages 40–54, 1985. (Cited on pages 15 and 30.)

[Fishel 08] **J.A. Fishel**, **V.J. Santos et G.E. Loeb**. *A robust micro-vibration sensor for biomimetic fingertips*. In Biomedical Robotics and Biomechatronics, 2008. BioRob 2008. 2nd IEEE RAS EMBS International Conference on, pages 659–663, Oct 2008. (Cited on pages xxi, 10, 138, 190, and 191.)

[Fishel 12] **Jeremy A. Fishel et Gerald E. Loeb**. *Bayesian exploration for intelligent identification of textures*. Frontiers in Neurorobotics, vol. 6, n° 4, 2012. (Cited on pages 14, 179, and 181.)

[Fraden 04] **Jacob Fraden**. Handbook of modern sensors: physics, designs, and applications. Springer, 2004. (Cited on pages xxi, 183, 185, 186, 187, 189, and 190.)

[Francomano 13] **M.T. Francomano**, **D. Accoto et E. Guglielmelli**. *Artificial Sense of Slip – A Review*. Sensors Journal, IEEE, vol. 13, n° 7, pages 2489–2498, July 2013. (Cited on pages 179 and 181.)

[Freescale 14] **Freescale**. *Miniature I2C Digital Barometer*. http://cache.freescale.com/files/sensors/doc/data_sheet/MPL115A2.pdf, 2014. Accessed May 12, 2014. (Cited on page 196.)

[Fukui 11] **Wataru Fukui**, **Futoshi Kobayashi**, **Fumio Kojima**, **Hiroyuki Nakamoto**, **Nobuaki Imamura**, **Tadashi Maeda et Hidenori Shirasawa**. *High-Speed Tactile Sensing for Array-Type Tactile Sensor and Object Manipulation Based on Tactile Information*. Journal of Robotics, vol. 2011, 2011. (Cited on pages xix, xxi, 137, 139, 180, and 184.)

[Gaiser 08] **I. Gaiser**, **S. Schulz**, **A. Kargov**, **H. Klosek**, **A. Bierbaum**, **C. Pylatiuk**, **R. Oberle**, **T. Werner**, **T. Asfour**, **G. Bretthauer et R. Dillmann**. *A new anthropomorphic robotic hand*. In Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on, pages 418–422, Dec 2008. (Cited on page 137.)

[Gates 07] **Bill Gates**. *A robot in every home*. Scientific American, vol. 296, n° 1, pages 58–65, 2007. (Cited on page 9.)

[Gibson 77] **James J Gibson et Robert Shaw**. *Perceiving, acting, and knowing: Toward an ecological psychology*. The Theory of Affordances, pages 67–82, 1977. (Cited on page 92.)

[Göger 09] **D. Göger**, **N. Gorges et Heinz Worn**. *Tactile sensing for an anthropomorphic robotic hand: Hardware and signal processing*. In Robotics and Automation, 2009. ICRA '09. IEEE International Conference on, pages 895–901, May 2009. (Cited on pages xi, xvii, xix, xxi, 11, 13, 14, 110, 111, 137, 140, 142, 179, 187, 192, 196, and 207.)

[Gonzalez 14] **Franck Gonzalez**, **Florian Gosselin et Wael Bachta**. *Analysis of Hand Contact Areas and Interaction Capabilities During Manipulation and Exploration*. IEEE Transactions on Haptics, vol. 7, n° 4, pages 415–429, octobre 2014. (Cited on page 146.)

[Grebenstein 12] **Markus Grebenstein**, **Maxime Chalon**, **Werner Friedl**, **Sami Haddadin**, **Thomas Wimbck**, **Gerd Hirzinger et Roland Siegwart**. *The hand of the DLR Hand Arm System: Designed for interaction*. The International Journal of Robotics Research, vol. 31, n° 13, pages 1531–1555, 2012. (Cited on pages 138 and 140.)

[Gunji 08] **D. Gunji**, **Y. Mizoguchi**, **S. Teshigawara**, **Aiguo Ming**, **A. Namiki**, **M. Ishikawaand et M. Shimojo**. *Grasping force control of multi-fingered robot hand based on slip detection using tactile sensor*. In Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, pages 2605–2610, May 2008. (Cited on page 180.)

[Habib 14] **Ahsan Habib**, **Isura Ranatunga**, **Kyle Shook et Dan O. Popa**. *SkinSim: A simulation environment for multimodal robot skin*. In Automation Science and Engineering (CASE), 2014 IEEE International Conference on, pages 1226–1231. IEEE, 2014. (Cited on pages 37 and 38.)

[Harmon 82] **Leon D. Harmon**. *Automated Tactile Sensing*. The International Journal of Robotics Research, vol. 1, n° 2, pages 3–32, 1982. (Cited on pages 179 and 180.)

[Hasegawa 10] **H. Hasegawa**, **Y. Mizoguchi**, **K. Tadakuma**, **Aiguo Ming**, **M. Ishikawa et M. Shimojo**. *Development of intelligent robot hand using proximity, contact and slip sensing*. In Robotics and Automation (ICRA), 2010 IEEE International Conference on, pages 777–784, May 2010. (Cited on pages xxii, 138, 142, 192, and 193.)

[Heyneman 12] **B. Heyneman et M.R. Cutkosky**. *Biologically inspired tactile classification of object-hand and object-world interactions*. In Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on, pages 167–173, Dec 2012. (Cited on pages xxii, 186, 205, and 206.)

[Ho 12] **Van Anh Ho**, **T. Nagatani**, **A. Noda et S. Hirai**. *What can be inferred from a tactile arrayed sensor in autonomous in-hand manipulation?* In Automation Science and Engineering (CASE), 2012 IEEE International Conference on, pages 461–468, Aug 2012. (Cited on pages 17, 20, and 81.)

[Hogan 84] **Neville Hogan**. *Impedance control: An approach to manipulation*. In American Control Conference, 1984, pages 304–313. IEEE, 1984. (Cited on page 52.)

[Honeywell 14] **Honeywell**. *Bridge Pressure Sensor*. http://sccatalog.honeywell.com/pdbdownload/images/26pc.smt.series.chart.1.pdf, 2014. Accessed May 12, 2014. (Cited on page 190.)

[Hosoda 06] **Koh Hosoda**, **Yasunori Tada et Minoru Asada**. *Anthropomorphic robotic soft fingertip with randomly distributed receptors*. Robotics and Autonomous Systems, vol. 54, n° 2, pages 104–109, 2006. (Cited on page 192.)

[Howe 94] **R.D. Howe**. *Tactile Sensing and Control of Robotic Manipualtion*. Journal of Advanced Robotic, vol. 8, n° 3, pages 245–261, 1994. (Cited on pages 13 and 15.)

[Inaba Rubber Company 14] **Ltd. Inaba Rubber Company**. *Conductive rubber*. http://www.inaba-rubber.co.jp/en/b_products/inastomer/index.html, 2014. Accessed April 29, 2014. (Cited on page 184.)

[iniLabs 15] **iniLabs**. *Dynamic Vision Sensors*. http://www.inilabs.com/buy/, 2015. Accessed May 9, 2015. (Cited on page 14.)

[Jara 14] **Carlos A. Jara**, **Jorge Pomares**, **Francisco A. Candelas et Fernando Torres**. *Control Framework for Dexterous Manipulation Using Dynamic Visual Servoing and Tactile Sensors Feedback*. Sensors, vol. 14, n° 1, pages 1787–1804, 2014. (Cited on pages xxii, 137, 186, and 206.)

[Jiang 12] **Liang-Ting Jiang et Joshua R. Smith**. *Seashell effect pretouch sensing for robotic grasping*. In Robotics and Automation (ICRA), 2012 IEEE International Conference on, pages 2851–2858. IEEE, 2012. (Cited on pages 138, 140, 141, and 193.)

[Joergensen 10] **Jimmy A. Joergensen**, **Lars-Peter Ellekilde et Henrik G. Petersen**. *RobWorkSim - an Open Simulator for Sensor based Grasping*. In Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK), pages 1–8, June 2010. (Cited on pages 37 and 38.)

[Johansson 09] **Roland S. Johansson et J. Randall Flanagan**. *Coding and use of tactile signals from the fingertips in object manipulation tasks*. Nature Reviews Neuroscience, vol. 10, n° 5, pages 345–359, mai 2009. (Cited on pages 13, 47, 146, 181, and 191.)

[Johnson 11] **Micah K. Johnson**, **Forrester Cole**, **Alvin Raj et Edward H. Adelson**. *Microgeometry Capture Using an Elastomeric Sensor*. ACM Trans. Graph., vol. 30, n° 4, pages 46:1–46:8, juillet 2011. (Cited on page 189.)

[Kampmann 14] **Peter Kampmann et Frank Kirchner**. *Integration of Fiber-Optic Sensor Arrays into a Multi-Modal Tactile Sensor Processing System for Robotic End-Effectors*. Sensors, vol. 14, n° 4, pages 6854–6876, 2014. (Cited on pages xix, xxii, 138, 139, 140, 189, 192, 193, and 207.)

[Kappassov 13] **Z. Kappassov**, **Y. Khassanov**, **A. Saudabayev**, **A. Shintemirov et H.A. Varol**. *Semi-anthropomorphic 3D printed multigrasp hand for industrial and service robots*. In Mechatronics and Automation (ICMA), 2013 IEEE International Conference on, pages 1697–1702, Aug 2013. (Cited on pages xxiii and 208.)

[Kappassov 15] **Zhanat Kappassov**, **Juan-Antonio Corrales et Vronique Perdereau**. *Tactile sensing in dexterous robot hands Review*. Robotics and Autonomous Systems, juillet 2015. (Cited on page 91.)

[Kawamura 13] **Takuya Kawamura**, **Naoto Inaguma**, **Ko Nejigane**, **Kazuo Tani et Hironao Yamada**. *Measurement of Slip, Force and Deformation Using Hybrid Tactile Sensor System for Robot Hand Gripping an Object*. International Journal of Advanced Robotic Systems, vol. 10, 2013. (Cited on page 192.)

[kew Lee 08] **Hyung kew Lee**, **Jaehoon Chung**, **Sun-Il Chang et Euisik Yoon**. *Normal and Shear Force Measurement Using a Flexible Polymer Tactile Sensor With Embedded Multiple Capacitors*. Microelectromechanical Systems, Journal of, vol. 17, n° 4, pages 934–942, Aug 2008. (Cited on page 185.)

[Koike 11] **M. Koike**, **S. Saga**, **T. Okatani et K. Deguchi**. *Sensing method of total-internal-reflection-based tactile sensor*. In World Haptics Conference (WHC), 2011 IEEE, pages 615–619, June 2011. (Cited on page 189.)

[Koiva 13] **R. Koiva**, **M. Zenker**, **C. Schurmann**, **R. Haschke et H.J. Ritter**. *A highly sensitive 3D-shaped tactile sensor*. In Advanced Intelligent Mechatronics (AIM), 2013 IEEE/ASME International Conference on, pages 1084–1089, July 2013. (Cited on pages xix, xxi, xxii, 11, 15, 137, 140, 141, 180, 182, 184, 205, and 207.)

[Koyama 13] **K. Koyama**, **H. Hasegawa**, **Y. Suzuki**, **Aiguo Ming et M. Shimojo**. *Pre-shaping for various objects by the robot hand equipped with resistor network structure proximity sensors*. In Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, pages 4027–4033, Nov 2013. (Cited on page 140.)

[Kyberd 92] **Peter J. Kyberd et Paul H. Chappell**. *Object-slip detection during manipulation using a derived force vector*. Mechatronics, vol. 2, n° 1, pages 1 – 13, 1992. (Cited on pages 14, 137, and 181.)

[Kyberd 98] **Peter J. Kyberd**, **Mervyn Evans et Stefan te Winkel**. *An Intelligent Anthropomorphic Hand, with Automatic Grasp*. Robotica, vol. 16, pages 531–536, 9 1998. (Cited on pages 10, 138, 142, 180, 181, and 193.)

[Lawitzky 10] **Martin Lawitzky**, **Alexander Mrtl et Sandra Hirche**. *Load sharing in human-robot cooperative manipulation*. In 19th International Symposium in Robot and Human Interactive Communication, pages 185–191. IEEE, 2010. (Cited on page 129.)

[Lee 99] **M.H Lee et H.R Nicholls**. *Review Article Tactile sensing for mechatronicsa state of the art survey*. Mechatronics, vol. 9, n° 1, pages 1 – 31, 1999. (Cited on pages 9 and 179.)

[Li 13a] **Qiang Li**, **Carsten Schürmann**, **Robert Haschke et Helge Ritter**. *A control framework for tactile servoing*. In Robotics: Science and Systems, 2013. (Cited on pages xvi, 18, 20, 53, 55, 93, 94, 96, 97, 101, 104, 180, 182, 185, 188, 196, and 206.)

[Li 13b] **Rui Li et E.H. Adelson**. *Sensing and Recognizing Surface Textures Using a GelSight Sensor*. In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pages 1241–1247, June 2013. (Cited on pages xxi, 189, and 190.)

[Li 15] **Qiang Li**, **Robert Haschke et Helge Ritter**. *A visuo-tactile control framework for manipulation and exploration of unknown objects*. In Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on, pages 610–615. IEEE, 2015. (Cited on pages 116 and 117.)

[Liu 12a] **Hongbin Liu**, **J. Greco**, **Xiaojing Song**, **J. Bimbo**, **L. Seneviratne et K. Althoefer**. *Tactile image based contact shape recognition using neural network*. In Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on, pages 138–143, Sept 2012. (Cited on pages xvii, xxii, 18, 110, 111, 137, 194, 205, and 206.)

[Liu 12b] **Hongbin Liu**, **Xiaojing Song**, **Joao Bimbo**, **Kaspar Althoefer et Lakmal Senerivatne**. *Intelligent Fingertip Sensing for Contact Information Identification*. In **Jian S Dai**, **Matteo Zoppi et Xianwen Kong**, éditeurs, Advances in Reconfigurable Mechanisms and Robots I, pages 599–608. Springer London, 2012. (Cited on pages xi, 10, 11, and 179.)

[Liu 12c] **Hongbin Liu**, **Xiaojing Song**, **T. Nanayakkara**, **Lakmal D Seneviratne et K. Althoefer**. *A computationally fast algorithm for local contact shape and pose classification using a tactile array sensor*. In Robotics and Automation (ICRA), 2012 IEEE International Conference on, pages 1410–1415, May 2012. (Cited on pages xvii, 110, and 111.)

[Liu 15] **Hongbin Liu**, **KienCuong Nguyen**, **Vronique Perdereau**, **Joao Bimbo**, **Junghwan Back**, **Matthew Godden**, **LakmalD. Seneviratne et Kaspar Althoefer**. *Finger contact sensing and the application in dexterous hand manipulation*. Autonomous Robots, pages 1–17, 2015. (Cited on pages 140, 141, and 205.)

[LTD. 15] **OptoForce LTD.** *Opto-Force sensor*. http://www.optoforce.com/3dsensor/, 2015. Accessed June 10, 2015. (Cited on pages 138, 140, 141, and 189.)

[Maiolino 13] **P. Maiolino**, **M. Maggiali**, **G. Cannata**, **G. Metta et L. Natale**. *A Flexible and Robust Large Scale Capacitive Tactile System for Robots*. Sensors Journal, IEEE, vol. 13, n° 10, pages 3910–3917, Oct 2013. (Cited on pages 129, 141, and 186.)

[Martin 04] **T. B. Martin**, **R.O. Ambrose**, **M.A. Diftler**, **Jr. Platt R. et M. J. Butzer**. *Tactile gloves for autonomous grasping with the NASA/DARPA Robonaut*. In Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on, volume 2, pages 1713–1718 Vol.2, April 2004. (Cited on pages 138, 141, and 188.)

[Martinez-Hernandez 13] **U. Martinez-Hernandez**, **T.J. Dodd**, **L. Natale**, **G. Metta**, **T.J. Prescott et N.F. Lepora**. *Active contour following to explore object shape with robot touch*. In World Haptics Conference (WHC), 2013, pages 341–346, April 2013. (Cited on pages 89 and 179.)

[Mason 81] **Matthew T. Mason**. *Compliance and Force Control for Computer Controlled Manipulators*. Systems, Man and Cybernetics, IEEE Transactions on, vol. 11, n° 6, pages 418–432, June 1981. (Cited on page 66.)

[Meguenani 15] **Anis Meguenani**, **Vincent Padois et Philippe Bidaud**. *Control of robots sharing their workspace with humans: an energetic approach to safety*. In Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, pages 4678–4684. IEEE, 2015. (Cited on page 129.)

[Metta 06] **Giorgio Metta**, **Paul Fitzpatrick et Lorenzo Natale**. *YARP: yet another robot platform*. International Journal on Advanced Robotics Systems, vol. 3, n° 1, pages 43–48, 2006. (Cited on page 196.)

[Metta 08] **Giorgio Metta**, **Giulio Sandini**, **David Vernon**, **Lorenzo Natale et Francesco Nori**. *The iCub Humanoid Robot: An Open Platform for Research in Embodied Cognition*. In Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems, PerMIS '08, pages 50–56, New York, NY, USA, 2008. ACM. (Cited on pages 141 and 196.)

[Mittendorfer 11] **P. Mittendorfer et G. Cheng**. *Humanoid Multimodal Tactile-Sensing Modules*. Robotics, IEEE Transactions on, vol. 27, n° 3, pages 401–410, June 2011. (Cited on page 13.)

[Moisio 13] **Sami Moisio**, **Beatriz Len**, **Pasi Korkealaakso et Antonio Morales**. *Model of tactile sensors using soft contacts and its application in robot grasping simulation*. Robotics and Autonomous Systems, vol. 61, n° 1, pages 1–12, janvier 2013. (Cited on pages 37 and 38.)

[Montana 88] **David J Montana**. *The kinematics of contact and grasp*. The International Journal of Robotics Research, vol. 7, n° 3, pages 17–32, 1988. (Cited on pages 15 and 53.)

[Mouri 02] **Tetsuya Mouri**, **Haruhisa Kawasaki**, **Keisuke Yoshikawa**, **Jun Takai et Satoshi Ito**. *Anthropomorphic Robot Hand: Gifu Hand III*. In Proc. Int. Conf. ICCAS, pages 1288–1293, 2002. (Cited on pages 137 and 141.)

[Muthukrishnan 87] **C. Muthukrishnan**, **D. Smith**, **D. Myers**, **J. Rebman et A. Koivo**. *Edge detection in tactile images*. In Robotics and Automation. Proceedings. 1987 IEEE International Conference on, volume 4, pages 1500–1505, Mar 1987. (Cited on pages 16 and 19.)

[Mser 09] **Michael Mser**. *Structure-borne sound*. In Engineering Acoustics, pages 117–142. Springer Berlin Heidelberg, 2009. (Cited on page 192.)

[Namiki 03] **A. Namiki**, **Y. Imai**, **M. Ishikawa et M. Kaneko**. *Development of a high-speed multifingered hand system and its application to catching*. In Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on, volume 3, pages 2666–2671 vol.3, Oct 2003. (Cited on pages xix, 137, 139, and 196.)

[Navarro 13] **S.E. Navarro**, **M. Marufo**, **Yitao Ding**, **S. Puls**, **D. Goger**, **B. Hein et H. Worn**. *Methods for safe human-robot-interaction using capacitive tactile proximity sensors*. In Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, pages 1149–1154, Nov 2013. (Cited on page 179.)

[Newell 01] **Fiona N. Newell**, **Marc O. Ernst**, **Bosco S. Tjan et Heinrich H. Blthoff**. *Viewpoint dependence in visual and haptic object recognition*. Psychological science, vol. 12, n° 1, pages 37–42, 2001. (Cited on page 2.)

[Nguyen 13a] **Kien-Cuong Nguyen**. *Multi-modal control of an anthropomorphic hand robot for the task of grasping and dexterous manipulation*. PhD Thesis, 2013. Accessed April 4, 2013. (Cited on pages xx and 149.)

[Nguyen 13b] **Kien-Cuong Nguyen et V. Perdereau**. *Fingertip force control based on max torque adjustment for dexterous manipulation of an anthropomorphic hand*. In Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, pages 3557–3563, Nov 2013. (Cited on page 179.)

[Nicosia 01] **Salvatore Nicosia**. Ramsete: Articulated and mobile robotics for services and technology, volume 270. Springer, 2001. (Cited on pages 13 and 89.)

[Odhner 14] **Lael U Odhner**, **Leif P Jentoft**, **Mark R Claffee**, **Nicholas Corson**, **Yaroslav Tenzer**, **Raymond R Ma**, **Martin Buehler**, **Robert Kohout**, **Robert D Howe et Aaron M Dollar**. *A compliant, underactuated hand for robust manipulation*. The International Journal of Robotics Research, 2014. (Cited on pages xix, xxi, 138, 139, 182, 191, and 196.)

[Overton 81] **Kenneth J. Overton et Thomas Williams**. *Tactile Sensation for Robots*. In Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'81, pages 791–795, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc. (Cited on page 15.)

[Peratech 14] **Peratech**. *Quantum Tunneling Composite*. http://www.peratech.com, 2014. Accessed May 9, 2014. (Cited on pages 138 and 188.)

[Perdereau 93] **Vronique Perdereau et Michel Drouin**. *A new scheme for hybrid force-position control*. In **A. Morecki, G. Bianchi et K. Jaworek**, éditeurs, RoManSy 9, volume 187 of *Lecture Notes in Control and Information Sciences*, pages 150–159. Springer Berlin Heidelberg, janvier 1993. (Cited on page 200.)

[Perdereau 96] **Vronique Perdereau et Michel Drouin**. *Hybrid external control for two robot coordinated motion*. Robotica, vol. 14, n° 02, page 141, mars 1996. (Cited on pages 52 and 65.)

[Petrovskaya 06] **Anna Petrovskaya, Oussama Khatib, Sebastian Thrun et Andrew Y. Ng**. *Bayesian estimation for autonomous object manipulation based on tactile sensors*. In Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on, pages 707–714. IEEE, 2006. (Cited on page 150.)

[Pezzementi 10] **Z. Pezzementi, E. Jantho, L. Estrade et G.D. Hager**. *Characterization and simulation of tactile sensors*. In Haptics Symposium, 2010 IEEE, pages 199–205, March 2010. (Cited on pages 37 and 38.)

[Pezzementi 11] **Z. Pezzementi, E. Plaku, C. Reyda et G.D. Hager**. *Tactile-Object Recognition From Appearance Information*. Robotics, IEEE Transactions on, vol. 27, n° 3, pages 473–487, June 2011. (Cited on pages xvii, 13, 89, 110, 111, and 181.)

[Platkiewicz 16] **Jonathan Platkiewicz, Hod Lipson et Vincent Hayward**. *Haptic Edge Detection Through Shear*. Scientific Reports, vol. 6, page 23551, mars 2016. (Cited on page 15.)

[PPS 14] **PPS**. *Tactile Sensors*. http://www.pressureprofile.com/products.php, 2014. Accessed May 9, 2014. (Cited on pages xx, 137, 141, 144, and 186.)

[Prats 13] **Mario Prats, Angel P. del Pobil et Pedro J. Sanz**. *Robot Physical Interaction through the combination of Vision, Tactile and Force Feedback*. In **Bruno Siciliano et Oussama Khatib**, éditeurs, Tracts in Advanced Robotics, volume 84 of *Springer Tracts in Advanced Robotics*, page 177. Springer, 2013. (Cited on pages 159 and 204.)

[Quigley 09] **Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler et Andrew Y. Ng**. *ROS: an open-source Robot Operating System*. In ICRA workshop on open source software, volume 3, page 5, 2009. (Cited on page 196.)

[Raibert 81] **Marc H Raibert et John J Craig**. *Hybrid position/force control of manipulators*. Journal of Dynamic Systems, Measurement, and Control, vol. 103, n° 2, pages 126–133, 1981. (Cited on pages 52 and 66.)

[Righetti 14] **Ludovic Righetti, Mrinal Kalakrishnan, Peter Pastor, Jonathan Binney, Jonathan Kelly, RandolphC. Voorhies, GauravS. Sukhatme et Stefan Schaal**. *An autonomous manipulation system based on force control and optimization*. Autonomous Robots, vol. 36, n° 1-2, pages 11–30, 2014. (Cited on page 10.)

[Robles-De-La-Torre 01] **Gabriel Robles-De-La-Torre et Vincent Hayward**. *Force can overcome object geometry in the perception of shape through active touch*. Nature, vol. 412, n° 6845, pages 445–448, juillet 2001. (Cited on pages 114 and 128.)

[Robotics 14] **Weiss Robotics**. *Tactile Sensors*. http://weiss-robotics.de/en/tactile-sensors.html, 2014. Accessed April 4, 2014. (Cited on pages 69, 137, 139, 141, 184, 185, and 203.)

[Robotiq 14] **Robotiq**. *3-Finger Adaptive Robot Gripper*. http://robotiq.com/en/products/industrial-robot-hand, 2014. Accessed May 12, 2014. (Cited on pages 129 and 137.)

[Rogers 10] **John A. Rogers, Takao Someya et Yonggang Huang**. *Materials and Mechanics for Stretchable Electronics*. Science, vol. 327, n° 5973, pages 1603–1607, 2010. (Cited on pages xxi and 184.)

[Romano 11] **J.M. Romano, K. Hsiao, G. Niemeyer, S. Chitta et K.J. Kuchenbecker**. *Human-Inspired Robotic Grasp Control With Tactile Sensing*. Robotics, IEEE Transactions on, vol. 27, n° 6, pages 1067–1079, Dec 2011. (Cited on pages 13, 137, 138, 142, 181, 186, and 192.)

[Russell 90] **R.A. Russell**. Robot tactile sensing. Prentice Hall, 1990. (Cited on page 183.)

[Schmidt 06] **Peer A. Schmidt, Eric Mal et Rolf P. Wrtz**. *A sensor for dynamic tactile information with applications in humanrobot interaction and object exploration*. Robotics and Autonomous Systems, vol. 54, n° 12, pages 1005 – 1014, 2006. (Cited on page 137.)

[Schmitz 10] **A. Schmitz, M. Maggiali, L. Natale, B. Bonino et G. Metta**. *A tactile sensor for the fingertips of the humanoid robot iCub*. In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pages 2212–2217, Oct 2010. (Cited on pages xxii, 137, 205, and 207.)

[Schmitz 11] **A. Schmitz, P. Maiolino, M. Maggiali, L. Natale, Giorgio Cannata et G. Metta**. *Methods and Technologies for the Implementation of Large-Scale Robot Tactile Sensors*. Robotics, IEEE Transactions on, vol. 27, n° 3, pages 389–400, June 2011. (Cited on pages xix, xxi, xxii, 11, 137, 139, 140, 182, 186, 195, 196, and 206.)

[Schneider 09] **A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt et W. Burgard**. *Object identification with tactile sensors using bag-of-features*. In Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, pages 243–248, Oct 2009. (Cited on pages xvii, 110, and 111.)

[Schreiber 10] **Gnter Schreiber, Andreas Stemmer et Rainer Bischoff**. *The fast research interface for the kuka lightweight robot*. In IEEE Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications How to Modify and Enhance Commercial Controllers (ICRA 2010), pages 15–21. Citeseer, 2010. (Cited on page 72.)

[Schulz 04] **S. Schulz, C. Pylatiuk, A. Kargov, R. Oberle et G. Bretthauer**. *Progress in the development of anthropomorphic fluidic hands for a humanoid robot*. In Humanoid Robots, 2004 4th IEEE/RAS International Conference on, volume 2, pages 566–575, Nov 2004. (Cited on pages 137, 141, 187, and 192.)

[Schunk 14a] **Schunk**. *2-Finger-Parallel Gripper*. http://www.schunk.com/schunk_files/attachments/OM_AU_PG__EN.pdf, 2014. Accessed April 29, 2014. (Cited on page 137.)

[Schunk 14b] **Schunk**. *3-Finger Gripping Hand sDH*. http://www.schunk.com/schunk_files/attachments/SDH_DE_EN.pdf, 2014. Accessed April 29, 2014. (Cited on pages 137 and 139.)

[Sekitani 10] **Tsuyoshi Sekitani**, **Ute Zschieschang**, **Hagen Klauk et Takao Someya**. *Flexible organic transistors and circuits with extreme bending stability*. Nature materials, vol. 9, n° 12, pages 1015–1022, 2010. (Cited on page 13.)

[Seminara 11] **Lucia Seminara**, **Marco Capurro**, **Paolo Cirillo**, **Giorgio Cannata et Maurizio Valle**. *Electromechanical characterization of piezoelectric {PVDF} polymer films for tactile sensors in robotics applications*. Sensors and Actuators A: Physical, vol. 169, n° 1, pages 49 – 58, 2011. (Cited on pages 186 and 187.)

[Shadowrobot 14] **Shadowrobot**. *Shadow Dexterous Hand*. http://www.shadowrobot.com/products/dexterous-hand/, 2014. Accessed May 12, 2014. (Cited on pages 137, 138, 141, 196, and 205.)

[Shimojo 04] **M. Shimojo**, **A. Namiki**, **M. Ishikawa**, **R. Makino et K. Mabuchi**. *A tactile sensor sheet using pressure conductive rubber with electrical-wires stitched method*. Sensors Journal, IEEE, vol. 4, n° 5, pages 589–596, Oct 2004. (Cited on page 184.)

[Siciliano 08] **Bruno Siciliano et Oussama Khatib**, éditeurs. Springer Handbook of Robotics. Springer, 2008. (Cited on pages 63 and 199.)

[Sikka 94a] **Pavan Sikka**, **Hong Zhang et Steve Sutphen**. *Tactile servo: Control of touch-driven robot motion*. In Experimental Robotics III, pages 219–233. Springer, 1994. (Cited on pages 48, 53, 54, 101, 116, and 117.)

[Sikka 94b] **Pavan Sikka**, **Hong Zhang et Steve Sutphen**. *Tactile servo: Control of touch-driven robot motion*. In Experimental Robotics III, pages 219–233. Springer, 1994. (Cited on page 51.)

[Simlab 14] **Simlab**. *Allegro-Hand*. http://www.simlab.co.kr/Allegro-Hand.htm, 2014. Accessed August 4, 2014. (Cited on page 136.)

[Smartec 16] **Smartec**. *Universal Transducer Interface*. http://www.smartec-sensors.com/en/products/uti-interface-en.html, 2016. Accessed May 9, 2016. (Cited on page 145.)

[Someya 14] **T. Someya et T. Sekitani**. *Bionic skins using flexible organic devices*. In Micro Electro Mechanical Systems (MEMS), 2014 IEEE 27th International Conference on, pages 68–71, Jan 2014. (Cited on pages 89 and 185.)

[Song 14] **Xiaojing Song**, **Hongbin Liu**, **K. Althoefer**, **T. Nanayakkara et L.D. Seneviratne**. *Efficient Break-Away Friction Ratio and Slip Prediction Based on Haptic Surface Exploration*. Robotics, IEEE Transactions on, vol. 30, n° 1, pages 203–219, Feb 2014. (Cited on page 179.)

[Stassi 14] **Stefano Stassi**, **Valentina Cauda**, **Giancarlo Canavese et Candido Fabrizio Pirri**. *Flexible Tactile Sensing Based on Piezoresistive Composites: A Review*. Sensors, vol. 14, n° 3, pages 5296–5332, 2014. (Cited on page 183.)

[Strohmayr 13] **M.W. Strohmayr et D. Schneider**. *The DLR artificial skin step II: Scalability as a prerequisite for whole-body covers*. In Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, pages 4721–4728, Nov 2013. (Cited on pages 138 and 140.)

[Stckler 11] **Jrg Stckler et Sven Behnke**. *Following human guidance to cooperatively carry a large object*. In Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on, pages 218–223. IEEE, 2011. (Cited on page 129.)

[SynTouch 13] **SynTouch**. *The BioTac*. http://www.syntouchllc.com/Products/BioTac/, 2013. Accessed May 12, 2014. (Cited on pages xxii, 138, 141, 190, and 193.)

[TakkTile 13] **TakkTile**. *Takktile kit*. http://www.takktile.com, 2013. Accessed Feb. 10, 2014. (Cited on pages xxi, 138, 139, 141, and 191.)

[Tekscan 13] **Tekscan**. *Flexiforce*. http://www.tekscan.com/flexiforce.html, 2013. Accessed April 29, 2014. (Cited on pages 137, 141, and 184.)

[Tenzer d] **Yaroslav Tenzer**, **Leif P Jentoft et Robert D Howe**. *Inexpensive and Easily Customized Tactile Array Sensors using MEMS Barometers Chips*. IEEE Robotics and Automation Magazine, (conditionally accepted). (Cited on page 191.)

[Teshigawara 09] **S. Teshigawara**, **S. Shimizu**, **K. Tadakuma**, **M. Aiguo**, **M. Shimojo et M. Ishikawa**. *High sensitivity slip sensor using pressure conductive rubber*. In Sensors, 2009 IEEE, pages 988–991, Oct 2009. (Cited on pages xxi and 184.)

[Teshigawara 11] **S. Teshigawara**, **T. Tsutsumi**, **S. Shimizu**, **Y. Suzuki**, **Aiguo Ming**, **M. Ishikawa et M. Shimojo**. *Highly sensitive sensor for detection of initial slip and its application in a multi-fingered robot hand*. In Robotics and Automation (ICRA), 2011 IEEE International Conference on, pages 1097–1102, May 2011. (Cited on pages xxii, 14, 137, 139, 181, 184, 195, and 196.)

[Tiwana 12] **Mohsin I. Tiwana**, **Stephen J. Redmond et Nigel H. Lovell**. *A review of tactile sensing technologies with applications in biomedical engineering*. Sensors and Actuators A: Physical, vol. 179, n° 0, pages 17 – 31, 2012. (Cited on pages 179 and 189.)

[Twendy-one 14] **Twendy-one**. *Twendy-one robot hand*. http://twendyone.com, 2014. Accessed August 4, 2014. (Cited on page 136.)

[Vukobratovi 90] **Miomir Vukobratovi**, **Branislav Borovac**, **Duan Surla et Dragan Stoki**. Biped Locomotion. Springer Berlin Heidelberg, Berlin, Heidelberg, 1990. (Cited on page 30.)

[Weiss 05] **K. Weiss et Heinz Worn**. *The working principle of resistive tactile sensor cells*. In Mechatronics and Automation, 2005 IEEE International Conference, volume 1, pages 471–476 Vol. 1, July 2005. (Cited on pages 137, 184, and 194.)

[Wettels 09] **N. Wettels**, **A.R. Parnandi**, **Ji-Hyun Moon**, **G.E. Loeb et G. Sukhatme**. *Grip Control Using Biomimetic Tactile Sensing Systems*. Mechatronics, IEEE/ASME Transactions on, vol. 14, n° 6, pages 718–723, Dec 2009. (Cited on page 191.)

[Wettels 14] **Nicholas Wettels**, **JeremyA. Fishel et GeraldE. Loeb**. *Multimodal Tactile Sensor*. In **Ravi Balasubramanian et Veronica J. Santos**, éditeurs, The Human Hand as an Inspiration for Robot Hand Development, volume 95 of *Springer Tracts in Advanced Robotics*, pages 405–429. Springer International Publishing, 2014. (Cited on pages 10, 89, 180, 190, 192, and 196.)

[Xie 14] **Hui Xie**, **A. Jiang**, **H.A. Wurdemann**, **Hongbin Liu**, **L.D. Seneviratne et K. Althoefer**. *Magnetic Resonance-Compatible Tactile Force Sensor Using Fiber Optics and Vision Sensor*. Sensors Journal, IEEE, vol. 14, n° 3, pages 829–838, March 2014. (Cited on pages xi, xxi, 11, 138, 189, and 190.)

[Xu 13] **Danfei Xu**, **G.E. Loeb et J.A. Fishel**. *Tactile identification of objects using Bayesian exploration*. In Robotics and Automation (ICRA), 2013 IEEE International Conference on, pages 3056–3061, May 2013. (Cited on pages xix, xxii, 13, 138, 140, 141, 182, and 207.)

[Yussof 09] **Hanafiah Yussof**, **Masahiro Ohka**, **Hirofumi Suzuki et Nobuyuki Morisawa**. *Tactile Sensing-based Control System for Dexterous Robot Manipulation*. In **Sio-Iong Ao**, **Burghard Rieger et Su-Shing Chen**, éditeurs, Advances in Computational Algorithms and Data Analysis, volume 14 of *Lecture Notes in Electrical Engineering*, pages 199–213. Springer Netherlands, 2009. (Cited on pages xxi, 138, 181, 189, and 190.)

[Zhang 00] **Hong Zhang et N.N. Chen**. *Control of contact via tactile sensing*. Robotics and Automation, IEEE Transactions on, vol. 16, n° 5, pages 482–495, Oct 2000. (Cited on pages xi, 17, 20, 21, 53, 55, 93, and 94.)

[Zhang 13] **Ting Zhang**, **Hong Liu**, **Li Jiang**, **Shaowei Fan et Jing Yang**. *Development of a Flexible 3-D Tactile Sensor System for Anthropomorphic Artificial Hand*. Sensors Journal, IEEE, vol. 13, n° 2, pages 510–518, Feb 2013. (Cited on pages xxi and 188.)

Appendix A
Principle Component Analysis

The acquired contact image $I(x, y)$ is binarized with an appropriate threshold, which results in contact points without the intensity information. The orientation of the edge is given by the principle component representing the direction, in which the contact points are spread the most. This principal component U_p is equal to the eigenvector with the highest eigenvalue of the covariance matrix Q:

$$Q(x, y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n - 1} \tag{A.1}$$

where n is the rank of $I(x, y)$, $\bar{x}$ and $\bar{y}$ are the mean values of x and y. The angle of rotation of the base vector of U in y (or x) direction, U_y (or U_x) around the z axis is given by:

$$Orientation_z = atan2(U_y, e_x) \tag{A.2}$$

where e_x is the base vector in x direction of the sensor frame $e(x, y)$. The derived angle of the rotation around the orthonormal direction to the sensor frame represents the orientation of the edge w.r.t the sensor frame.

Appendix B

Tactile sensing technologies

Tactile sensors can be defined as a tool that can evaluate a given property of an object through physical contact between the hand and the object [Lee 99]. When a tactile sensor is represented by an array, each sensing element of the sensor is referred differently in robotics literature, e.g. sensing cell, taxel or tactel.

Tactile sensors meet the following task-related requirements of in-hand manipulation [Dahiya 13a]:

1) Response. In collision avoidance [Navarro 13] and human-robot interaction tasks, tactile sensors must provide information about the presence of contact and measure the strength of contact force, respectively.

2) Exploration. During exploration, tactile sensors should provide information about: surface properties from measurements of a texture, hardness, and temperature [Fishel 12]; structural properties from shape [Martinez-Hernandez 13]; and functional properties from detection of contacts and vibrations [Göger 09].

3) Manipulation. In autonomous manipulation tasks, tactile data is used as a control parameter in: slip detection; estimation of grasp stability[Dang 14]; contact point estimation, surface normal and curvature measurement [Liu 12b]; tangential and normal forces measurements for achieving stable grasps [Song 14]; and contact force measurements for fingertip force control [Nguyen 13b].

Depending on the task, the sensor has different design specifications, which were first determined by Harmon [Harmon 82]. The basic design criteria for tactile sensors have been previously reported in [Dahiya 10] for humanoid robots, in [Tiwana 12] for biomedical engineering, in [Francomano 13] for prosthetic hands, and in [Dahiya 13b]

for manufacturing and large tactile system implementation. In autonomous manipulation applications, tactile sensors meet requirements for object characterization and identification (e.g. they estimate the compliance, thermal and textural properties) and for manipulation (e.g. they control the force applied to the object) [Wettels 14].

The most important design criteria for tactile sensors with application in manipulation tasks are summarized in Table B.1 and discussed in following:

1) Requirements on **spatial resolution** of a tactile sensing array depends on both the size of the objects to be recognized and the location of the sensor on a robot hand. A rather high spatial resolution is desirable in in-hand object manipulation [Corrales 13] or tactile servoing [Li 13a] tasks, whereas in the cases when high sensitivity or high frequency response are desirable, e.g. reactive force control [Kyberd 98], the spatial resolution is limited by for the following reasons. A higher spatial resolution unavoidably leads to a longer acquisition time [Cutkosky 08], a larger number of wire connections and a stronger sensitivity to external electromagnetic noises. The first two consequences are straightforward, high resolution requires a large number of sensing cells, which in turn causes longer processing time. These sensing cells also require more wire connections. The highest limit of sensitivity is given by the minimum detectable variation of the measured signal. As sensing cells become smaller the sensitivity to external electromagnetic noises and crosstalk increases. Thus, the sensitivity degrades because the level of noise can become comparable with the signal. By considering these pros and cons, the requirements on spatial resolution can vary for different parts of a robot hand. It was previously investigated that the resolution on the fingertips should be as high as $1mm$ since the fingertips are mostly involved in fine manipulation [Harmon 82]. In the current state of the art, fingertip tactile sensors integrated with robot hands have a spatial resolution of around $5mm$ [Koiva 13], [Fukui 11]. On less sensitive parts of a robot hand like the palm, the spatial resolution decreases up to $5mm$ as stated in [Dahiya 10]. Requirements for spatial resolution can be omitted when only slippage is of importance, e.g. automatic grasping using vibrations to achieve stable grasp [Kyberd 98] and slip-detection with center-of-pressure tactile sensor [Gunji 08].

2) **Sensitivity** in the tactile sensors is given by the smallest detectable variation in

pressure/force. A small detectable variation means a high sensitivity. High sensitivity is very important in manipulation tasks with fragile and deformable objects as in [Yussof 09] or [Romano 11]. However, the range from the minimum to maximum detectable pressure/force, i.e. dynamic range, shrinks with the increase of the sensitivity of a tactile sensor, which is caused by the technology used in the structure of the current sensors. An area of sensing cells the sensor also causes contradiction between sensitivity and spatial resolution as was discussed above.

Dahiya et al. [Dahiya 10] impose following requirements. The sensitivity on the fingertips should be not less than $1mN$, while a dynamic range of 1000 : 1 is desirable.

3) Requirements for **frequency response** highly depend on the application. In general, tactile sensors can be dynamic or static [Cutkosky 14]. If the hand is required to detect vibrations during slippage, the frequency response should be as high as the vibration frequencies occurring during a slippage [Francomano 13], [Kyberd 98], [Romano 11], [Teshigawara 11]. In human hands, the detectable vibration frequencies vary from $5Hz$ to $50Hz$ and from $40Hz$ to $400Hz$ for different afferents [Johansson 09]. Thus the frequency response of a dynamic tactile sensor should be at least $400Hz$, i.e. the sampling rate must be at least $800Hz$ according to NyquistShannon sampling theorem. When only spatial resolution is of importance (e.g. tactile object recognition [Pezzementi 11]), then the frequency response is not restricted by the response time. On the contrary, when measurements of vibrations are used to prevent a slippage [Kyberd 92], to detect a contact of a grasped object with an environment [Romano 11] or to recognize a texture of a surface [Fishel 12], then the response time of a sensor becomes crucial. The frequency response (bandwidth) is limited by the softness (elasticity) of a tactile sensor. The use of soft materials, that are used to increase surface friction, causes phase delay in propagation of the waves of the mechanical vibrations that occur at the point of contact.

4) **Hysteresis** and memory effect ideally should be as low as possible. Tactile sensing arrays incorporating flexible foam in their structure it unavoidably leads to an elastic behavior of the sensors. Once the sensor is pressed and released, the flexible foam first compresses and then regains its form but not immediately (hysteresis effect) and sometimes not to the previous shape (memory effect). Moreover, the sensor could be covered by a

soft material, e.g. silicon rubber as in [Odhner 14]. The advantage of using flexible materials is the increase of a contact friction. However, sensitivity and frequency response of a sensor may degrade with the increase of flexibility. Though, reading devices can have high sampling rate, a sensor may have significant hysteresis, which reduces dynamic response [Cutkosky 14]. The memory effect could be avoided by use of a thinner foam, which in turn decreases the dynamic range, since the maximum charge (in capacitive sensors) that can be stored is proportional to the thickness of the foam. This maximum charge represents the largest detectable force.

5) **Wiring** of tactile sensors should not affect the workspace of robot hands [Dahiya 13b]. Integration of a high number of tactile sensors in the robot hand is challenging due to wiring constraints. As an example, in [Xu 13] a multimodal tactile sensor is installed as a complete fingertip with bulky backside instead of distal and middle phalanges. Shielding and smart wiring should guarantee minimum sensitivity to noise and minimum tactile cross-talk. Use of serial communication protocol decreases the number of connection wires as in iCub skin [Schmitz 11], but it increases the sampling rate.

6) A sensor itself should be **flexible** so it can be attached to any type of robot hand [Schmitz 11], unless the sensor is designed as a complete part of a robot hand, as for example the 3D-shaped tactile sensing fingertip in [Koiva 13].

7) **Surface properties** of tactile sensors, such as mechanical compliance and surface friction coefficient should fit to various manipulation tasks. Elastic material with given friction coefficient and compliance can cover tactile sensors. If the contact sensing surface has very low friction, then the hand must apply high normal forces to keep the object stable, which can lead to breaking the object [Koiva 13]. However, the low friction of the sensor surface is needed in tactile exploration procedures [Li 13a].

8) A **robust** sensor design should guarantee that the sensor can withstand highly repetitive usage without its performance being affected. The sensor should endure normal as well as lateral forces.

Table B.1: Design criteria: pros and cons.

Criteria	pros	cons	Application
High spatial resolution	A smaller objects can be recognized and features with a higher precision can be extracted.	A smaller sensitivity and a longer processing time.	Contact pattern recognition, fine manipulation.
High sensitivity	Detection of a rather small change of a contact force.	Dynamic range of the sensor shrinks, spatial resolutions decreases.	Light touch detection and fragile object manipulation.
High frequency response	A rather fast response to the changes in the level of the contact force	Spatial resolution and dynamic range decrease	Detection of a slip and texture recognition.
Low hysteresis	High frequency response	Degrease of the sensor's surface friction and dynamic range.	Detection of a slip and texture recognition.
Low number of wire connections	The workspace of robot hands does not change.	Decrease of the frequency response (in case of using serial data communication).	Dexterous manipulation
High surface friction	Insuring stable grasp without applying high forces.	Imped tactile exploration procedure. Reduces the frequency response of the sensor (in case of using soft paddings).	Grasping

B.1 Tactile sensor types

Change of capacitance, resistance, optical distribution, electrical charge can be used in the sensing systems [Fraden 04], [Russell 90].In the robotics literature, these different ways to construct the sensing systems are referred as transduction of contact information [Dahiya 13a]. And the types of tactile sensors vary depending on the transduction.

In the following we describe the basic types of tactile sensors and their transduction methods. The advantages and disadvantages of each sensor type are given in Table B.2.

B.1.1 Piezoresistive sensors

The piezoresistive effect is a physical process during which electrical resistance changes when the material is mechanically deformed (Figure B.1(a)) [Fraden 04]. Materials possessing this effect are called piezoresistors [Stassi 14].

There are several technologies for artificial tactile sensing based on piezoresistive materials: Force Sensing Resistors (FSR), pressure-sensitive conductive rubber, piezoresis-

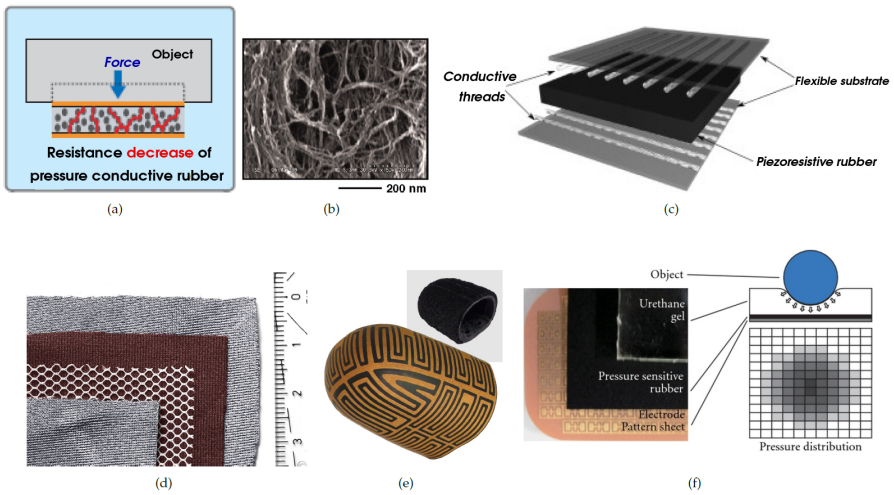


Figure B.1: Piezoresistive Tactile Sensor Arrays: (a) illustration of resistance changes in conductive rubber [Teshigawara 09],(b) nano-scale image of conductive rubber [Rogers 10], (c) structure of piezoresistive tactile array [Drimus 14], (d) piezoresistive fabric tactile sensor [Büscher 15], (e) schematic of electrode layer of the 3D-shaped tactile sensor [Koiva 13], (f) tactile image of a piezo-resistive pressure sensor array [Fukui 11].

tive foam, and piezoresistive fabric.The simplest way to incorporate tactile sensing via discrete components is by using FSRs [Cutkosky 14] and they are widely used in positioning devices such as joysticks [Dahiya 08]. Piezoresistive rubber is a composite material made by mixing non-conductive elastomer with homogeneously distributed electrically conductive carbon particles [Weiss 05], [Teshigawara 09]. Figure B.1(b) shows the structure of conductive rubber at nano-scale level [Rogers 10]. Sensors based on conductive rubber with multilayer structures as in [Drimus 14], [Teshigawara 11], [Fukui 11], [Büscher 15] (Figure B.1(c), B.1(d)) may suffer from delamination of top layers. This can be avoided by using a single layer of the conductive rubber with a stitched array of wires in orthogonal orientations as in [Shimojo 04]. Another method of designing tactile sensing arrays using the conductive rubbers incorporates a non-flexible pattern of the electrodes on one layer and piezoresistive rubber on a second layer (Figure B.1(e) [Koiva 13]. Some of the sensors and components are commercially available from Interlink [Electronics 13] and the Tekscan [Tekscan 13] (FSRs), Weiss Robotics [Robotics 14] (rigid tactile sensors based on carbon enriched silicone rubber), Inaraba [Inaba Rubber Company 14] (pressure conductive rubber), Eeonyx [Eeonyx 14] (piezoresistive fabric), ATi industrial

automation [ATi 14] (Force/Torque sensors).

It is worth mentioning that currently developed tactile sensors based on pressure sensitive rubber and organic transistors, such as the ones used in the bionic skin [Someya 14], are exceptionally thin and highly flexible. Conductive rubbers used in piezoresistive sensors have a nonlinear force-resistance characteristic (please refer to sensor calibration plot in [Li 13a]). As a consequence of using elastic materials, the sensors have severe hysteresis. The sensitivity in the piezoresistive sensors may decrease due to wearing and tearing off, since the resistance of the conductive rubber does not depend on deformation only but also on thickness. Moreover, materials used in the piezoresistive sensors could change their properties due to variation of the temperature and moistness [Fraden 04].

Piezoresistors also suffer from lower repeatability:after multiple deformations, an elastic material may never regain its initial form. Some of the piezoresistive sensing arrays are also fragile to shear forces, e.g. Weiss tactile sensors [Robotics 14]. In spite of these drawbacks, a number of robot hands incorporate piezoresistive tactile sensing arrays, since the sensors are relatively simple to manufacture, can be flexible and many commercial solutions exist. Compared to capacitive sensors that will be discussed in the next section, the piezoresistive sensors are more robust (not completely) to electromagnetic noises.

B.1.2 Capacitive sensors

Capacitive sensors consist of two conductive plates (Figure B.2(a)) separated by a compressible dielectric material (Figure B.2(b)). When the gap between plates changes under the applied forces, the capacitance is also changed. Besides normal forces, the shear forces can be calculated by the sensor with the use of embedded multiple capacitors [kew Lee 08]. Pressure sensing arrays can be constructed by overlapping row and column electrodes isolated from each other by elastic dielectric [Cutkosky 08]. Sensitivity to small forces can be achieved by using more compressible elastic materials or thin sensors. As a flexible foam between two plates gets thinner than a smaller charge in the sensor could be measured that is in turn means a higher sensitivity.

Capacitive technology is very popular among the sensing transducers and it has been

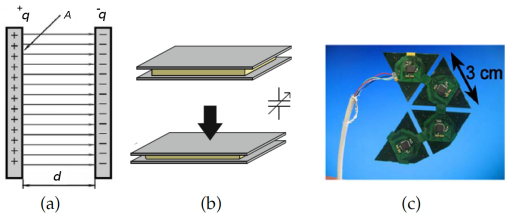


Figure B.2: Capacitive Tactile Sensing Technology: (a) capacitance of a parallel plate capacitor depends on distance between plates d and area of the plates A (q is the stored charge) [Fraden 04]; (b) two conductive plates are separated by an elastic dielectric – as force is applied, the distance between the plates reduces, changing the capacitance [Cutkosky 14]; (c) mesh of triangle shape capacitive sensors for the palm of the iCub humanoid robot [Schmitz 11].

widely used in robotic applications [Dahiya 10]: for example, in tactile the skin (Figure B.2(c)) for the iCub humanoid robot [Schmitz 11], in the PR2 robot grippers [Romano 11], with the multifingered "Allegro" robot hand [Jara 14], and with the Robotiq robot gripper [Heyneman 12].

There are commercial capacitive pressure sensing arrays such as "DigiTacts" from Pressure Profile Systems (PPS) [PPS 14] and capacitance-to-digital-converter (CDC) chips such as "AD7147" from *Analog Devices* [AD 14].

The major disadvantages of capacitive sensors are susceptibility to electro-magnetic noise, sensitivity to temperature, non-linear response (please refer to the plot with response of excited taxel in [Maiolino 13]), and hysteresis. Their advantages include a higher frequency response relatively to piezoresistive sensors. Since capacitive technologies are used in every day life applications, as for example touch screens, this type of tactile sensing have been well investigated and used in robotics and especially in robot hands.

B.1.3 Piezoelectric sensors

The piezoelectric effect (Figure B.3(a)) is described as electrical charge generation in the crystalline material due to deformation caused by applied force/pressure [Fraden 04]. The piezoelectric effect is produced in quartz crystals, as well as in human-made ceramics and polymers, such as polyvinylidene fluoride (PVDF) [Seminara 11]. A piezo-

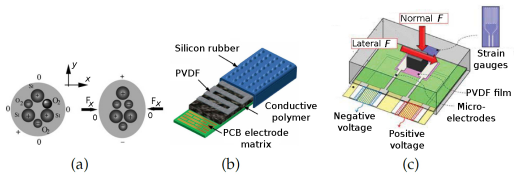


Figure B.3: Piezoelectric Tactile Sensing: (a) the piezoelectric effect – an applied force causes rearrangement of positive Si and negative O_2 particles leading to an increase of potential [Fraden 04]; (b) a tactile sensing array based on the piezoelectric effect with electrodes on the bottom layer, piezolectric material in the middle and rubber on the top [Göger 09], (c) schematic model of a piezoelectric sensing tactel [Chuang 13].

electric tactile sensor can be created with the PVDF film strips embedded into a rubber material. Piezoelectric materials, being restricted for dynamic measurements and used in ultrasonic-based sensors, are suitable for dynamic tactile sensing [Dahiya 10], [Cutkosky 08]. Among other piezoelectric materials, PVDF polymer has features such as flexibility and chemical stability, which makes it preferable for use in touch sensors. Seminara et al. [Seminara 11] conducted research on PVDF electro-mechanical design of tactile sensors with frequency range of 1 Hz to 1 kHz. Goger et al. [Göger 09] developed a combined dynamic/static tactile sensor (Figure B.3(b)) based on PVDF polymer and piezoresistive foam from Weiss Robotics for a fluidic robot hand [Schulz 04]. Chuang et al. [Chuang 13] developed a flexible tactile sensor based on piezoelectric film with structural electrodes for grasping an object of unknown weight (Figure B.3(c)).

Piezoelectric materials have high bandwidth up to $7kHz$ as reported in [Göger 09]. These materials have faster dynamic response than capacitive sensors. Their disadvantages include fragility of electrical junctions, temperature sensitivity [Dahiya 08] and they are suitable for dynamic measurements only.

B.1.4 Quantum Tunnel Effect Sensors

Quantum Tunnel Composite (QTC) sensors can change their properties from insulators to conductors under compression [Dahiya 10]. QTC sensors are more technologically advanced compared to piezoresistive and capacitive sensors. The metal particles in QTC get so close to each other that quantum tunneling (of electrons) takes place between the par-

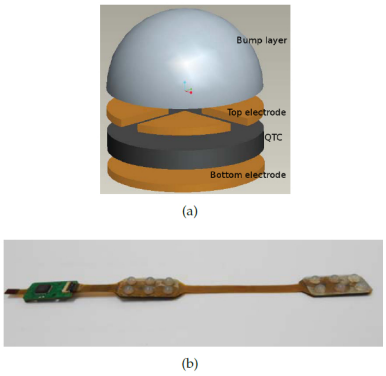


Figure B.4: Quantum Effect Tactile Sensing: (a) structure of a tactel of the QTC based tactile sensing array with capability of measuring shear and normal forces [Zhang 13]; (b) the flexible tactile sensing array for a finger of an anthropomorphic robot hand with the tactels that can measure shear forces [Zhang 13].

ticles. Using QTC material, Zhang et al. 2013 [Zhang 13] (Figure B.4) developed a flexible tactile sensor for an anthropomorphic artificial hand with capability of measuring shear and normal forces. The sensor has sensitivities of $0.45mV/mN$ in x- and y-directions and of $0.16mV/mN$ in z-directions, and dynamic ranges up to $8N$ in z- and y-directions and $20N$ in x-direction. QTC-based tactile sensors [Peratech 14] were integrated with previous versions of the Shadow robot hand [Company 04] and used in the tactile glove for the Robonaut hand [Martin 04]. The sensors have linear response (please refer to sensor outputs w.r.t normal force in [Zhang 13]) and a dynamic range starting from 0 to $22N$ which outperform the piezoresistive sensor with a maximum force of 5 N [Li 13a] in terms of the dynamic range. These sensors suffer from wear and tear of and, therefore, their sensitivity decreases as in the case of the piezoresistive sensors. To the best of our knowledge, for the tactile sensing materials within this category, there are no commercial products that are designed for use with robot hands.

B.1.5 Optical sensors

Optical sensing is based on optical reflection between mediums with different refractive indices. Conventional optical tactile sensors consist of an array of infrared light-emitting diodes (LEDs) and photo detectors (Figure B.5(a)). The intensity of the light is propor-

tional to the magnitude of the pressure [Fraden 04]. Optical sensors can also be made sensitive to shear forces, e.g. Yussof et al. [Yussof 09] developed an optical three-axis tactile sensor for the fingertips of a two-fingered hand (Figure B.5(b)). The sensor consists of 41 sensing elements made from silicon rubber, a light source, an optical fiber-scope, and a current charged coupled device (CCD) camera. With the optical tactile sensor, the hand is capable of manipulating a light paper box (Figure B.5(c)). Kampmann et al. [Kampmann 14] embedded fiber optic sensors to a multi-modal tactile measuring system of a three-fingered robot gripper (Figure B.7(d)). Xie et al. developed a flat 3x3 optical tactile sensor array (Figure B.5(d)) with elements of the sensor that are magnetic resonance compatible for use in Magnetic Resonance Imaging [Xie 14]. Johnson et al. [Johnson 11] proposed a novel "GelSight" tactile sensor to capture surface textures using an elastomer coated with a reflective membrane and a camera with resolution of up to 2 microns. (Figure B.5(e)). A fingertip with a "GelSight" (Figure B.5(f)) tactile sensor can measure the surface roughness and texture, the pressure distribution, and even a slip [Li 13b]. Another example of an optical tactile sensor with transparent elastomer material is presented in [Koike 11], where an LED and a photo-diode distant from each other are placed against a reflecting (contact) planar surface. When surface deforms it causes changes in reflected beams. Similar concept is used in the OptoForce sensors [LTD. 15]. These sensors are based on the use of infrared light to detect deformation of the contact surface, which in turn transforms to force. The forces in three dimensions are estimated from measurements of four photo-diodes that surround one infrared source. The reflecting surface has a semi-spherical shape.

Sensors within this category have good spatial resolution, sensitivity, high repeatability and immunity from electro-magnetic interference [Tiwana 12]. The disadvantages of these tactile sensors are their relatively big size, high-power consumption and high computational costs [Dahiya 13a].

B.1.6 Sensors based on barometric measurements

Tactile sensors within this group use pressure transducers that have been long used for measuring the pressure in liquids and air [Fraden 04]. Use of liquid inside a tactile sensor

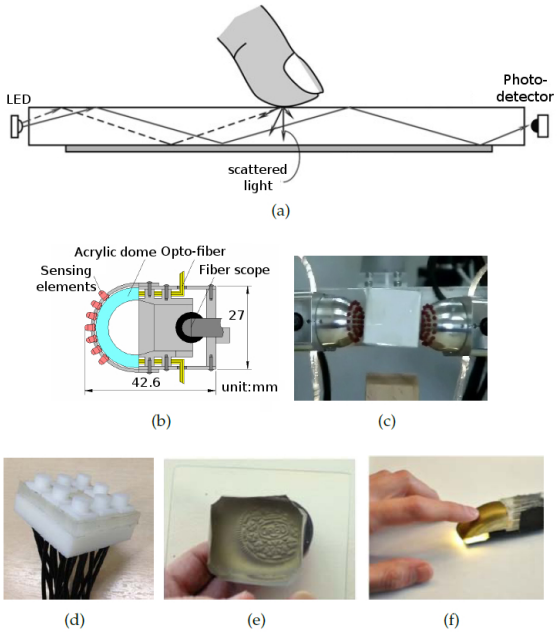


Figure B.5: Optical Tactile Sensors: (a) an optical tactile transducer based on the principle of frustrated total internal reflection [Fraden 04], (b) a structure of optical three-axis tactile sensor: a displacement of a sensing element fixed on flexible finger surface causes changes in light propagation in opto-fibers [Yussof 09], (c) fingers with the sensitive optical sensors manipulating a light paper box [Yussof 09], (d) photo of an optical 3 x 3 tactile array with magnetic field compatibility [Xie 14], (e) "GelSight" optical sensor consisting of a piece of clear elastomer coated with a reflective membrane senses the shape of the cookie surface [Li 13b], (f) finger configurations of the "GelSight" sensor [Li 13b].

allows getting high frequency response and deformability of the sensor at the same time. A liquid is used as propagation media for vibrations, which are represented by changes in pressure value. This approach takes advantage of the conventional pressure sensors, as for example the digital barometer [Honeywell 14]. Wettels et al. [Wettels 14] introduced the sensing system that incorporates electro-conductive fluid to produce both constant and dynamic signals (Figure B.6(d)). Micro-vibrations, caused by either motion over textured surface or slippage at any contact point, propagate as sound waves through the liquid media to a pressure transducer [Fishel 08]. The bandwidth of the sensor is 1 kHz, which makes the sensing system well suitable for slip detection applications. The sensor is embedded in the multi-modal biomimetic ®BioTac fingertip sensor from SynTouch LLC [SynTouch 13].

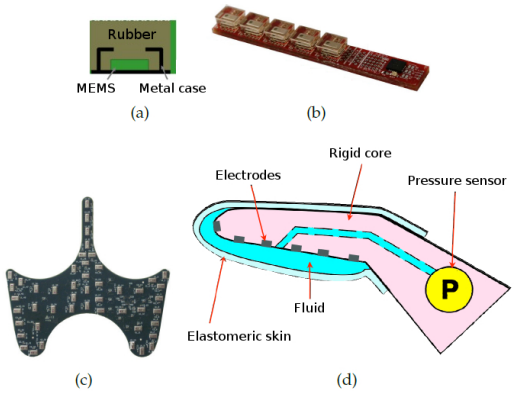


Figure B.6: Sensors based on barometric measurements: (a) the structure of a tactile sensing cell with a barometer and silicon rubber (b), the TakkStrip tactile array of these cells [TakkTile 13], (c) custom shaped array of the pressure sensing barometers of the iHY hand [Odhner 14], (d) micro-vibration sensing system based on a fluid pressure sensor of the BioTac tactile sensor [Fishel 08].

In [Tenzer d], no liquid is used as a propagation media, a barometer is instead molded within a silicon rubber in each tactel. The rubber acts as a membrane (Figure B.6(a)) . Once the rubber is deformed due to the contact with an environment, it causes changes in the pressure values of the barometer. Using the same digital barometer, Odhner et al. [Odhner 14] developed a tactile sensor array (Figure B.6(c)) with a spatial resolution of around $(3-5mm)$, sensitivity $1mN$, and dynamic range up to $4.9N$ for a three-fingered robot hand [Odhner 14].

Sensors involving liquid and barometers have high frequency response [Wettels 09]. Sensors with silicon rubber and barometer are low-cost, but has low frequency response [Tenzer d] as a result of the elasticity of the silicon rubber. Hence, the use of a liquid as a propagation media is more suitable when frequency response is of importance.

B.1.7 Multi-modal Tactile Sensors

To match the human hand's different types of tactile sensing modalities (thermal, fast adapting and slow adapting afferents) [Johansson 09] as close as possible, a robot hand should be equipped with multi-modal tactile sensors. Current multi-modal tactile sensing systems incorporate static pressure distribution arrays, dynamic tactile sensors, ther-

mal sensors, and proximity sensors. The BioTac finger-shaped sensor array (Figure B.7a,b) provides information about the contact forces, microvibrations, and temperature produced during contact with external objects [Wettels 14]. Some tactile sensors have the ability to sense dynamic and static contact forces since they have been constructed using a combination of piezoresitive and piezoelectric materials. Examples of such material include piezoresistive rubber with PVDF (Figure B.3(b)) [Göger 09] that is integrated with an anthropomorphic fluidic hand [Schulz 04] and pressure variable resistor ink with PVDF that is integrated with a four fingered robot hand [Choi 06]. Another hybrid sensing system with similar combination of dynamic and static transducers combines carbon micro-coil touch sensor and a force tactile sensor [Kawamura 13]. Hasegawa et al. integrated proximity and pressure sensors on the fingertip (Figure B.7(c)) to enhance autonomous grasping [Hasegawa 10]. Optical sensors also found their application in the multi-modal approach. A three-fingered robot gripper described in [Kampmann 14] incorporates optical sensors and combines measurements of absolute forces by strain gauge sensors, dynamic forces by piezoelectric sensors, and force distribution by fiber optic sensors, as shown in Figure B.7(d). Unlike the above multimodal sensors, in which locations of sensing units are known, a sensing system of a robot fingertip proposed by Hosoda et al. [Hosoda 06] has random distribution of the sensing units. Similar to [Göger 09], the sensing system consists of piezoresistive and piezoelectric sensors to measure static forces and vibrations. The piezoelectric sensors are placed at a skin layer and inside the fingertip thus giving possibility to measure internal vibrations. The only drawback of the multimodal tactile sensors is their size.

B.1.8 Structure-borne sound tactile sensors

Vibrations and waves in solid structures are summarized by the term ”Structure-borne sound” [Mser 09]. In manipulation tasks, structure born sounds occur at the initial contact of a manipulated object with the environment or during the slippage. Accelerometers and microphones can be used as detecting devices. In pick-and-place manipulation tasks, these structure born sounds can serve as indicators to trigger the placement of the object by the manipulator. Romano et al. [Romano 11] use a high sensitive 3-axis accelerometer

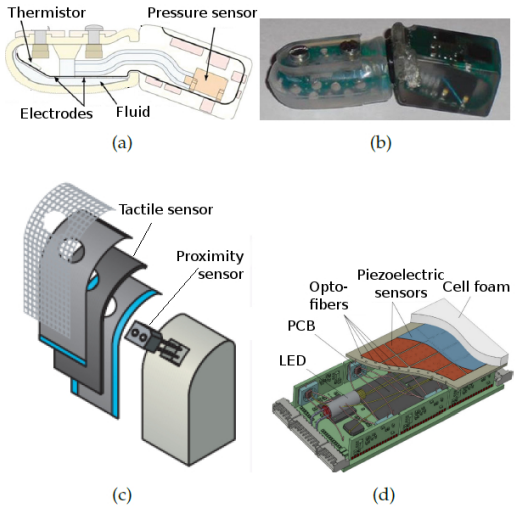


Figure B.7: Multimodal Tactile Sensors: (a) schematic of the biomimetic BioTac tactile sensor with 19 electrodes, fluid pressure sensor and thermometer [SynTouch 13], (b) photo of the multimodal BioTac tactile sensor, (c) combined tactile-proximity sensor that can measure both the distance to an object and the contact pressure [Hasegawa 10], (d) drawing of a multi-modal tactile sensing module consisting of optical and piezoresistive sensors [Kampmann 14].

in the base of PR2 robot gripper in order to detect the contact of the object with the table and to release the object. Earlier, Kyberd et al. [Kyberd 98] integrated a microphone with an anthropomorphic prosthetic hand for automated grasping.

Sensors within this group have wide bandwidth, but are suitable for dynamic measurements only. However, in a close proximity of an object, it is possible to estimate the distance to the object by comparing a level of an environmental acoustic noise and a level of noise within the sensor as has been shown by Jiang et al. [Jiang 12]. The presented concept of the sensor is based on Seashell Effect - increase of a level of noise in cavities due to resonance of sound waves and intrinsic resonance frequency of a cave. The sensor incorporates a cavity and a microphone located inside the cavity. The cavity has its own resonance frequency that depends on both the structure of the cavity and the distance from an object to be grasped.

Data stream coming from tactile sensors has different physical meanings for different transduction technologies. In general it can be dynamic or static according to the time response and may represent an array of data, vector or scalar value. Hence, data

Table B.2: Tactile sensing types: advantages and disadvantages of major sensor types. Abbreviations for the names: PRes. - piezoresistive sensors, Cap. - capacitive, PEl. - piezoelectric sensor, Opt. - optical sensors, BarS. - sensors based on barometric measurements, MultiM. - multimodal sensors, SoundS. - structure borne sound sensors.

Type	Advantages	Disadvantages
PRes.	Many commercial solutions exist, simpler for manufacturing, can be flexible.	Non-linear response, temperature and moistness dependence, fatigue, permanent deformation, hysteresis
Cap.	A number of commercial solutions, can be flexible, may have higher bandwidth than PRes.	Susceptibility to electro-magnetic noise, sensitivity to temperature, non-linear response, hysteresis.
PEl.	Very high bandwidth	Temperature dependence, dynamic sensing only
QTC	Linear response, higher dynamic range (w.r.t Cap. and PRes)	More complex for manufacturing (w.r.t in Cap. and PRes)
Opt.	High spatial reso- lution, high sensitivity, repeatability, immunity to EM noise	Bulky, high-power consumption, high computational costs
BarS. (fluid)	High bandwidth, high sensitivity, temperature and moistness independence	Low spatial resolution
SoundS.	High bandwidth	Dynamic sensing only

acquisition from different sensors has its own approaches as discussed in the following section.

B.2 Tactile data acquisition

In capacitive and piezoresistive sensors, data from each tactel can be acquired either directly, which means that a high amount of wires is required, or by using a multiplexing circuit (Figure F.1(c)), which decreases twofold the number of wire connection.

Piezoresistive tactile arrays consist of a common electrode, of sensing electrodes that are arranged as a matrix, and of conductive rubber in between. Pressing on the sensor's surface provides an image of the applied pressure profile [Weiss 05]. Figure B.1(f) illustrates the image of the sensing array which is produced when a spherical object is pressing the tactile surface. Tactile images can be used for contact pattern recognition [Liu 12a], grasp stability estimation [Bekiroglu 11], object classification [Drimus 14], and

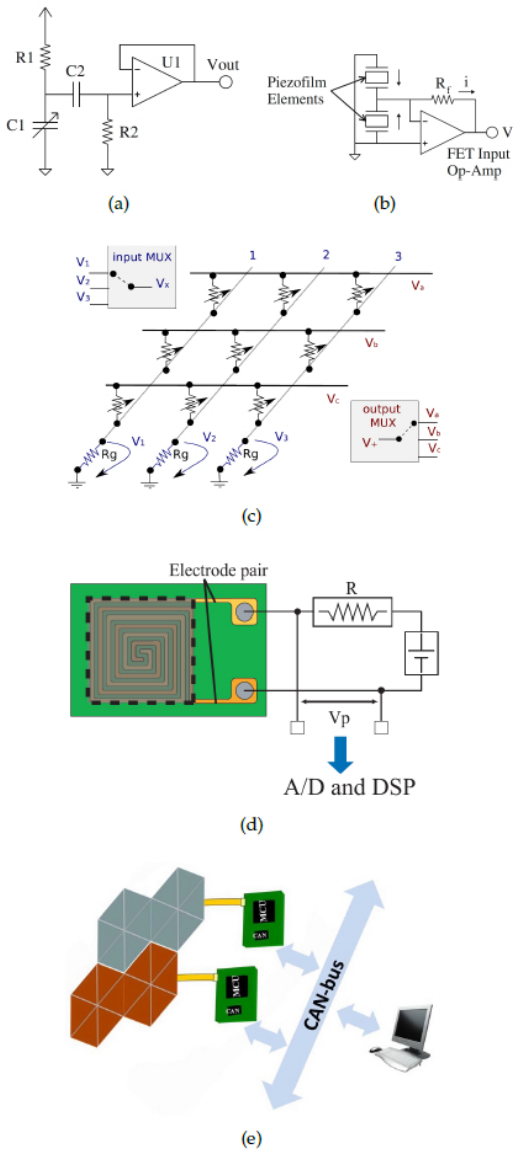


Figure B.8: Tactile sensing reading circuits: (a) the condenser microphone circuit for capacitive sensors [Cutkosky 14]; (b) a circuit for utilizing piezoelectric PVDF film as a stress rate sensor [Cutkosky 14]; (c) signal conditioning and voltage multiplexing for a 3 x 3 tactile sensing array [Drimus 14]; (d) the voltage-divider circuit for a pressure conductive rubber [Teshigawara 11]; (e) network structure of the iCub tactile sensing skin using CAN-bus for connecting tactile sensing patches, 12 tactels in each patch, with a main processing unit [Schmitz 11].

tactile servoing [Li 13a].

For tactile sensing arrays, data acquisition involves the usage of analog to digital converters (Figure B.8(d)) as well as of microprocessing units for polling each tactel [Dahiya 13b]. The capacitance of capacitive tactile sensors can be measured by commercial CDCs chips. The CDC chips can include I^2C serial interface. Digital barometers such as the absolute digital pressure sensor "MPL115A" [Freescale 14] that is used in the iHY robot hand [Odhner 14] has also I^2C serial interface. Communication with processing units can be realized via different transmission protocols (e.g. controller area network (CAN), and universal serial bus (USB), RS232). In iCub skin [Schmitz 11], local measurements are sent by an on-board processing unit over a CAN bus (Figure B.8(e)). Multimodal tactile sensing data in the BioTac sensor [Wettels 14] is acquired by the PIC microprocessor and sent to the host processing unit over serial peripheral interface (SPI). In order to minimize memory use of micro-processing units, data coming from sensors can be preprocessed by signal conditioning circuits, which can be implemented as system on chip (SOC) or system in package (SIP) [Dahiya 13b].

In some specific applications, for example in fast reaction to slip [Göger 09], signals from tactile sensors can be analyzed and processed within a controller without sending information to the host computer. In most of the applications, middleware and high-level software installed onto the main processing unit is used to compute acquired data and control the system. For these purposes versatile open source and commercial robot control platforms are available: in [Corrales 13], robot operating system (ROS) [Quigley 09] is used to control Shadow robot hand [Shadowrobot 14], robot platform (YARP) [Metta 06] is used to control iCub humanoid robot [Metta 08]; the controlling operating system dSPACE from dSPACE Co. is used in [Teshigawara 11] to control high speed-robot hand [Namiki 03] in real-time and C++ libraries of open Robot control software (OROCOS) [Bruyninckx 03]. Among open source robot control platforms, ROS is the most widely used and supports both simulation (Gazebo simulator) and control of the Shadow hand, the Barret hand and many other manipulators and robots.

Appendix C

Contact frame transformations

In order to incorporate control signals expressed in the sensing frame, they should be spatially transformed (Fig. C.1).

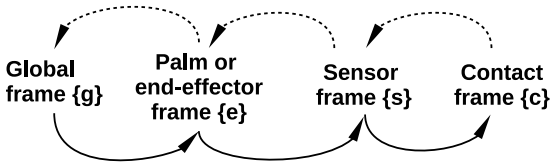


Figure C.1: Translational and Rotational Transformations from the contact to the global frame.

Therefore, the error ${}_{\{s\}}\Delta \boldsymbol{x''}$ will be expressed in the base frame $\{g\}$ of the robot as:

$$ {}_{\{g\}}\Delta \boldsymbol{x''} = {}_{\{g\}}\boldsymbol{T}^{\{s\}} \cdot {}_{\{g\}}\Delta \boldsymbol{x'} \tag{C.1} $$

Thus, the contact frame error will be transformed from the sensor frame to the global (base) frame. An example of this spatial transformation is shown in Fig. C.2. There are global, actual end-effector, desired end-effector, sensor, actual contact, and desired contact frames depicted as $\{g\}$, $\{e\}$, $\{e'\}$, $\{s\}$, $\{c\}$, and $\{c'\}$, respectively. The $\{s\}$ is rigidly attached to the $\{e\}$. The first example given in Fig. C.2a shows how a translational error $\Delta y = \{c'\} - \{c\}$ given in $\{s\}$ is transformed to the error between $\{e'\}$ and $\{e\}$ as given in $\{g\}$. The increment of the position of the end-effector expressed in the global frame is the following:

$$ {}_{\{g\}}\boldsymbol{\Delta p_e} = {}_{\{g\}}\boldsymbol{Rot}^{\{e\}} \cdot {}_{\{e\}}\boldsymbol{Rot}^{\{s\}} \cdot {}_{\{s\}}(\boldsymbol{p_{c'}} - \boldsymbol{p_c}), \tag{C.2} $$

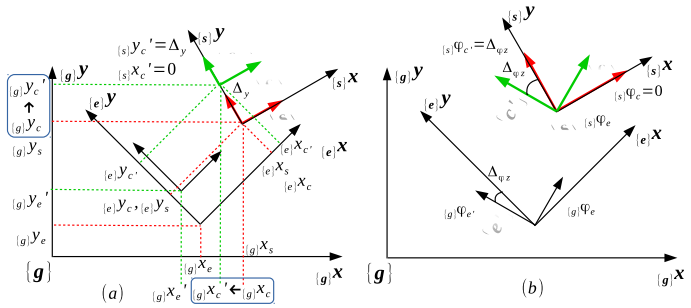


Figure C.2: Contact frame transformations: (a) translational and (b) rotational.

where ${}_{\{s\}}\boldsymbol{p_{c'}}$ and ${}_{\{s\}}\boldsymbol{p_c}$ are the desired and actual positions of the contact frame, ${}_{\{g\}}\boldsymbol{Rot}^{\{e\}}$ and ${}_{\{e\}}\boldsymbol{Rot}^{\{s\}}$ are the matrices of rotation from $\{g\}$ to $\{e\}$ and from $\{e\}$ to $\{s\}$, respectively and ${}_{\{s\}}p_{c'} - {}_{\{s\}}p_c$ is the translational error between the desired and actual contact frames.

The rotational transformation is:

$$ {}_{\{g\}}\Delta\varphi_e = ({}_{\{c\}}\boldsymbol{Rot}^{\{c'\}})^{-1} \cdot {}_{\{g\}}\boldsymbol{Rot}^{\{e\}}, \tag{C.3} $$

where ${}_{\{g\}}\Delta\varphi_e$ is the end-effector's rotational increment; $({}_{\{c\}}\boldsymbol{Rot}^{\{c'\}})$ is the rotational matrix from the actual to the desired contact orientation $\left(({}_{\{c\}}\boldsymbol{Rot}^{\{c'\}})^{-1} = ({}_{\{c\}}\boldsymbol{Rot}^{\{c'\}})^{T}\right)$. Recalling the equations eq. C.2 and C.3, the error vector wrt global frame in Cartesian space is:

$$ {}_{\{g\}}\Delta x = \begin{bmatrix} {}_{\{g\}}Rot^{\{e\}} \; {}_{\{e\}}Rot^{\{s\}} \cdot \Delta \begin{bmatrix} x_c \\ y_c \\ z \end{bmatrix} \\ \Delta \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix}^{-1} \quad {}_{\{g\}}Rot^{\{e\}} \end{bmatrix} \tag{C.4} $$

Appendix D

Tactile error convergence

The convergence of the tactile feature error is achieved by the external PI regulators and the internal PD+gravity compensation motion control. In the following the stability of the PD with gravity compensation motion control will be described in brief[1].

As mentioned in Chapter3, Assuming that the robot has six joints, the dynamics of the robot is given by the following equation of motion [Siciliano 08]:

$$\mathbf{M}(q)\ddot{q} + \mathbf{C}(q,\dot{q})\dot{q} + g(q) = \tau_a \tag{D.1}$$

where q and $\dot{q}$ are the vector of joint angles and joint velocities, respectively, $\mathbf{M}(q)$ is the inertia matrix, $\mathbf{C}(q,\dot{q})$ is the matrix representing the Coriolis and centrifugal effects, $g(q)$ is the vector representing the gravitational effect, τ_a is the set of actuating joint torques. The dynamics of the system under the joint torque control (eq. (3.13)) is given by:

$$\mathbf{M}(q)\ddot{q} + \mathbf{C}(q,\dot{q})\dot{q} + g(q) = \mathbf{J}_\theta^T(\mathbf{K_P}(e) - \mathbf{K_D}\dot{q}) + g(q) \tag{D.2}$$

where $e = x_d - x$. This equation represents the closed loop motion dynamics. The stability achieved by PD control with gravity compensation can be analyzed according to the above closed-loop dynamic equation. It has been shown [Siciliano 08] that there exist a Lyapunov positive-definite candidate function, $V(\dot{q},e) = \frac{1}{2}\dot{q}^T\mathbf{M}(q)\dot{q} + \frac{1}{2}\dot{e}^T\mathbf{K_p}(e) > 0 \quad \forall \dot{q}, e \neq 0$. The time derivative of this function, $\dot{V} = -\dot{q}^T J_\theta^T \mathbf{K_D} J_\theta \dot{q}$, decreases as long as $\dot{q} \neq 0$ Hence, the system (D.2) reaches an equilibrium posture so that $J_\theta^T \mathbf{K_p} e = 0$. J_θ is the Jacobian matrix of the robot kinematics. In the directions that are not constrained

[1] it can be found in the related literature.

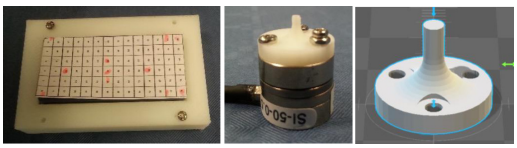
by the stiffness of the environment (also, but not important, in the absence of interaction forces), the tactile feature error convergence is guaranteed by the robot motion controller. In the directions, where interaction forces exist, the efficacy of the external force control algorithm with the internal position control using the PI regulator in the outer loop has been shown by Perdereau et al. [Perdereau 93].

Appendix E
Sensor parameters

E.1 Calibration setup

The following tools (Fig. E.1) are used to characterize the sensor's response:

- a schematic drawing of the tactels' locations (Fig. E.1, a),
- the ground-truth ATi Nano17 force and torque sensor (Fig. E.1, b),
- custom made cylindrical indenter with the diameter of 1 mm (Fig. E.1, c),
- the proxxon mf70 D-54518 milling device (Fig. E.2) used as a three axes manipulator.



(a) (b) (c)

Figure E.1: Auxiliary tools used during calibration: schematic of the sensing grid (a), the ground-truth force and torque ATI Nano 17 sensor (b), the indenter printed using a 3D printer (c).

The sensor was fixed (with a thin double side scotch) to a flat surface within the three axes manipulator's workspace.The manipulator has the resolution of 0.2 mm in the horizontal directions and 0.05 mm in the vertical direction (see red arrows in the Fig. E.2). The ATI Nano17 force sensor was attached to the milling head using a custom made adapter piece. This force sensor with an attached indenter was pushed against the elastic sensing surface over the centers of the sensor's tactels. The deformations in the z direction of the sensing frame, tactile sensor outputs, and the force sensor measurements were captured by pressing the elastic surface incrementally up to the saturation of the sensor output.

The signals from the tactile array and ground-truth force sensor were synchronized using the Robot Operation System (ROS) framework.

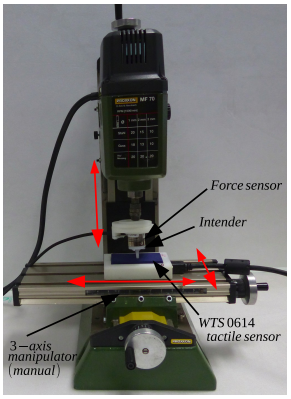


Figure E.2: Calibration setup for the WTS0614 tactile sensor.

E.2 Calibration results

The sensor response was recorded in three ways. The first way was dedicated to find the relation between the ground-truth forces and responses of each tactel in the array. The indenter pressed once on different tactels (at the center and corners). The second way was dedicated to test the repeatability of a given tactel. The responses of a single tactel were recorded from multiple trials by pressing several times on it with the indenter. The third way was dedicated to identify the mechanical compliance of the sensing surface. The indenter pressed on a tactel increasing the penetration depth slowly, recording meanwhile the penetration depth and the force response.

I - Tactile sensor output calibration with the ground truth force.

Fig. E.3 shows the sensor's responses versus the ground-truth force measurements. The linearized response is depicted with the black line. We can assume that interaction forces are within the linear range (from 0.2 N to 1.8 N with a ratio output→force of $k_{out \rightarrow force} = (1.8 - 0.2)/(3840 - 0) = 0.0004 \frac{N}{out}$) of the tactile sensor's response[1]. The maximum detectable pressure applied by the indenter with the diameter of 1 mm is given

[1] though the response of an element of the sensor can be characterized with a 3rd order polynomial (depicted with the red color)

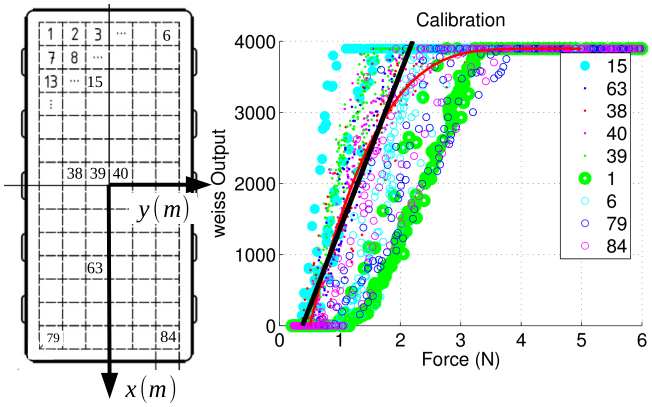


Figure E.3: Responses of the individual sensor elements (tactels) to the applied forces on each element. The sensor outputs are the raw values of the WEISS WTS 0614 sensor that is based on the piezoresistive technology. Locations of tactels are on the left-hand side of the picture and the tested tactels are shown in the legend of the plot.

by the maximum normal force F_{max} applied onto the sensor surface with the area of $\pi \cdot r^2$: $F_{max}/(\pi \cdot r^2) = (2N/0.785m^2) \cdot 10^6 = 2.548 \cdot 10^6$ Pa, that is $0.25N/mm^2$or 2.89 N per tactel. There is a significant difference in the minimum trigger level (i.e. the minimum detectable contact force) of the tactel 15, which is close to the center, and the tactel 1, which is located at the border: the tactel in the center is more sensitive than the tactel near the border. This irregularity is the side effect of the construction of the sensor: the rubber at the corners is not as flexible as at the center, because the rubber is attached to the base of the sensor from the sides.

II - Repeatability from multiple indentations.

The sensor responses have a creep behavior over a series of deformations as shown in the Fig. E.4. The response of the same tactel changes from one trial to another: the indenter pushed the tactel number 39 several times for each trial.

The repeatability of the response of the tactel had a variation of almost 1.5 times at the point around 1.5 N. This happens because of the hysteresis of the flexible surface, temperature variations, and a creep behavior under a constant force [Robotics 14].

III - Compliance constant of the sensing surface.

In order to estimate the compliance constant C_z, the sensing surface was deformed by the displacements of 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, and 1.4 mm. Fig. E.5 shows the response

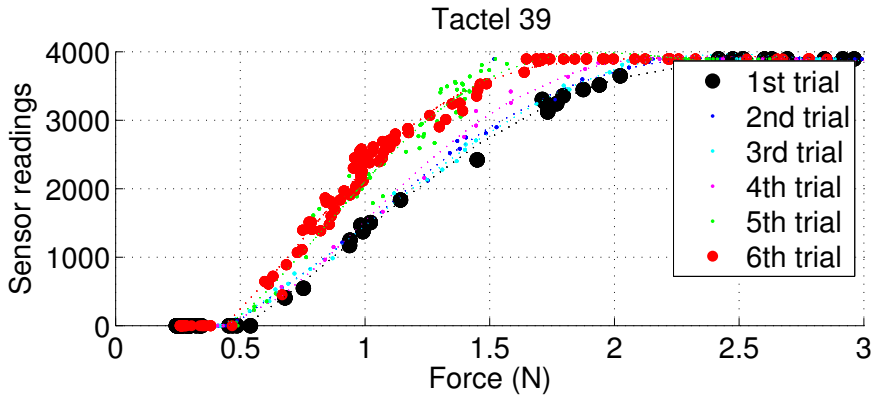


Figure E.4: Sensor noise. Step responses of the tactel 39 from multiple trials.

of the ATI Nano17 force sensor (not the response of a tactel) to the indentations of the different depths and the linear fit (red line). The estimated stiffness of the sensor is $k \simeq 2$ N/m.

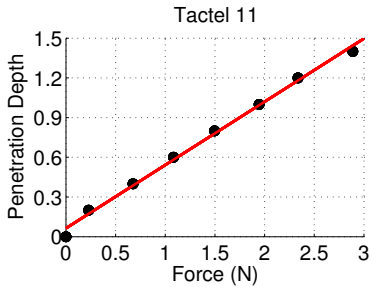


Figure E.5: Response of the tactel number 11 to indentations of different depths.

As it was stated in the robotics literature, e.g. [Prats 13], there are significant uncertainties in the data acquisition of tactile sensing arrays, which make the current tactile arrays unusable for precise force control algorithms. The identified characteristics of the sensor remain only rough estimates. Therefore, in the most of applications with real tactile sensors, the force should be estimated in a such way so that it will be proportional to the area of contact (see Section 3.5.2 and Fig. 3.19 for our implementation of force control with tactile sensors).

Appendix F

Other issues in sensor integration

F.1 Issues related to the shape of the attachment surface

Mounting tactile sensors on a robot palm, jaw grippers or fingers with flat surfaces is relatively straightforward; one of the simplest ways involving using a double side tape. Figure F.1(a) shows an experimental setup containing the Shadow Hand and the Tekscan tactile sensing system (Model 4256E), which was used for contact shape recognition [Liu 12a]. In another manipulating setup, off-the-shelf capacitive arrays have been installed on the fingertips of the four-fingered "Allegro" robot hand (Fig. F.1(b)). Fig. F.1(c) illustrates the Robotiq adaptive gripper covered by capacitive pressure sensing arrays used for the recognition of the type of the slip [Heyneman 12].

Attaching tactile sensors on fingers and fingertips is a complex process as curved surfaces with small radius of curvature have to be taken into account. Tactile sensors should be either:
a) flexible and appropriately shaped to envelop a given surface, as in iCub tactile fingertip sensors (Fig. F.2(a, b) [Schmitz 10]; b) rigid and shaped as an attachment part, e.g. [Koiva 13] or [Liu 15] where a 3D-shaped tactile sensing array and an ellipsoid F/T sensor (Fig. F.2(c) and (e)) replace the fingertips of the Shadow robot hand [Shadowrobot 14]. In another version of the Shadow robot Hand with the integrated BioTac multimodal tactile sensor, each finger loses one DoF (Fig. F.2(d)), – the sensor is as big as the two last links, distal and middle phalanges of the human index finger.

The shape of the links of the fingers in robot hands is different from the shapes of human finger phalanxes. The proximal and middle links of fingers in artificial robot hands

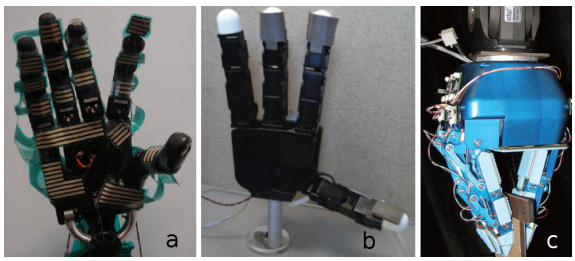


Figure F.1: Simple integration of tactile sensing arrays: (a) the Tekscan tactile sensing system consisting of 349 taxels with the Shadow robot hand [Liu 12a], (b) the Allegro robot hand with PPS RoboTouch capacitive arrays [Jara 14], (c) the Robotiq adaptive gripper with sensor suite installed on the contact surface [Heyneman 12]

have a smaller contact surface than those of humans, a fact that significantly decreases the sensing area and causes difficulties with attachment. Fig. F.3 shows the difference between sensing areas on the middle and proximal links of a human finger and a robot finger. Current artificial tactile sensors are not as flexible as human skin and cannot cover the empty space between the links for closing the finger of robot hands.

F.1.1 Wiring issues

A key issue in tactile sensing array integration is the amount of wires required to read and transmit the data from the sensing arrays. Any increase in the number of tactels in tactile sensing array causes an increase either in the number of wires or/and on the time needed for the data acquisition from sensors. A serial data communication can be used to reduce the number of connections. For example, in the iCub skin, communication was implemented through I2C serial bus, where only four wires were connected to the PCB of the sensing array [Schmitz 11]. However, serial access of data is slower than parallel access. In iCub, the skin sampling rate for each tactel decreases from 100 Hz to 25 Hz as the number of tactels increases. If the real-time pressure distribution is of interest, as for example in tactile servoing [Li 13a], the serial data access may fail to produce time-series images of the contact image. The parallel access of data provides higher acquisition rate, but requires a higher number of wires than the serial one. Employing advanced addressing schemes is a way of reducing the number of wires needed in the parallel

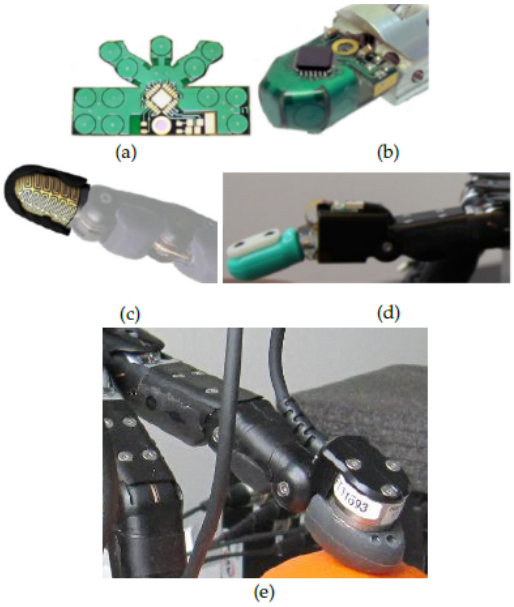


Figure F.2: Advanced integration of tactile sensors on the robot fingertips: (a) a flexible PCB for a capacitive tactile sensing array with 12 taxels designed for the iCub humanoid robot [Schmitz 10],(b) the iCub flexible PCB wrapped around the inner support of the fingertip [Schmitz 10], (c) a 3D-shaped rigid tactile sensing array with 12 sensing elements attached to the fingertip of the Shadow robot hand [Koiva 13], (d) the BioTac multimodal tactile sensor installed on the Shadow robot hand by replacing the two last links of the finger [Xu 13], (e) ATi nano 17 force/torque sensor on the fingertip of the Shadow robot hand [Corrales 13].

access schemes. For example, in the row-column scheme [Drimus 14] $n + m$ wires are needed for $n * m$ arrays of sensors instead of $n * m + 1$ wires required in the schemes with one common ground [Koiva 13]. Other approaches dedicated to reducing wiring issues include wireless data and power transmission and implementation of a decentralized data pre-processing of tactile signals [Göger 09], [Kampmann 14].

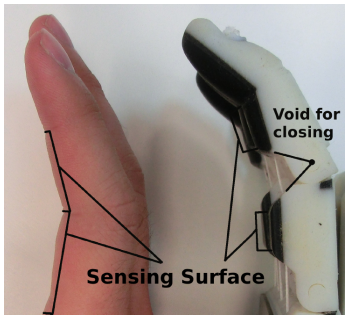


Figure F.3: Difference in contact surfaces between a human finger and a robot finger [Kappassov 13]