



# Continuous Tactile Sensing for Enhanced Human-Robot Collaboration

DOCTOR OF PHILOSOPHY THESIS

PRESENTED BY

*Augusto Gómez Eguíluz*

May, 2018

*I confirm that the word count of this thesis is less than 100,000 words*



# Contents

List of Figures	v
List of Tables	ix
Acknowledgements	xi
Abstract	xiii
Abbreviations	xv
Notation	xvii
Declaration	xix
<b>1 Introduction</b>	<b>1</b>
1.1 Objectives of the Thesis . . . . .	3
1.2 Thesis Contributions . . . . .	4
1.3 Outline of the Thesis . . . . .	6
<b>2 Tactile Sensing in Robotics</b>	<b>9</b>
2.1 Introduction . . . . .	9
2.2 From Human to Robotic Tactile Sensing . . . . .	10
2.3 Robotic Tactile Sensing Hardware . . . . .	11

2.4	An Overview of Robotic Tactile Sensing Applications . . . . .	17
2.4.1	Contact Localization . . . . .	18
2.4.2	Tactile Sensing and Control . . . . .	19
2.4.3	Slip Detection and Prediction . . . . .	22
2.4.4	Object Recognition . . . . .	23
2.4.5	Identification of Object Properties . . . . .	24
2.5	Material Identification . . . . .	25
2.6	Grasping . . . . .	32
2.7	Object Handover . . . . .	39
2.8	Summary . . . . .	44
<b>3</b>	<b>Robotic Platform &amp; Methods</b>	<b>47</b>
3.1	Introduction . . . . .	47
3.2	Robotic Platform: The Shadow Dexterous Hand . . . . .	48
3.3	Machine Learning . . . . .	51
3.3.1	Principal Component Analysis . . . . .	52
3.3.2	Expectation Maximization . . . . .	53
3.3.3	Linear Regression . . . . .	56
3.3.4	Classification Techniques . . . . .	58
3.4	Signal Processing . . . . .	65
3.4.1	Fourier Transform . . . . .	65
3.4.2	Random Sample Consensus Algorithm . . . . .	67
3.5	Robotics and Control Systems . . . . .	69
3.5.1	Robot Kinematics . . . . .	69
3.5.2	Differential Motions . . . . .	73
3.5.3	Motion Control . . . . .	75
3.5.4	Bayesian Filtering . . . . .	79
3.6	Summary . . . . .	83



<b>4</b>	<b>Continuous Material Identification</b>	<b>85</b>
4.1	Introduction . . . . .	85
4.2	Continuous Material Identification . . . . .	87
4.2.1	Signal Processing and Vibration Features . . . . .	87
4.2.2	Statistical Modelling of Vibration . . . . .	91
4.2.3	Material Recursive Bayesian Estimation . . . . .	92
4.2.4	Multimodal Continuous Material Classification . . . . .	94
4.3	Experimental Results . . . . .	98
4.3.1	Experimental Set-Up . . . . .	100
4.3.2	Microvibrations-based Material Identification . . . . .	101
4.3.3	Multimodal Material Identification . . . . .	109
4.4	Summary . . . . .	115
<b>5</b>	<b>Precision Grasp control through Continuous Tactile Sensing</b>	<b>117</b>
5.1	Introduction . . . . .	117
5.2	Precision Grasp Control through Tactile Sensing . . . . .	119
5.2.1	Cartesian Force Estimation . . . . .	120
5.2.2	Precision Grasp Controller . . . . .	122
5.3	Precision Grasp Control for Tabletop Manipulation . . . . .	126
5.3.1	Scene Segmentation . . . . .	127
5.3.2	Kinect-Robot Frame Transformations . . . . .	129
5.3.3	Grasping Objects with Unknown Shape . . . . .	131
5.4	Experimental Results . . . . .	135
5.4.1	Scene Segmentation . . . . .	136
5.4.2	Finding the Shadow Robotic Hand and the Object . . . . .	138
5.4.3	Precision Grasp Control . . . . .	141
5.5	Summary . . . . .	144

<b>6</b>	<b>Continuous Tactile Sensing for Reliable Robot-Human Handovers</b>	<b>147</b>
6.1	Introduction . . . . .	147
6.2	Grasping Effort Controller . . . . .	149
6.3	Release Detection . . . . .	152
6.3.1	Robot-Human Handovers: A Naïve User Study . . . . .	152
6.3.2	Load Force Variations for Event Detection . . . . .	156
6.3.3	Features for force direction detection . . . . .	158
6.3.4	Statistical learning of perturbation force directions . . . . .	160
6.4	The Reliable Object Handover Algorithm . . . . .	161
6.5	Experimental Results . . . . .	162
6.5.1	Force Evolution Filtering . . . . .	165
6.5.2	Object Perturbation Release Detection . . . . .	167
6.5.3	Force Adaptation and Object Handover . . . . .	173
6.6	Summary . . . . .	177
<b>7</b>	<b>Conclusions and Future Work</b>	<b>179</b>
7.1	Summary of Thesis Contributions . . . . .	180
7.1.1	Continuous Material Identification . . . . .	181
7.1.2	Robotic Grasping of Unknown Shaped Objects . . . . .	182
7.1.3	Robot-Human Object Handovers . . . . .	183
7.2	Future Work . . . . .	184
7.3	Conclusion . . . . .	187

# List of Figures

2.1	Conceptual scheme of a piezoresistive sensor [1]. . . . .	12
2.2	Conceptual scheme of tactile sensors based on capacitive effect. . . . .	13
2.3	Conceptual scheme of a PVDF film used in a piezoelectric tactile sensor [2]. . . . .	14
2.4	Conceptual scheme of a quantum tunnel effect sensor [3]. . . . .	15
2.5	BioTAC hardware scheme [4]. . . . .	16
2.6	Generic definition of tactile sensing solutions for robotic grasping and object handover. . . . .	45
3.1	Scheme of the joints in a regular Shadow hand [5]. . . . .	49
3.2	The Shadow robotic hand. . . . .	50
3.3	Examples of the EM algorithm running on different datasets. . . . .	54
3.4	Examples of linear regression applied on different datasets. . . . .	57
3.5	The classification problem. . . . .	58
3.6	Artificial Neural Network. . . . .	60
3.7	Comparison between linear regression and robust line estimation. . . . .	68
3.8	PID controller schemes. . . . .	77
3.9	Response of different PID-type controllers for similar input signals. . . . .	78
4.1	FFT representation of four materials used in the experiments. . . . .	88

4.2	Log-likelihood of a GMM as a function of the number of clusters $k_j$ for three different materials. . . . .	93
4.3	Material identification pipeline scheme. . . . .	99
4.4	Experimental setup for material identification. . . . .	101
5.1	Precision grasp control diagram. . . . .	124
5.2	The Shadow robot hand attached to a Shunk arm. . . . .	126
5.3	Tabletop manipulation scene diagram. . . . .	129
5.4	Flow chart of the proposed precision sphere grasp operation. .	133
5.5	Tabletop manipulation experimental set-ups. . . . .	135
5.6	Segmentation of a tabletop manipulation scene. . . . .	137
5.7	Objects used in the experiment presented in Section 5.4.3. . .	140
5.8	Sequences of grasping different objects. . . . .	142
5.9	Force magnitude evolution over time for grasping and lifting an octagon prism. . . . .	144
6.1	Flow chart of the proposed reliable object handover algorithm.	148
6.2	Changes in the fingertip force length for effort vs. position control. . . . .	150
6.3	Receiver positions. . . . .	153
6.4	Initial configurations. . . . .	154
6.5	Azimuth $\theta$ and elevation $\phi$ angle variations for all participant first fingers. . . . .	155
6.6	Experimental set-up. . . . .	164
6.7	Index finger angle variations during downwards perturbation force. . . . .	166
6.8	Recursive estimation of perturbation force directions. . . . .	168
6.9	Identification speed for different values of $t_{th}$ . . . . .	169

6.10	Identification accuracy for different values of $t_{th}$ . . . . .	170
6.11	Object rotation and pulling event sequence. . . . .	173
6.12	Forces response against object rotation and pulling events. . .	175
6.13	Perturbation on object classification of a sequence of events. .	176



# List of Tables

2.1	State-of-the-art in tactile material identification. . . . .	31
2.2	State-of-the-art in tactile sensing for robotic grasping. . . . .	38
2.3	State-of-the-art in robotic object handover. . . . .	43
4.1	List of materials used in all our classification experiments. . .	102
4.2	Average time (in seconds) needed for material classification using the FFT of the vibration signals and 10-fold cross-validation.	105
4.3	Average time (in secs) needed for multimodal material classification using 10-fold cross-validation. . . . .	111
4.4	Summary of material identification results. . . . .	113
4.5	Iterative estimation of material 17 using vibration only approach with FFT-PCA. . . . .	114
4.6	Multimodal iterative estimation of material 17. . . . .	115
5.1	Thumb D-H parameters of a standard Shadow Dexterous hand.	125
5.2	Standard deviation $\sigma_1$ of the position estimation for three different objects. . . . .	138
5.3	Mean $\mu$ and standard deviation $\sigma$ (in degrees) of the robot forearm orientation relative to the Kinect reference frame. . .	140
6.1	Confusion matrix of perturbation force directions for $t_{th} = 0.15$ .	171
6.2	Confusion matrix of perturbation force directions for $t_{th} = 0.25$ .	172





# Acknowledgements

I would like to thank those who has supported me during the course of my PhD program. Firstly, I would like to express my sincere gratitude to my supervisors Dr. Iñaki Ranó, Prof. Sonya Coleman and Prof. Martin McGinnity for the continuous support of my PhD study and related research, for their patience, motivation, and knowledge. They have been a role model from which I have developed my knowledge of scientific research and the academic life.

My special thanks are extended to my colleagues and friends of the Intelligent Systems Research Centre of the Ulster University, who have facilitated my everyday life and contributed to make a delightful experience of studying this PhD program. In particular, I would like to offer my special thanks to Mr. Michael Callaghan for encouraging me to undertake my PhD studies.

The financial support of a Vice-Chancellor's Research Scholarship (VCRS) has been of great value to conduct this research and I would like to express my very great appreciation to the Ulster University for this opportunity.

Last but not the least, I would like to thank my family, especially my parents and María Eugenia, and friends for their moral support throughout the last three years and my life in general.



# Abstract

Collaborative manipulation of objects is usually a trivial activity for humans but is still very challenging for robots. Such tasks involve many complex aspects, such as human and object safety, social and handling context, grasping stability, slip detection, and ergonomics. Although huge research efforts have been devoted over decades to endow robots with the skills required for grasping, manipulation, sharing of objects, and collaboration with humans, there is still a need for reliable systems capable of reacting to unexpected events. As for humans, the sense of touch is essential for robots to perform many tasks as it provides information that can not be obtained through contactless sensing modalities. Thus, recent trends in robotics research explore the use of tactile sensing in human-robot object manipulation.

An important aspect that is often overlooked in the existing literature is that tactile sensing is inherently sequential and therefore should be approached as a continuous process. The aim of this thesis is to explore continuous tactile sensing to enhance robot collaboration capabilities for object manipulation. The contribution of this work is threefold: firstly, an innovative multimodal technique that identifies the surface materials of objects using continuous tactile sensing is developed. Secondly, continuous tactile sensing is used to provide contact information to a control system that grasps objects of unknown geometry. Finally, an approach to hand over objects between

a robot and a human, relying on continuous tactile sensing, is developed to ensure the safety of the robot and the object during the transfer.

In this thesis, the proposed approaches are evaluated on real physical robotic platforms. A comparison with the state-of-the art techniques in material recognition shows that the proposed multimodal approach enhances identification speed and accuracy. The experimental results also show excellent performance of the proposed approach for grasping objects even when information about their geometry is not available. Finally, the proposed object handover algorithm is proven to adapt to unexpected force perturbations on the object and release it in a timely manner without dropping. This work entails significant progress towards the development of autonomous robots that collaborate with humans in everyday tasks.

# Abbreviations

**ANN** Artificial Neural Network

**DFT** Discrete Fourier Transform

**DMP** Dynamic Motion Primitive

**DoF** Degrees of Freedom

**EM** Expectation Maximization

**EMG** Electromyography

**FFT** Fast Fourier Transform

**FK** Forward Kinematics

**FT** Fourier Transform

**GMM** Gaussian Mixture Model

**HMM** Hidden Markov Model

**JGKSC** Joint Group Kernel Sparse Coding

**KF** Kalman Filter

**$k$ NN**  $k$ -Nearest Neighbours

**LBP** Local Binary Patterns

**LQG** Linear Quadratic Gaussian

**MAP** Maximum A Posteriori

**ML** Maximum Likelihood

**MLP** Multilayer Perceptron

**OvA** One versus All

**OvO** One versus One

**PCA** Principal Components Analysis

**PCL** Point Cloud Library

**PID** Proportional-Integral-Derivative

**PVDF** Polyvinylidene Fluoride

**QTC** Quantum Tunnel Composite

**RANSAC** Random Sample Consensus

**RBE** Recursive Bayesian Estimator

**RBF** Radial Basis Function

**SVD** Singular Value Descomposition

**SVM** Support Vector Machine

# Notation

In this thesis, the notation will be kept as simple as possible but, in order to be consistent, it is useful to define the general terms for symbols and abbreviations. We write:

$x$  for scalars (i.e. lowercase);

$\mathbf{x}$  for vectors (i.e. bold lowercase);

$\mathbf{X}$  for matrices (i.e. bold capital letters);

$x_i$  referring to the  $i^{th}$  element of the vector  $\mathbf{x}$  and, similarly,  $X_{ij}$  corresponding to the element in the  $i^{th}$  row and  $j^{th}$  column of the matrix  $\mathbf{X}$ ;

$\hat{x}$  for predictions;

$\tilde{x}$  for random variables;

$p(\mathbf{x})$  is the compact way of denoting  $p(\tilde{x} \leq \mathbf{x})$  for continuous variables and  $p(\tilde{x} = \mathbf{x})$  for discrete variables.





# Declaration

“I hereby declare that for 2 years following the date on which the thesis is deposited in Research Student Administration of Ulster University, the thesis shall remain confidential with access or copying prohibited. Following expiry of this period I permit

1. the Librarian of the University to allow the thesis to be copied in whole or in part without reference to me on the understanding that such authority applies to the provision of single copies made for study purposes or for inclusion within the stock of another library.
2. the thesis to be made available through the Ulster Institutional Repository and/or EThOS under the terms of the Ulster eTheses Deposit Agreement which I have signed.

IT IS A CONDITION OF USE OF THIS THESIS THAT ANYONE WHO CONSULTS IT MUST RECOGNISE THAT THE COPYRIGHT RESTS WITH THE UNIVERSITY AND THEN SUBSEQUENTLY TO THE AUTHOR ON THE EXPIRY OF THIS PERIOD AND THAT NO QUOTATION FROM THE THESIS AND NO INFORMATION DERIVED FROM IT MAY BE PUBLISHED UNLESS THE SOURCE IS PROPERLY ACKNOWLEDGED.



# Chapter 1

## Introduction

Touch is at the core of many human skills such as grasping, material identification, and temperature detection. Although a large component of human perception is obtained through vision, it is often complemented with tactile sensing; for instance, when grabbing a pencil. Experimental analysis of human visual and tactile representations show that they are closely linked, and that the same brain regions are activated during visual and tactile exploration of objects. Tactile sensing in humans involves two main sub-modalities, kinesthetic and cutaneous. The former receives sensory inputs that occur within the body, i.e. muscles or tendons, while the latter receives the sensory inputs from different types of receptors of the skin.

Recent trends in robotics explore the applications of tactile sensing to endow robots with human-like capabilities. By convention, robotics tactile sensing refers only to the cutaneous sub-modality. Hence, in this thesis, no distinction is made between cutaneous and tactile sensing. In robotics, kinesthetic information is typically perceived using kinematic/dynamic models of the robot and joint positional/torque sensors, while tactile (i.e. cutaneous) information is obtained using tactile sensors. Despite human tactile

sensing being continuous in nature, the existing literature on robotic tactile sensing often relies on episodic readings or does not consider the potential of tactile sensing to provide adaptation upon contact. This work investigates tactile perception as a continuous process to enhance awareness of the outer surface of the robot. In particular, this thesis focuses on continuous tactile perception from robotic fingertips.

Robots have traditionally perceived their environment through contactless sensing (e.g. proximity, laser, visual). While extensive research in robotics has focused on vision, the research on robotic tactile sensing is still in its infancy. However, contact properties such as the friction coefficient, compliance or roughness cannot be perceived using contactless sensing. Identifying the material from which an object is made is crucial for robots, as the objects contact properties are related to the surface material. The contact properties of an object can be inferred when its surface material is known and, therefore, can be used to improve robotic capabilities for object manipulation.

For a robot to manipulate an object, a required and fundamental skills is the ability to grasp. Most of the object manipulation tasks in everyday scenarios involve grasping an object. While humans grasp objects intuitively, it is still very challenging for robots, especially if the geometry of the object is not known. Tactile sensing plays a significant role towards the integration of robotic grasping systems that adapt to different object geometries. This work will research the applications of continuous tactile sensing to enhance robotic grasping of objects when their geometry is unknown.

In the near future, robots are expected to collaborate with humans in everyday tasks. Reliable grasping systems are essential for robotic manipulation of objects, moreover, the ability to collaborate with humans often

requires material or object interactions. Such interactions require, on the part of the robot, complex sensory and cognitive systems that consider the physical properties of the object and the unexpected events that may arise during the manipulation. A typical object manipulation task for humans when collaborating with each other is object handover. State-of-the-art object handover systems are still limited to laboratory experimental conditions and the safety of the robot and the object has not been addressed properly in current research. This challenge is addressed in this thesis.

In summary, this research explores robotic tactile sensing as a continuous process and presents applications of this paradigm for material identification, grasping and robot-human object handover.

## 1.1 Objectives of the Thesis

The aim of this research is to explore continuous tactile sensing to enhance robot collaboration capabilities for object manipulation. Addressing tactile perception as a continuous process will advance the incorporation of robots in human-centric environments. In particular, this work paves the way for the development of autonomous robots in three ways. Firstly, continuous tactile sensing will be used to enhance material identification of object surfaces, which provides crucial information necessary for underpinning effective grasping strategies. Secondly, robotic grasping of unknown shaped objects will be addressed by taking advantage of the contact properties determined by the surface material and adapting the grasp using tactile sensing. Thirdly, a system to handover objects between a robot and a human, while simultaneously providing continuous adaptation to perturbation forces on the object, will be implemented on a real robot. To fulfil the aim of this thesis, the

following objectives have been identified:

- Design a continuous material surface identification mechanism based on tactile sensing that produces optimal classification results in a minimum time;
- Implement a robotic system capable of grasping a range of objects without previous information of their geometry;
- Develop a tactile sensing based system to hand over objects in a secure manner between a robot and a human.

As it will be shown in Chapter 2, these objectives entail a number of open challenges. For instance, material identification through tactile sensing is a difficult problem as state-of-the-art tactile information is multimodal and, in order to achieve high classification accuracy, relevant descriptors need to be extracted from sequential data. Moreover, the enhancement of robotic grasping and manipulation through tactile sensing involves the design of advance control systems using contact models from multimodal data.

## 1.2 Thesis Contributions

The work detailed in this thesis provides a significant and innovative contribution to robotics. The main contribution of this thesis is to approach robotic tactile sensing as a continuous process and explore its applications to enhance human-robot collaboration in object manipulation tasks. The research has been published in four peer-reviewed top-tier conferences [6–9] and has contributed to two journal papers [10, 11]. Thus, the specific contributions of this thesis are:

- Continuous material identification through tactile sensing:
  - A material identification approach that takes into account of the continuous nature of tactile sensing [6, 10];
  - The design of a multimodal approach for continuous material identification of object surfaces [7, 10];
  - The definition of new descriptors for material identification based on the heat transfer between the tactile sensor and the object surface material [7, 10];
  - A comparative study with state-of-the-art tactile material identification techniques and descriptors [10];
- Robotic grasping of unknown shaped objects:
  - An approach for an accurate modelling of the contact forces in the fingertips of a robot using BioTAC tactile sensors [8];
  - A method to find the homogeneous transformation between a Shadow Robot hand and a Microsoft Kinect sensor;
  - The implementation of a precision grasp controller that allows grasping of objects of unknown shape;
- Robot-human object handovers through continuous tactile sensing:
  - Design of an effort controller that allows robots to adapt object grasps when unexpected perturbation forces act on the object during grasping [8];
  - Design of a continuous tactile perception approach for detection of perturbation force directions during object grasping [9, 11];

- Development of a system to hand over objects between a robot and a human that provides adaptation to perturbations on the object and releases only when it is safe for the robot and the object [8, 11].

## 1.3 Outline of the Thesis

Chapter 2 presents a critical review of the current state-of-the-art in robotic tactile sensing with special emphasis on its applications to material identification, grasping and object handover. Chapter 3 describes the robotic platform used in this work and provides insights into the techniques used to fulfil the aims of this research. A detailed description of the contributions of this research is provided in Chapters 4, 5, and 6. Finally, Chapter 7 concludes this thesis and details future potential research directions. A synopsis of the remaining chapters in this thesis is detailed below:

- **Chapter 2** includes a thorough literature review of tactile sensing. Bioinspiration of robotic tactile sensing, the taxonomy of tactile sensing hardware, and applications of robotic tactile sensing are described. Special attention is paid to material identification, grasping and object handover to understand the shortcomings of the state-of-the-art techniques.
- **Chapter 3** details the robotic platform used in this work. Additionally, a description of the techniques used in this thesis is detailed in this chapter and covers three main areas: machine learning, robotics and control systems, and signal processing.
- **Chapter 4** presents a material identification approach using contin-



uous tactile sensing of vibration signals generated when sliding a tactile sensor over the surface of an object. The proposed approach is extended to include additional sensing modalities and a method to enhance material identification using thermal information is presented. Both approaches, vibration only and multimodal, are compared with state-of-the-art material identification techniques.

- **Chapter 5** describes a technique to grasp objects of unknown shape. A model of the contact forces using BioTAC tactile sensors is presented and further used in a precision grasp controller that adapts the hand configuration upon contact to perform the grasp. The proposed strategy is evaluated in a tabletop manipulation scenario, providing a solution for grasping a variety of objects.
- **Chapter 6** outlines the development of a system to hand over objects between a robot and a human. The system guarantees continuous adaptation of a robotic hand to perturbations on the object using an effort controller based on the contact force model detailed in Chapter 5. Additionally, continuous identification of perturbation force directions is used to trigger the object release when the human pulls the object.
- **Chapter 7** concludes this thesis and summarises the results. A discussion of future directions of this research is presented and how these directions, together with the work in this thesis, could contribute to the development of autonomous robots that interact in human environments.



## Chapter 2

# Tactile Sensing in Robotics

### 2.1 Introduction

Tactile sensing in robotics has been for decades an under researched area, yet has great potential to improve object perception, grasping and manipulation. While applications of other sensing modalities (e.g. vision) have been widely explored, the application of tactile sensing for many fundamental problems in robotics has received much less attention. Just as humans use the sense of touch to perceive their surrounding objects, robotic perception can be enhanced by tactile information. Although human tactile perception is inherently continuous (i.e sequentially unfolds over time), existing works on tactile perception treat it as an episodic event. This work aims at exploring the continuous nature of tactile sensing and its application to material identification, grasping and object handover. This chapter provides a critical review of robotic tactile sensing research for these areas.

The chapter is organised as follows. Section 2.2 provides a brief introduction to the importance of tactile sensing in humans and robots. Section 2.3 provides an overview of the tactile sensing hardware. Section 2.4 presents

an introduction to the applications of tactile sensing in robotics. The three subsequent sections present thorough literature reviews of the applications of robotic tactile sensing explored in this thesis. Section 2.5 reviews the state-of-the-art of robotic material identification systems. A literature review of robotic grasping using tactile sensing is presented in Section 2.6. Section 2.7 reviews the state-of-the-art techniques for handing over objects between robots and humans. Finally, Section 2.8 concludes the chapter with a summary of the literature on tactile sensing and highlights the issues addressed in this thesis.

## 2.2 From Human to Robotic Tactile Sensing

Biological perception is the way a living being organises, identifies, and interprets sensory information [12]. It is a complex process in which typically a large number of sensors are involved. For instance, in the fingertips of a human adult, 241 mechanoreceptors (i.e. pressure and vibration sensors) per square centimetre are embedded in the skin at different depths [13]. Humans gather information from their environment through their senses and, thus, they can perceive the world from different modalities like vision, smell or touch.

The combination of multiple sensory modalities is needed to build accurate representations of the environment [14] as this typically maximises the information derived from each modality and enhances the reliability of the sensory estimation. In human cognition, different weights are given to the combination of sensing modalities depending on the nature of the perceived stimuli and the task to be executed [15]. For instance, the perception of size and shape obtained combining visual and haptic information in an optimal

manner [16] is more reliable than the unimodal estimate [15].

Furthermore, the analysis of visual and haptic representations in humans suggests that they are closely linked, since it has been observed that the same brain regions are activated during visual and haptic exploration of objects [17]. However, priority of vision over tactile sensing when humans intend to identify an object is proved in [18]. In contrast, the sense of touch seems to play a more significant role than vision for material perception [19]. One can conclude that tactile sensing is better suited to detect microgeometric properties while vision is better for macrogeometric properties. Moreover, tactile sensing is especially important during object grasping and manipulation, as the lack of tactile information precludes the coordination and control of a grasp [20], reducing the accuracy of the grasp and temporal performance.

Similarly, traditional robotic perception and object manipulation systems (i.e. using vision only) have severe limitations caused by sensor noise, the complex ways to combine perceived information and the uncertainty of the environments. Relevant information can be obtained using tactile sensing to deal with these limitations and enhance robotic manipulation [21]. Although, in comparison with other robotic sensing modalities (e.g. vision) tactile sensing is still in its infancy, applications and development of robotic tactile sensing have significantly grown during the last decade, with the advent of commercially available sensors.

## 2.3 Robotic Tactile Sensing Hardware

Although robotic tactile sensing is a relatively young area of research, different sensors designs i.e. pressure sensors [22] or haptic tools [23], have been explored during the last decades. Nowadays, robotic tactile sensing outper-

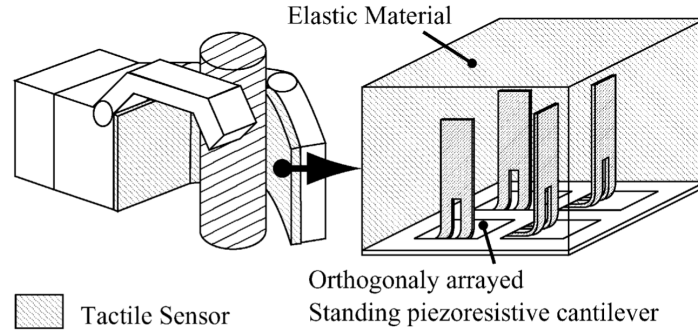


Figure 2.1: Concept of tactile sensor consisting of standing cantilevers with piezoresistors arrayed in orthogonal directions for shear stress detection [1].

forms human skills on surface material identification [24] and state-of-the-art sensors provide a variety of sensing modalities (e.g. 3-axis forces, microvibrations, pressure, fingertip compliance, temperature and thermal flux) that allow robots to understand the meaning of haptic adjectives [25]. A common practice when it comes to classifying tactile sensor types is to use their transduction (i.e. the different ways to build the sensing devices) to define a taxonomy [26]. There are eight main different types of tactile sensors, based on their transduction of contact information, which are:

- Piezoresistive sensors. These are composed of materials that change their electrical resistance when mechanical strain (i.e. deformation) is applied [27]. Therefore, variations in the forces applied to the material map into electrical resistance, which can be measured as a change in a resultant current or voltage. The materials possessing this feature are called piezoresistors [27]. An overview of a piezoresistive sensor is shown in Figure 2.1. Piezoresistive sensors present a number of drawbacks. The materials used in this type of sensor may change their properties according to the temperature and moistness [28], they are

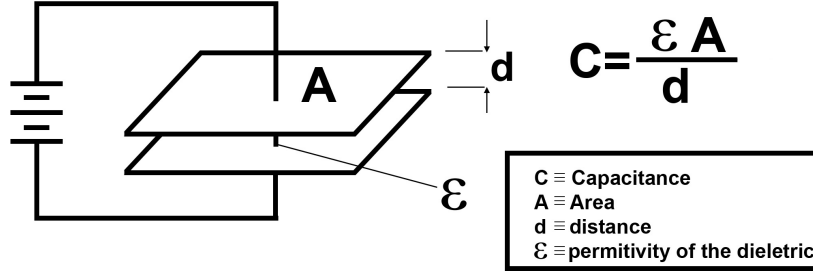


Figure 2.2: Conceptual scheme of tactile sensors based on capacitive effect.

fragile to shear forces and their response is non-linear. Additionally, the repeatability is highly limited as a piezoresistor may never recover its initial form after deformation, a phenomena known as hysteresis [29]. Nevertheless, piezoresistive sensors are used in robotics as they are easy to manufacture, and piezoresistors are commercially available while providing a flexible solution.

- **Capacitive sensors.** Capacitance is the ability of a body to store an electrical charge. Thus, capacitive sensors are built with two conductive plates separated by a dielectric material as in Figure 2.2. Applied forces are measured by detecting changes to the capacitance [22, 30]. Their main advantage is the higher frequency response in comparison with piezoresistive sensors [26]. However, capacitive sensors are sensitive to electromagnetic noise and changes of temperature, and their response is non-linear.
- **Piezoelectric sensors.** The piezoelectric effect refers to the electric charge that is accumulated in certain materials, such as ceramics, quartz crystals or polymers, in response to applied forces [31]. Piezoelectric tactile sensors frequently use polyvinylidene fluoride (PVDF) layers, such as the one in Figure 2.3, embedded into a rubber cover as they

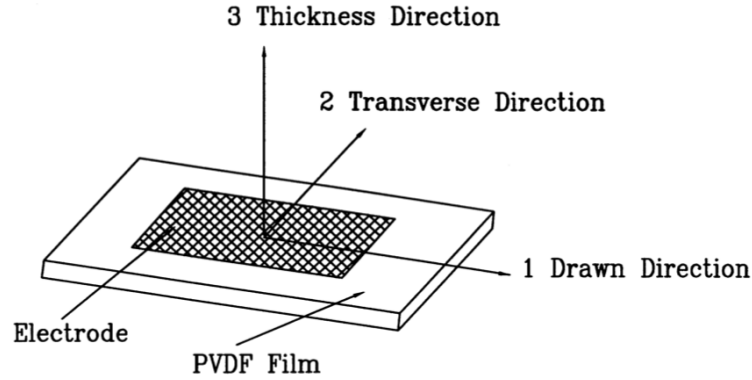


Figure 2.3: Conceptual scheme of a PVDF film used in a piezoelectric tactile sensor [2].

provide flexibility and chemical stability [26]. Despite providing faster dynamic response than capacitive sensors they are only suitable for dynamic measurements, sensitive to temperature [32], and their electrical junctions are fragile.

- Quantum tunnel effect sensors. Quantum tunneling happens when a particle breaks through an energy barrier that it typically could not cross [33]. Sensors composed of quantum tunnel composite (QTC) materials are capable of measuring applied forces. For instance, Figure 2.4 shows a conceptual scheme of a 3-axial sensor presented in [3]. Four electrodes sandwiching the electrode layers are used to measure the changes in the resistance. When a force is applied, the QTC pill is compressed and, therefore, the resistances are changed. Quantum tunnel effect sensors deteriorate with use and, therefore, lose sensitivity. Nonetheless, tactile sensors based on quantum tunnel composites have been integrated in some robots such as older versions of the Shadow robot hand [34].



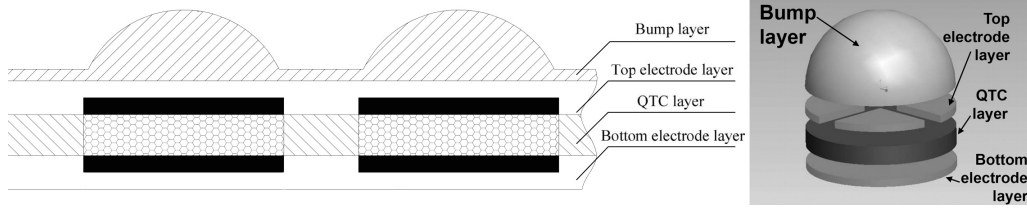


Figure 2.4: Conceptual scheme of the cells of a three-axis quantum tunnel effect sensor [3].

- Optical sensors. The optical reflection between elements with different refractive indices is used in optical sensing to measure the pressure applied. Typically, photo detectors measure the light intensity generated by an array of light transmitters, for example LEDs, which is proportional to the magnitude of the pressure applied [35]. Moreover, some optical sensor designs are sensitive to shear forces [36] and multi-degree-of-freedom forces [37]. Although optical sensors suffer from high energy consumption and computational cost [38], they provide good sensitivity, high repeatability and are immune to electromagnetic noise [39].
- Structure-borne sound tactile sensors. The term structure-borne sound refers to vibrations and waves spread in solid bodies. Accelerometers or microphones can be used to measure structure-borne sounds and detect contact with an object [40]. Although some approaches have been explored to use structure-borne sound tactile sensors during close proximity situations, for instance to estimate the distance to an object [41], they are only suitable for dynamic measurements.
- Sensors based on barometric measurements. This type of device contains a liquid that propagates the sound waves generated by contact microvibrations [42]. These sensors usually implement pressure trans-

### The BioTac

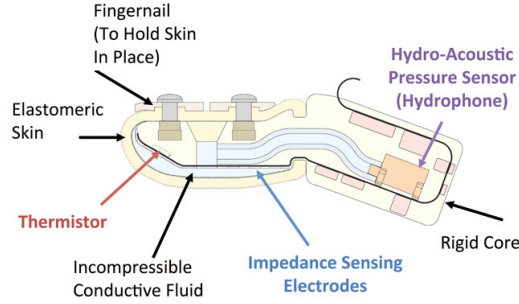


Figure 2.5: BioTAC hardware scheme [4].

ducers to measure the pressure in liquids inside the sensor [28]. Thus, sensors within this group provide high frequency responses and allow the sensors to be deformed. Additionally, some sensors based on barometric measurements have been implemented in a low cost manner using non liquid materials such as silicon rubber and a barometer [43]. However, they have low frequency responses and, therefore, it is preferred to use a liquid as a propagation media when a high frequency response (e.g 10-1040 Hz) is desired.

- Multimodal tactile sensors provide information about different cutaneous sensing modalities [42], such as microvibrations, forces or thermal properties, to emulate human hand fingertips [44]. State-of-the-art multimodal tactile sensors contain static pressure distribution arrays, thermal, proximity and dynamic tactile sensors [26].

Until recently, tactile sensing hardware was built *ad hoc* by researchers in the field but the development of commercially available sensors opened a window of opportunity to further investigate this sensing modality. Nowa-

days, a number of tactile sensors are commercially available [26]. Among them, it is worth highlighting the BioTAC sensor [4, 42, 45], which has been the experimental platform for this research. The BioTAC is a biomimetic sensor designed to provide multimodal tactile information. According to the taxonomy listed above, the BioTAC consists of a multimodal design that embeds a barometric measurement based sensor. The BioTAC (Syntouch, Los Angeles, CA, USA) consists of a rigid core containing all the required electronics and surrounded by an elastic skin, which is filled with a conductive fluid (see Figure 2.5). The BioTAC enables detection of three sensory modalities. Firstly, the fluid is deformed when forces are applied to its skin. The deformation of the sensor skin causes changes in the impedance perceived by an array of electrodes built into the surface of the core. Secondly, vibrations are detected by a hydro-acoustic pressure transducer, which is installed inside the core. Thirdly, thermal properties are measured by a thermistor installed in the BioTAC's tip. By detecting 19 impedance measurements, absolute pressure and temperature, vibration, and dynamic temperature, the BioTAC tactile sensor presents an accurate set of contact data from different tactile modalities.

## 2.4 An Overview of Robotic Tactile Sensing Applications

During the last decades, the development of new tactile sensing hardware and the subsequent appearance of commercially-available tactile sensors has led to significant developments of a number of applications of robotic tactile sensing. Similar to humans, tactile sensing can provide information that is inaccessible using other contact-less sensing modalities. Hence, tactile sensing

has been used to enhance the performance of robots in contact localization, identification of object properties, object recognition, robot control, slip detection and prediction, and grasping. An overview of the existing literature of tactile sensing applications in robotics is presented in this section.

### 2.4.1 Contact Localization

A fundamental application of tactile sensing in robots is contact detection and location. The advantages of tactile sensing for locating the contact during manipulation tasks (i.e. rotation and translation of objects) were shown in [46]. The authors compared the use of tactile sensing for contact location with other force-torque based approaches and concluded that, although both solutions are valid, tactile sensing was immune to problems caused by calibration inaccuracies, transient forces, and low grasp force. Further research has been subsequently explored in order to improve contact location systems using tactile sensing [47–53]. Among these works it is worth mentioning [51] and [52], which proposed a Bayesian perception system for contact location using tactile sensing. Interestingly, their approach achieved better perceptual accuracy than the available sensor resolution (i.e. hyperacuity) by interpolating between taxel positions.

These works explored contact location between the tactile sensor and the object. A recent area of research interest is locating the contact between two objects; for instance, when using a hammer to pound down a nail. In [54] tactile sensing was used to estimate the location of contact between a grasped object and the environment using force and vibration data obtained from a BioTAC sensor. Three data-driven approaches (i.e. Artificial Neural Networks, Gaussian Processes, and Support Vector Machines) were used to detect the positions and directions of the contact points with an error of

2.5cm/10°.

### 2.4.2 Tactile Sensing and Control

The perception of incipient physical stimulus through tactile sensing is being explored for robot control. One of the pioneering works in tactile sensing for robotic manipulation was presented in [55]. The authors implemented a control system for pushing objects that rest on a surface along a desired trajectory using only tactile feedback. A control framework for event-based dexterous manipulation using tactile sensing was proposed in [56]. Their proposed approach relies on the changes in contact conditions between the grasped object and either the robot or the external environment to achieve smooth transitions between manipulation tasks. Despite the fact that the tactile sensors used were rudimentary in comparison with today's state-of-the-art sensors, the authors showed the potential of using tactile sensing to make smooth transitions between different manipulation tasks. Another example of early research in robot manipulation approaches using only tactile sensing was presented in [57]. Using an *ad-hoc* tactile sensor [58] to estimate the contact location, the authors proposed a control system that requires no prior information of the grasped object and prevents object rolling during manipulation.

Although previous works have presented diverse techniques to control a robot's end-effector using tactile feedback, [59] introduced the term "tactile servo" to refer to this practice. The authors relied on different moments of the tactile sensor image to track edges and manipulate a pin rolling using a PUMA 260 industrial robot manipulator. In [60] a control framework is proposed for tactile servoing using a PID controller. The authors used a 16x16 tactile sensor array mounted in a KUKA lightweight robot (LWR)

to perform a series of tactile exploration experiments. By simply adjusting a projector matrix that sets what tactile servoing primitives should be applied, their control framework was used to perform a number of tasks such as tracking the contact point, a cable, the edge of an object, exploring the shape of an unknown object, and controlling the area of contact between the finger tip and the object. In [61] a system to insert a USB stick in a mating hole (the well known peg-in-hole problem) using tactile images from a GelSight optical sensor and a Rethink Robotics Baxter robot was implemented. Recently, [62] combined active perception and haptic exploration to perceive surface features of objects such as edges on ridges. The authors implemented their tactile servoing exploration system to follow the edge of circular disks and to follow ridges with different shapes using a 6-DoF robot arm with a TacTip tactile fingertip mounted as the end effector. In [63] a Schunk SDH-2 hand attached to a KUKA LWR was used to perform a set of exploratory movements using a tool covered with a tactile matrix. The authors present a technique to obtain the homogeneous transformation of a grasped tool object. The learned transformation was used with the robot kinematics to make a new kinematic chain, which was used for tracking the edge of an unknown object.

Another work on tactile servo was presented in [64]. Zero-Moment pressure (ZMP) features were used to control the rotation of the end-effector around two axes and a novel inverse tactile Jacobian matrix was used to generalize a hybrid (i.e. tactile-position) controller for different tactile servoing tasks. Their approach was tested in two robots, a KUKA LWR and a pneumatic Shadow arm with a Shadow robot hand attached, both equipped with 6x14 Weiss Robotics WTS0614 tactile arrays to follow edges, roll a cylinder and keep a plate in balance. The performance of a novel 6-axis force/torque

sensor installed under a rubber skin was tested for tactile servoing tasks in [53]. Although their proposed approach to follow the contour of a surface was prone to getting stuck in local minima, the authors defined an interesting set of algorithms to contact sensing (i.e. contact location on the fingertip, the friction, normal forces, and the local torque) using their novel tactile sensor.

The integration of visual and tactile sensing information has been also explored in servoing tasks. A control framework was implemented in two KUKA LWR arms to combine tactile and visual servoing [65]. A robot equipped with tactile sensor arrays maintained desired contact patterns during surface exploration while the pose of an unknown object was monitored using computer vision. Another control framework using tactile and visual sensing was presented in [66]. The authors tested their dexterous manipulation framework on a real manipulator robot and implemented a position control using visual features in combination with fingertip contact force control. Precise in-hand location of a tool was achieved using computer vision in [67] to perform a scraping task with a spatula in an altered environment using Dynamic Motion Primitives (DMPs) and tactile feedback. The authors implemented their approach in a real 7-DoF KUKA LWR mounting a low-cost robot gripper equipped with tactile sensors. Probabilistic hierarchical object representations were used for object localization and a motion controller was learnt through a reinforcement learning approach that used tactile feedback.

More applications of tactile sensing for robot manipulation were presented in [68]. Their work proposed progressively refining the estimate of an object's pose using tactile data and their proposed approach was used to manipulate a box and to grasp a door handle. Despite the technique to enhance object perception during manipulation presented in this work being interesting, it is

worth noting that the number of sides of the object model linearly increases the time needed for the algorithm to achieve high precision and, therefore, there is a trade-off between estimation precision and run time.

### 2.4.3 Slip Detection and Prediction

Although humans can distinguish incipient events during the manipulation of objects through tactile sensing, for instance slippage, this is still difficult for robots. Incipient slippage during the manipulation of objects may result in breaking the contact and the object falling. For this reason slip detection and prediction techniques have been explored in order to allow the robot to ensure robust grasps during manipulation [69–72] .

In [69] an approach for on-line estimation of the friction coefficient during contact is presented. The authors proposed an adaptive strategy for slipping avoidance during object manipulation using the friction coefficient and the Kalman Filter residual, i.e. the difference between the observed value and the estimated value. The work in [71] used the measures obtained from an array of dynamic tactile sensors to classify slippage between the robot and the object. The authors ran a classification algorithm that constantly distinguishes between hand/object and object/world slips to prevent incipient contact breaking during grasping and manipulation. Although promising, the work in [71] suffers from occasional false positives when no slip occurs and noise is sensed by the tactile arrays. Three random forest classifiers were used in [70] to create generalizable slip predictors for three different look-ahead periods, i.e. 0.01, 0.015, and 0.02 seconds. Using the output of these classifiers in a feedback loop, an object stabilization controller counteracts incipient slip events and keeps the grasp of unknown objects stable.

A Barret hand and BioTAC tactile sensors were used in [72] to estimate



the finger contact forces. Different machine learning methods were compared when classifying slip events and the authors concluded that a 3-layer Neural Network showed better performance than Locally Weighted Projection Regression (LWPR) and an analytical approach using linear regression. A grip force controller was also designed to maintain a stable grasp relying on tactile information and the output of the classifier. With similar purpose, an ad-hoc three-axis tactile sensor mounted on the fingertips of a multi-fingered humanoid robot arm was used in [73] to adjust grasp pressure using the perceived normal and shear forces as inputs for a control system. Although the authors also presented an approach to determine the object's hardness during object manipulation, it is limited to binary detection, i.e. an object being hard or soft.

#### 2.4.4 Object Recognition

Tactile sensing has also been used for object recognition [74]. In [75] an active exploration/touch strategy for learning and identifying objects using a tactile skin was implemented. The authors proposed a two-step approach where initially a robot explores the workspace and then learns the physical properties (i.e. surface texture, stiffness, and thermal conductivity) of the objects using active touch to minimise the number of exploratory movements. Similarly, a series of grasps was performed in [76] to identify objects. Using low-resolution intensity images obtained from tactile sensors, a feature vocabulary was learnt using  $k$ -means clustering. The authors used a bag-of-features classifier [77] to identify the object among a group of 21 trained objects. In order to minimise the number of grasps required for identification, a decision framework based on the reduction of uncertainty in the estimation of the object was proposed in [76].

Object recognition using tactile sensing has also been explored in combination with visual information [78]. A multivariate-time-series model of the tactile sequence and the covariance descriptor of the image were used to learn 18 household objects using  $k$ -Nearest Neighbours and Kernel Sparse Coding methods. Promising results show that this method, named Joint Group Kernel Sparse Coding (JGKSC), outperforms other methods evaluated and this work concludes that combining tactile and visual information enhances object identification.

### 2.4.5 Identification of Object Properties

Furthermore, robotic tactile sensing has been explored to identify the physical properties of an object. The works in [79] and [80] implement a series of algorithms to enable BioTAC tactile sensors to quantify an object's compliance, thermal properties, roughness and fineness of its texture. [61] used high-resolution tactile maps obtained from a GelSight optical sensor [81] to localize the pose of small objects grasped by the robot hand. [73] estimated the hardness of an object by using the normal and shear forces measured by an ad-hoc tactile sensor. A switching velocity-force controller implemented in a Willow Garage PR2 was presented in [82] to estimate an object's deformation properties during grasping and detect the internal state of bottles (i.e. being full of empty) using the high frequency components of tactile information.

While most of these works extract the tactile information in a passive manner (i.e. without intentional exploratory movements), active tactile sensing methods (also called active touch procedures) use the motion of the robot to extract additional information [83, 84]. For instance, when detecting an object's edge sharpness static touch on the edge will provide partial infor-

mation compared with sliding it across the edge [85]. An active sensing method to gauge the compliance of various rubber samples using tactile information was proposed in [86]. A closed-loop control was implemented in a Barrett arm/hand system equipped with a BioTAC tactile sensor to perform a number of exploratory movements following an active touch procedure to estimate the compliance of an object. Another active touch procedure was proposed in [87]. Using tactile sensing and two sensimotor control strategies, autonomous object exploration for shape reconstruction was improved when compared with passive perception, i.e. data not affecting the motion. [88] presented a method to estimate the shape of an object by performing two in-hand exploratory movements (i.e. squeezing and re-grasping) using an iCub. A grasp stabilization controller was used to add robustness to a Kernel Regularized Least-Squares algorithm in order to learn an object's shape and softness.

In addition to all these works, surface material identification through tactile sensing has attracted significant research interest. A thorough literature review of robotic material identification through tactile sensing is presented in Section 2.5 as it is one of the main research areas explored in this work. Object grasping is another of the areas explored in this dissertation and, therefore, Section 2.6 details the state-of-the-art of robotic grasping systems using tactile sensing. Moreover, Section 2.7 reviews the existing approaches to pass objects between a robot and a human.

## 2.5 Material Identification

One of the most fundamental capabilities required to endow robots with object manipulation skills is material identification. The surface material of an

object provides information about some of its underlying properties, such as friction coefficient or elasticity, that are essential to perform reliable object grasping and manipulation. Material identification is usually performed by sliding the tactile sensor on the object surface and analysing the data obtained during the contact. Existing research on material identification relies on batch surface recognition approaches, i.e. a whole sliding movement of the sensor over the material surface has to be performed for the identification to occur. Although this limits the identification speed, several excellent works provided unimodal (vibration only) batch approaches to material identification using different types of tactile sensors and techniques. In a pioneering work in tactile sensing for surface recognition [89] a finger with a microphone was used to detect the vibration induced by 3D printed textured surfaces. The authors defined a set of features to characterise the vibration signal captured by the microphone such as the modal frequency and power, and the average vibration amplitude. The performance of the  $k$ -Nearest Neighbour ( $k$ NN) algorithm was evaluated using these features and the Fast Fourier Transform (FFT) of the raw signal projected through Principal Component Analysis (PCA). Although this work focuses on texture identification, not real material identification, it established the methodological approach of using features for material recognition using tactile sensing, which was followed by later works. The work presented in [90] uses a three-axial force sensor to classify 10 different paper types through two different techniques. In one approach, the most likely texture was selected by finding the minimum euclidean distance between the mean frequency spectrum of the evaluation set and training examples. A second approach used a set of five distinctive features (friction coefficient, mean, variance, kurtosis and spectrum slope of the vibration signal) as inputs to an Artificial Neural Network (ANN). The

paper shows that both methods were proven to be equally accurate and shows evidence of the usefulness of the defined features. An accelerometer based vibrotactile sensor attached to a fingertip was used in [91] to measure the changes in the norm of the acceleration vector and to create spectrotemporal histograms as features for classification. A set of five exploratory movements was performed to gather data from the material surfaces, and Support Vector Machines (SVM) and  $k$ NN were used to successfully identify 20 materials.

A tactile sensor measuring the strain applied on the finger surface is presented in [92]. After segmenting the input signal, removing the average and band-pass filtering, the authors extracted a set of features consisting of five peaks identified on the smoothed FFT profile. Combining these features with the average strain readings, the authors compared different machine learning approaches, which successfully differentiate between nine materials with high accuracy. Another comparison of machine learning algorithms for material identification based on tactile sensing is presented in [93], where two kernel methods (SVM and Regularised Least Square), and one neural network were used to classify pairs of materials based on the raw strain measurements of the sensor. Although the authors concluded that the SVM showed the best trade-off between classification accuracy and computational complexity, they also observed that the raw sensor signals did not provide a good discrimination performance compared to other works. [94] presents a tactile micro-sensor able to differentiate surfaces with spatial periods within a  $40\ \mu\text{m}$  difference. This sensor was used to classify textiles through a robotic finger that slid across the materials for two seconds. Using wavelet transforms, the evolution over time of features like the peak power was obtained and fed into a  $k$ NN classifier. The work in [95] presents a texture based material classification through SVMs, with a set of temporal domain features obtained

from one second time windows as input. Specifically, the components of the feature vector were obtained from the accelerometer in the finger, and whitened individually before feeding the classifier. This work splits the whole set of readings into short time intervals to obtain more training data, but the approach is still batch-based, since the authors do not exploit sequential information for classification.

Another accelerometer based fingertip texture recognition approach is presented in [96], where seven different fabrics are classified based on a mixture of temporal (acceleration variance) and frequency (power spectra) features. Since these features do not provide enough discriminative power on their own, the authors use a neural network with the FFT coefficients over a given frequency range as the input. A GelSight optical sensor was used in [97] to capture high-resolution tactile images of different materials. The authors extracted Local Binary Patterns (LBP) of several tactile images with different orientations for each texture and used them to identify new samples according to the distance between their histograms, i.e. Hellinger similarity metric [98]. Their method was capable of identifying 40 materials with high accuracy. However, its application in real time is compromised by the number of training samples used due to a trade-off between identification accuracy and computational cost.

Recently, [99] explored real time classification of eight materials using a soft three axis tactile sensor with a new set of features. Specifically, the mean value of the three dimensional vibration signal and the Frobenius norm of the covariance matrix were used to train and classify a cascade of binary SVM classifiers, grouping materials together in each classification step. Although their approach is fast and accurate it becomes computationally expensive for a large number of materials since the depth of the cascade and the number

of classifiers to train grows quickly. A flexible tactile skin attached to the fingers of a DLR Hand-II was used in [100] to discriminate 6 tubes made from different materials by stroking at them with the thumb and index fingers. The authors present a comparative study of different classification techniques and conclude that a convolutional deep learning network outperforms other classifiers such as  $k$ NN and SVM, especially when repositioning the tactile skin. In other recent work [101, 102] a custom made tactile sensor was used for material identification using only the heat transfer from the sensor to an object. The authors showed that, although changes on the experimental set-up (initial conditions, ambient temperature and contact duration) have an impact on the performance, a multi-class SVM can classify eleven materials with high accuracy.

Nearly all the works mentioned so far rely on vibration signals obtained from *ad hoc* sensors. However the appearance of the SynTouch BioTAC commercial fingertip has made tactile sensing widely accessible and provided the opportunity of performing multimodal sensing. Using this tactile sensor, a series of works [79, 80, 86] explored Bayesian exploratory movements to classify materials using vibration and temperature signals. Bayesian inference was used to perform different exploratory movements in order to increase the classification certainty. The classification itself was based on features extracted from the sequences and the external force needed to slide the fingertip across the material, i.e. information about the friction coefficient of the material and the BioTAC's rubber skin was included. The combination of several exploratory movements enabled a high material identification rate with a large range of materials. A novel approach for identify the surface texture of a grasped object was presented in [103]. The authors used a Shadow robotic hand equipped with five BioTAC tactile sensors to perform small

sliding motions of the fingertips during grasping. Although the system was designed for in-hand object recognition, their results showed that their approach was capable of identifying objects with the same geometrical shape but with different surface materials. Using the vibrations generated during motion and geometric information obtained from the fingertip deformation upon contact, a SVM classifier identified 10 objects with identical geometrical shape with high accuracy.

One of the first multimodal approaches to material identification is presented in [104] and [24] where raw temperature and vibration signals were projected through PCA and used as inputs to an ANN. The authors show that this classifier outperforms humans in similar experimental conditions. The approach presented in [105] distinguishes between 49 objects with high accuracy using multimodal data from 5 different object explorations. The authors performed a thorough analysis of tactile features found in the literature and concluded that simple descriptors, such as average values of the filtered signal, outperform more sophisticated feature extraction techniques.

All these works achieve good material recognition ratios using one or several batch readings and combinations of exploratory movements (see Table 2.1). However, some materials might have a characteristic texture or special thermal properties and might therefore be identified faster than others, i.e. using shorter readings. Moreover, combining these two modalities (i.e. vibration and temperature) might improve the identification accuracy, for instance, for materials with similar textures but different thermal properties. Chapter 4 presents a multimodal recursive identification approach to material identification through tactile sensing.



Table 2.1: State-of-the-art in tactile material identification.

Paper	Sensor	Data	Exploratory Movements	Materials	Accuracy
Simapov et.al. [91]	<i>Ad-hoc</i> vibrotactile sensor	Vibration	5	20	80%.
Kerr et.al. [24, 104]	BioTAC	Multimodal (vibration/temperature)	1	14	70.05%.
Baishya et.al. [100]	Grip VersaTek sensor 4256E	Spatio-temporal signal	1	6	97.3%.
Bhattacharjee, Wade et.al. [101, 102]	<i>Ad-hoc</i> temperature sensor	Heat transfer	1	11	98% in 1.5 secs. explorations.
Fishel, Xu et.al. [79, 80]	BioTAC	Multimodal (vibration, temperature/compliance)	Several Bayesian explorations	117	95.4%.
Hoelscher et.al. [105]	BioTAC	Multimodal (vibration/temperature/ compliance)	5	49	97.6%.
Kaboli et.al. [103]	BioTAC	Vibration	10 small sliding movements	20	96%.

## 2.6 Grasping

An essential skill for robotic object manipulation is grasping [106]. Grasping is a complex problem that requires advances in a number of areas, such as planning, control or object detection among others, for different types of end-effectors (e.g. dexterous hands and grippers). There are two strategies for the grasping problem in the existing literature: grasp planning and execution, and grasp control. While the first uses known models of the object that will be grasped, the latter relies on sensory feedback to find a stable grasp. Splitting the grasping into planning and execution tasks allows for the estimation of a stable grasp configuration off-line, computation of collision free trajectories using a path planner, and then movement of the robot towards the grasping pose. Significant research has been conducted in grasp planning during the last three decades [107–110] in order to generate hand configurations that ensure a precise grip of an object. One of the problems of planning and execution strategies is that the success of the execution stage depends on the planning process being accurate, which assumes that a perfect model of the object and the environment is available. However, uncertainties on the object pose (i.e. position and orientation) and shape are very frequent when perceiving the object through contact-less modalities. Tactile sensing feedback has been used to adjust the contact points locally when touching the object surface and, thus, correct the inaccuracies caused by uncertain object shape and pose estimation (often computed using vision) during the planning stage [111–116].

A capacitive contact sensor (i.e. 22 cells array) was used in [111] to grasp objects using a PR2 gripper. Their approach first computes where to grasp the object using vision (i.e. grasp planning) and subsequently corrected the

initial estimation using tactile feedback until a robust grasp is obtained. Although their experimental results show that the robot was capable of grasping 30 objects with different shapes, it is limited to robot grippers and requires several explorations before achieving a robust grasp. Another grasping system using the PR2 gripper was presented in [40]. The authors implemented a position control to pick up objects using only one-point pressure signals (i.e. no visual information). Their system assumed the grasp location was provided (i.e. using the approach in [111]) and proposed a framework to perform a number of manipulation tasks such as close, load, lift and hold, replace, unload, and open objects. The approach presented in [117] adjusts the contact configuration using tactile sensing to obtain local object geometry measurements at the contacts. Their null-space grasp control uses a non-linear control strategy that slides the contacts over the object surface until convergence to unit frictional equilibrium, i.e. a special case of force-closure grasp.

The work in [113] proposed a data driven approach to evaluate the viability of the grasp before performing it. During grasp execution, contact information was exploited to update the object model used for planning. Thus, the system can predict an unstable grasp using a SVM classifier and then plan a new grasp using reconstructed local geometry from tactile sensing to grip objects using the fingertips (i.e. precision grasp). [114] presented other grasp adaptation framework using tactile sensing. A stability estimator was implemented using two non-linear classifiers (GMM and SVM) to solve a one-class classification problem, i.e. stable data only. A precision grasp was predicted as stable if its GMM likelihood or the SVM classifier output exceeds a given threshold. The output of the stability estimator was used to adapt the grasp stiffness when the grasp was unstable. Their approach was

tested in an Allegro hand equipped with three BioTAC tactile sensors and proved to adapt to physical uncertainties during grasping.

An approach to grasp objects under position uncertainty was proposed in [115]. Although no tactile sensing was used, the authors used a joint torque sensor to detect contact during the grasp approach. A reactive compliant control strategy was used to soften the impact and then adjust the grasp position using a virtual spatial spring framework. An approach to perform precision grasps under shape uncertainty was presented in [116]. A control strategy to achieve smooth transitions between position and force control for grasp execution was implemented in two robots: an Allegro hand equipped with BioTAC tactile sensors and a Barret hand. The authors rely on their previous work to estimate the force using a data driven approach [114], and their compliant closing-finger controller implements the transition between controllers as a function of the distance between the finger position and the output position of the grasp planner. Besides dealing with uncertainties, tactile sensing was used in [112] to maintain the contact between the fingers and a grasped object, thus, avoiding losing the grasp due to perturbation forces arising on the object during manipulation. A Barret hand equipped with tactile pressure sensors was used to increase the joint torques applied to the fingers when contact is below a given threshold. Their results show that the algorithm avoids contact breaking or slippage of objects although its centre of gravity changes during manipulation.

On the other hand, other approaches treat grasping exclusively as a control problem [113]. Those methods aim to reach the desired contact configuration between the robot and the object using sensing data feedback. Although humans rely heavily on tactile sensing when grasping and manipulating objects [20], many grasp control approaches use mixed position/force

control [106, 118] or impedance control [119, 120]. This thesis focuses on the applications of tactile sensing for grasp control strategies.

One of the first attempts to grasp objects using a grasp control strategy with tactile sensing was presented in [121]. The authors implement a grasp and lift algorithm in a two-fingered manipulator and used three controllers (position, force, and stiffness) during the different stages of the experimental procedure. Although their results were preliminary, the authors concluded that tactile and force sensing improves grasping and smooths movement transitions during manipulation. A novel approach for combining position control and fingertip force control in a three fingered custom robotic hand was presented in [122]. The position control guides the hand closing stage and the force controller performs the grasp. External force limitation and collision systems were designed to switch both controllers smoothly. Additionally, an adaptive grasp force control adjusts online the initial grasp force estimation according to the slippage measured using tactile sensing.

The work in [123] implemented a grasp controller for a Barret Hand equipped with a force/torque sensor and pressure sensors on the fingertips. In order to adapt to inaccuracies on the object's expected conditions, a number of grasp primitives were used to find a stable grasp before applying pressure on the object. [124] combined visual and tactile sensing to grasp a rod using the gripper of a Zebra-Zero manipulator. Using tactile feedback and motion primitives, a gripper was controlled to point towards a single axis while aligning with the object. A system to increase the number of points of contact between a fingertip tactile sensor and an object was presented in [125]. Their proposed approach adapts the contact of the fingertips to grasp objects without the need of a model for the environment or the object. The grasp control strategy in [126] used tactile sensing to centre the robot gripper joints over an

object. A controller for object reaching was implemented using the relative positions between the object and the robot, obtained using vision. Then, the authors use a self-organized neural network to learn a mapping from tactile images centroids to displacements that were used during the grasping phase to drive the robot.

Some researchers performed object exploratory procedures to find a valid grasp configuration without previous grasp planning. [127] approached the grasping and manipulation problems relying solely on tactile and force feedback and a set of exploratory movements. The experimental platform was a three fingered robotic hand totally covered with *ad-hoc* tactile sensors consisting of a Hall-effect (proximity) sensor with a rubber coating. Motion primitives were used to guide the hand until the object and the palm were in contact. Although no tactile feedback was used for the grasping, experimental results show that low impedance values allowed the fingers to adapt to different object shapes. Moreover, the authors concluded that tactile data collected during the manipulation of an object implicitly carries information regarding its shape.

The work in [128] grabs a curved object placed on a table by rolling a pair of fingers on the object's surface. Tactile sensing was used to detect the points of contact and, therefore, localizing the fingers with respect to the object. When two antipodal contacts are achieved, the grasp control strategy uses the fingers to apply forces along inward normal directions. Although their approach was proven to work in simulation, no experiments on real robots were presented and the object rolling solution for grasping may require several explorations.

Another approach to guide the search for contact locations was presented in [129]. Their grasp strategy switches between two controllers according to

the local curvature of the object. Despite providing a valid control strategy, the gradient following algorithm is prone to getting stuck into local minima. A series of works in [130] and [131] proposed a tactile exploration strategy to find stable configurations to grasp objects using multifinger robot hands. Their approach explored an unknown object to find opposite object faces that could be grasped, discarding impossible grasps and selecting the grasp affordance with the highest score according to four geometric features: parallelism, minimum face size, mutual visibility, and face distance. However, their approach was only tested on simulation and required a previous estimation of the object size, position, and orientation.

Grasp synthesis (i.e. finding a suitable set of contacts given an object and a set of constraints) was presented in [132] in terms of contact relative motions (i.e. units of control where the contacts move from one contact configuration to another) and, thus, re-state grasping as an optimal control problem. Their approach finds the strategy that minimises the number of steps required to grasp an object. The problem was stated as a  $k$ -order Markov Decision Process and solved it using Reinforcement Learning on a NASA-JSC space humanoid robot. Although sometimes their approach was capable of synthesising the grasp only in one step, some objects and initial configurations required a sequence of contact relative motions.

A summary of the state-of-the-art in tactile sensing for robotic grasping is shown in Table 2.2. Although most of these works present valid grasp control approaches, they either use force sensors or rely on data-driven approaches to estimate the contact forces. In this thesis we present an approach for adaptive control grasping using a contact model of the BioTAC fingertips. The proposed approach allows to grasp a significant number of objects with different shapes by only knowing their approximate location, i.e. without

Table 2.2: State-of-the-art in tactile sensing for robotic grasping.

Paper	Approach	Description
Hsiao et.al [111]	Planning/Execution + Control	Improves initial grasp estimation using tactile feedback to grip objects of different shapes with a PR2 Robot.
Corrales et.al [112]	Planning/Execution	Avoids contact breaking or slippage during manipulation.
Dang et.al. [113]	Planning/Execution + Control	Improves the initial planning using tactile sensing.
Li et.al [114]	Grasp adaptation framework	Data driven stability estimation to perform precision grasps.
Platt Jr. et.al. [117]	Null-space grasp control	Tactile sensing for contact configuration adjustment.
Felip et.al [123]	Control	Finds an stable grasp using grasp primitives.
Sommer [125]	Control	Increases the points of contact without having models of the object and environment.
Bierbaum et.al [130, 131]	Control	Exploratory procedures to grasp unknown objects by finding opposite faces.
Platt [132]	Optimal control	Reinforcement Learning to minimise the number of explorations needed to grasp the object.



grasp planning. Chapter 5 presents the methods and experimental results of the proposed approach to grasp unknown objects using a custom cartesian position/tactile-force controller and a Microsoft Kinect camera.

## 2.7 Object Handover

Object handover is a fundamental skill to endow robots with the ability to collaborate with humans in everyday tasks. Human-robot object handover involves many complex aspects such as human and object safety, social and handling context, grasping stability, slip detection, and ergonomics. Huge research efforts have been devoted to endow robots with the skills required for sharing objects, working and collaborating with humans. However, there is still a need for safe, smooth, and reliable interaction in any combined task.

Humans show a high degree of adaptability when exchanging objects with a robot [133]. Notwithstanding, to ensure the safety of the robotic hand and the object, robot-human object handovers typically aim to facilitate the task for the human [134]. Some efforts have been devoted to study human preferences during robot-human handover. After conducting a survey with human subjects, [135] concluded that the participants preferred different approaches for a robot handing over objects (i.e for instance placing the object on a table or the robot approaching the human for deliver the object) depending on their current activity or the scenario in which it takes place. [136] showed human receivers have a preference for the robot arm to be extended, the object to be delivered being visible, and the object handle in a default orientation (i.e. straight up).

Another study of the handover process from the receiver's perspective was presented in [137]. A biomechanic model of the receiver was used to

find an obstacle-free handover position that allowed the receiver to reach the object in an optimal manner considering the receivers height, weight, range of motion, and strength. Additionally, [137] showed that the electromyography (EMG) signal of a human receiver during a series of trials showed a decrease in the effort exerted during the handover when the robot picked the handover location using their algorithm, i.e. less muscle activity. Their algorithm was later used in [138] for assisting amyotrophic lateral sclerosis patients by using their electronic health record. The height, weight, and gender of the patients were used to customise a patient recognition system and the object handover position selection.

A fundamental problem in safe robot-human handover is deciding when to release the grasp and, thus, allow the object transfer. Since a robotic hand has to maintain a stable grasp on the object until the human is ready to hold it, the approach in [139] considered a tight relationship between object handover and grasp stability to trigger the object release. A three fingered robotic hand released an object when the forces applied on the object induced a change in a number of grasp stability metrics. In particular, the authors used a combination of joint angles, contact, kinematic, and dynamic stability indices of the hand to decide whether to release the object (robot-human handover), to hold it (human-robot handover), or to re-grasp it using a new finger configuration. The approach presented in [140] used a Kinect sensor to detect the receiver hand and the object, and released the object when both were detected as a single cluster of 3D points. Although the authors implemented a complete handover procedure, the system was not reliable since the object could be released even when the receiver was not grasping it. The work presented in [141] proposed a mechanism to hand over a drink, where the robot opened the gripper only if a human face was detected and

simultaneously the compliant hand of a PR2 was displaced by one centimetre in the vertical direction. Despite handing over the object in a secure manner, their approach requires the human receiver to pull the object strong enough to move the arm above the threshold, which impacts on the system responsiveness. The first effort to imitate the actual way humans handover objects is presented in [142]. Their experiments showed that humans adapt the grasping force according to the change in the estimated weight of the object. The authors used a two finger hand and force sensors to release the object according to its slippage, i.e. the tipping point on the Coulomb force.

A thorough analysis of human handover was presented in [143] and their results were subsequently used in [144] to implement a release controller on a PR2 robot. Their human-inspired handover system controlled the grip force of the robot according to the weight of the obstacle the robot perceived in the wrist. Moreover, the authors found a user preference for the human-inspired controller when compared to four other handover controllers for quick release and constant grip forces. Another approach relying on the sensed load force was presented in [145], which implements a grip force controller based on the feedback of a force/torque sensor installed on a KUKA LWR robot with an Allegro hand attached. The authors found that the object handling occurs faster when using their controller compared with state-of-the art approaches. Furthermore, their approach significantly reduces the forces applied on the object by the robot and the human, resulting in fast and smooth handovers.

Although most of these works rely on some force estimate acting upon the object, they all assume the handover is going to take place without any problem. In a recent work [146] a system consisting of acceleration and force sensors mounted on a gripper was used to ensure fail-safe handovers. The authors compared the grip force with the sum of forces applied to the

object for a given static friction coefficient, and implemented a controller for re-grasping if the object's downward acceleration exceeds a given threshold.

To endow robots with the ability of safely interacting in complex situations, such as a workshop or an operating room, the control system should ensure smooth and reliable handovers even if the human cannot securely grasp the object. Performing reliable robot to human object handover requires a system capable of adapting against uncertain events and perturbations that are not meant to end in a handover such as a receiver's unsecured grasping or collisions. In situations like these, the robot should be able to keep itself and the object safe. Avoiding damage to the hand and the object falling is an extremely complex problem which requires quick readjustment of the fingers to maintain a stable grasping during a potentially large perturbation. In addition, a multi-fingered grasp may not be manipulable as forces during an unintended handover could make the object move in a direction that the robot cannot reach if it does not have enough degrees of freedom. Under these circumstances the robot could be damaged and result in the object dropping. However, state-of-the-art approaches often assume that there are no perturbation forces applied on the object during the handover, i.e. see Table 2.3 for a summary of robotic handover state-of-the-art approaches. This thesis contributes to reliable object handover by presenting a tactile sensing based handover system (see Chapter 6) that ensures neither the robot nor the object are damaged. We achieve this by adapting the fingers to force perturbations on the object and releasing only if the human is ready to hold the object.

Table 2.3: State-of-the-art in robotic object handover.

Paper	Release detection	Sensing Modality	Reactive to perturbations
Nagata et.al. [139]	The grasp is not stable	Kinaesthetic (Force/torque) and Tactile sensing	No.
Aleotti et.al. [140]	Object and receiver are perceived as a single cluster	3D vision	No.
Bohren et.al. [141]	Face recognition + PR2 compliant hand displaced above a threshold	Vision and Kinaesthetic (Force/torque)	Compliant hand.
Kim et.al. [142]	Changes in the weight to adapt grasping force	Kinaesthetic (Force/torque)	No.
Chan et.al. [143, 144]	Changes in the object weight perceived in the wrist	Kinaesthetic (Force/torque)	No.
Medina-Hernandez et.al. [145]	Smooth handovers through grip force control	Kinaesthetic (Force/torque)	No.
Parastegari et.al [146]	Perceived forces on the object	Kinaesthetic (Force/torque + Acceleration)	Re-grasping when the object is falling.

## 2.8 Summary

The combination of diverse stimulus modalities (visual, hearing, and tactile) allows humans to build an accurate representation of their surroundings and solve potential perceptual conflicts, i.e. to make decisions and to act under uncertain or incomplete information. Similarly, the integration of a range of sensing modalities is a must in order to develop fully autonomous robots that are capable of interacting with complex environments. The advent of commercially available tactile sensors has attracted increasing research interest of the robotics community towards endowing robots with touch sensing and combining it with other sensing modalities. However, advances in the development of robotic tactile sensing are still well behind other modalities like proximity sensing or vision systems.

Despite tactile sensing being inherently continuous, many existing approaches in the literature treat it as an episodic process. In particular, the recognition of objects and their properties is normally not addressed as a continuous process. Episodic tactile sensing limits the accuracy and restricts identification speed of the robotic systems as they rely only on the latest sensor measurements, i.e. discarding previous information. Moreover, although a number of tactile sensing modalities exist (i.e. vibration, temperature, and pressure) state-of-the-art systems rarely take advantage of data from a variety of sensors. The combination of several tactile sensing modalities can enhance the sense of touch in robots. For instance, an object can be difficult to distinguish by its shape, but it may have characteristic thermal or microgeometric properties.

Different to tactile object recognition, tactile sensing is generally treated as a continuous process in grasping and manipulation. However, the inte-

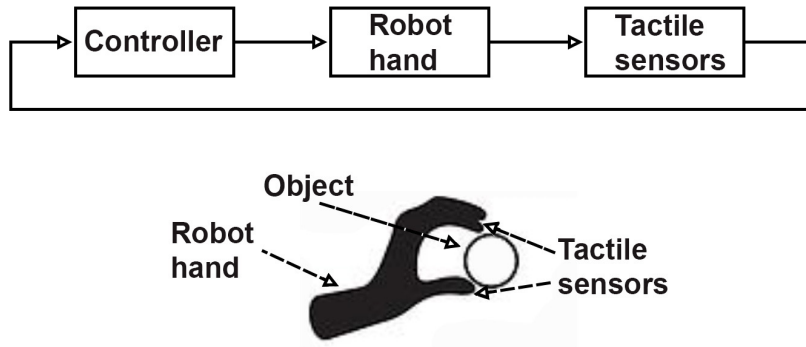


Figure 2.6: Generic definition of tactile sensing solutions for robotic grasping and object handover.

gration of continuous tactile sensing to enhance the reliability of the robotic systems needs to be explored for many applications. Robotic grasping of objects is still a challenging task that usually requires *a priori* knowledge of the object. When this information is not available, a robot could rely on tactile sensing to perform the grasp in a similar manner to that of a human.

Another unexplored application of tactile sensing is to enhance the reliability of robots in uncontrolled environments. Continuous adaptation to perturbations or unexpected events is fundamental in order to endow robots with the capability to interact in human environments. For instance, when grasping or handing over objects between a robot and a human, perturbations on the object may damage the robot and the object. Tactile sensing can be used to quickly react to unexpected events and guarantee the object is delivered in a secure manner. Figure 2.6 shows a generic definition for robotic grasping and handover systems using tactile sensing.

Consequently, this thesis will combine robotics with signal processing, machine learning, and control theory techniques to tackle the following issues that have not been addressed in the existing literature:

- Design of a continuous identification system that allows integration of various sensing modalities and its implementation for object surface material identification.
- Design and implementation of a tactile contact model and a precision grasp control that allows object grasping when information of the object geometry is not available.
- Design and implementation of an algorithm using robotic tactile sensing to ensure that a robot hands over objects to humans in a reliable manner.

Chapter 3 reviews the methods used in this thesis and describes the robotic platform in which the algorithms have been implemented. This will provide the foundations to address the issues stated above in Chapters 4, 5, and 6.



## Chapter 3

# Robotic platform & methods

### 3.1 Introduction

This chapter presents a systematic review of the robotic platform used in Chapters 5 and 6 and the methods used during the research described in Chapters 4, 5 and 6. Although this thesis focuses in robotic tactile sensing and, in particular, its applications in perception and manipulation, the techniques used in this research can be grouped into four main areas: Machine Learning, Signal Processing, Robotics, and Control Systems.

The rest of the chapter is organised as follows. A detailed description of the robotic platform is provided in Section 3.2. Section 3.3 will detail the Machine Learning techniques used in this thesis. Section 3.4 will provide insights of the techniques used for signal processing. A review of the methods used to control the robotic platform used in this research will be detailed in Section 3.5. Finally, Section 3.6 concludes this chapter and summarises the robotic platform and methods detailed in this chapter.

## 3.2 Robotic Platform: The Shadow Dexterous Hand

In dexterous manipulation a number of mechanisms (i.e. fingers) collaborate to grasp and manipulate an object [147]. Considering the human dexterous manipulation skills it seems logical that most of the robot hands designed for this purpose are anthropomorphic. The Shadow Dexterous hand is a humanoid robotic hand build by the Shadow Robot Company. The robot is similar in shape (see Figure 3.2) and size to an average human hand.

The standard tendon driven Shadow hand has 20 actuated degrees of freedom (DoF) and a further 4 under-actuated movements using a total of 24 joints. This range of DoF is very close to the 27 DoF a human hand has [148] and, together with its anthropomorphic shape, allows the robotic hand to manipulate objects in a human-like manner.

Although some versions of the Shadow hand mount only three fingers, the Shadow hand used in this work has 5 fingers. In this version of the robot, the first, middle and ring finger have three actuated joints plus one under-actuated i.e. joint 1. The little finger includes an additional joint (i.e. joint 5) that allows it to rotate towards the thumb. While the design is similar for the first, middle, ring and little finger, with the exception of the additional joint on the little finger, the thumb implements a totally different design and all its 5 joints are actuated. Additionally, the wrist has two joints that allow the hand to move forward, backwards, and in lateral directions. A complete scheme of the links and joints of a five fingered Shadow hand is shown in Figure 3.1.

As shown in Figure 3.2, the Shadow hand used in this research has been modified to be equipped with three BioTAC tactile sensors installed in the

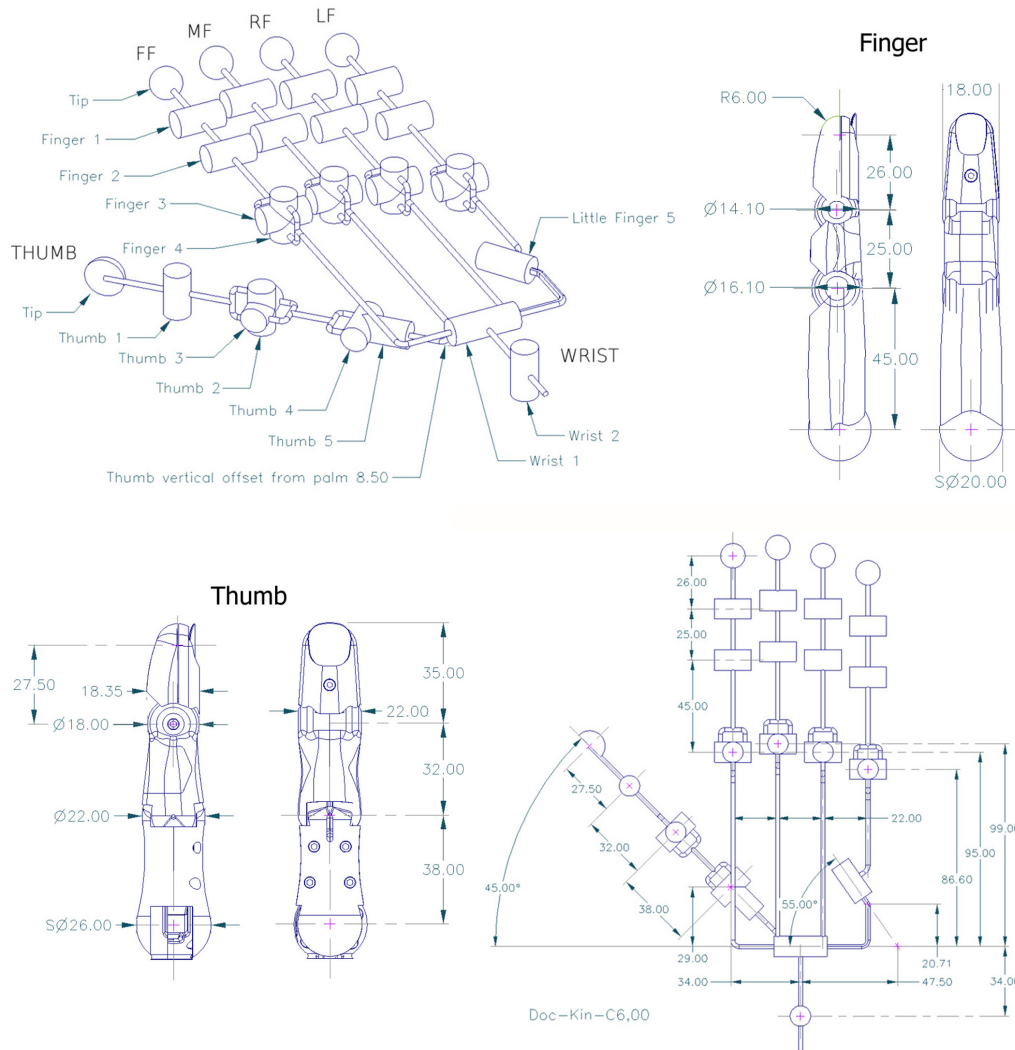


Figure 3.1: Scheme of the joints in a regular Shadow hand [5].

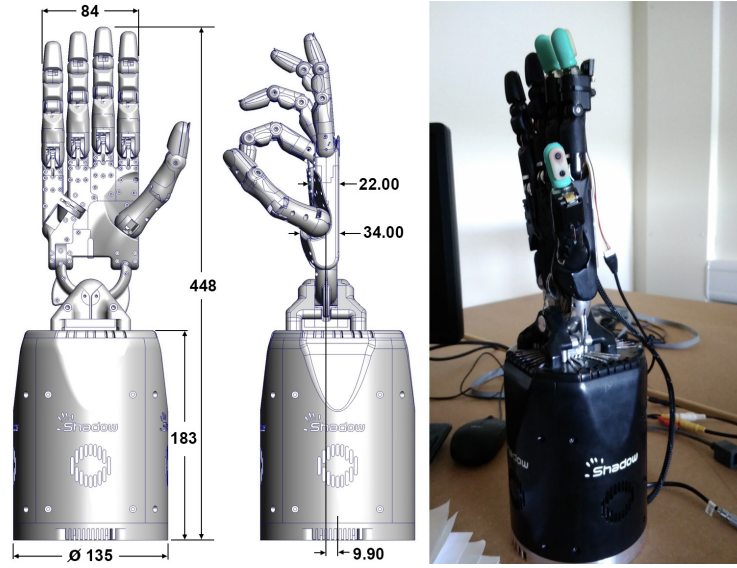


Figure 3.2: The Shadow robotic hand. On the left, robot dimensions scheme [5]. On the right, the Shadow hand equipped with BioTAC tactiles sensors on the thumb, first and middle fingers.

thumb, first and middle fingers. The fingertips were attached to the last joint (joint 1) of each of those fingers, which was disabled. Therefore, two under-actuated joints (i.e. in first and middle fingers) and one actuated joint (i.e. in the thumb), as well as their corresponding DoF, were lost because of the integration of the tactile sensors.

In addition to providing tactile sensing, the Shadow hand measures the angle of rotation of the joint by the use of a Hall effect sensor (i.e. transducer that responds to a magnetic field by varying its output voltage) mounted on each joint locally. The Shadow hand also provides data of the tension difference between both ends of each tendon. It is worth noting that, although described as force measurements in the Shadow Robotic Hand documentation [5], they are not the joint torques nor can they be directly used to calculate the wrenches applied at the end-effectors (see Section 3.5.2.1).

The Shadow hand implements two PID controllers that can be used to perform effort control or joint's position control. The first one controls the effort applied on the tendons to move each joint guaranteeing that the effort is within the safety constraints and implements a friction compensation system. The latter relies on the effort control to keep the rotation angle of the joints at certain values and, thus, controls their positions. During the research in this thesis all controllers have been implemented as an additional layer over the joint's effort controller.

Considering its dexterity and the integration of state-of-the-art tactile sensors, the Shadow hand is the ideal platform for conducting the research that is presented in this thesis.

### 3.3 Machine Learning

During this thesis, a number of supervised and unsupervised learning techniques are used for surface material identification and detecting perturbation force directions during object grasping. In particular, dimensionality reduction of tactile microvibration signals after their conversion to frequency domain is computed using Principal Component Analysis in Chapter 4. Linear regression is also used in Chapter 4 to approximate the thermal flux of materials upon contact with a robotic fingertip. The Expectation Maximization algorithm is used in both Chapters 4 and 6 to model the likelihood functions of perturbation force directions and surface materials. Finally, the approach presented in Chapter 4 is compared with three algorithms for data classification, which are commonly used in the state of the art (see Chapter 2). All these Machine Learning techniques were used in Chapters 4 and 6 and the input data were collected using the experimental platforms described in these

chapters. Examples of the datasets are displayed on Sections 4.2.1 and 6.5.3, i.e. for material identification and object handover respectively. In order to understand the rationale of the application of these techniques, this section presents a thorough revision of the Machine Learning algorithms used in this thesis.

### 3.3.1 Principal Component Analysis

When dealing with highly dimensional data it is useful to consider that, as the dimensionality of the space increase, the data become sparser. High-dimensional functions are likely to be more complicated than low-dimensional ones and, therefore, harder to deal with [149]. This problem is known as the curse of dimensionality. To illustrate the problematic nature of this sparsity we can address the problem of grouping data into similar clusters. If data lays on a high dimensional space, sparse data appear to be dissimilar and prevent clustering algorithms from being efficient. Therefore, sometimes it is useful to project the data to a lower dimensional subspace while minimizing the average reconstruction error.

One of the most used techniques for dimensionality reduction is Principal Components Analysis (PCA), also known as the Karhunen-Loève transform. Using this technique it is possible to find a lower dimensional projection of a dataset in which data are better represented by minimizing the sum-square error. Assuming we have a set of data points  $\mathbf{x}_i \in \mathbb{R}^n$  where  $i = 1, \dots, n$  following a multivariate Gaussian  $x \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ , we can define a data PCA projected matrix  $\mathbf{Y}$  as:

$$\mathbf{Y} = \mathbf{R}^\top (\mathbf{X} - \boldsymbol{\mu}), \quad (3.1)$$

where  $\mathbf{X} \in \mathbb{R}^n$  is input data,  $\boldsymbol{\mu}$  is a  $d$ -dimensional mean vector and  $\mathbf{R}$  is a  $n \times n$  rotation matrix. We could find a solution to PCA in terms of eigenvectors

of the covariance matrix of the input data. However, we also can generalise the notion of eigenvectors from square matrices to any kind of matrix using an alternative approach based on Singular Value Decomposition (SVD). Any real  $n \times d$  matrix  $\mathbf{M}$  can be decomposed using SVD as follows:

$$\mathbf{M} = \mathbf{U}\mathbf{D}\mathbf{V}^*, \quad (3.2)$$

where  $\mathbf{U}$  is an  $n \times n$  unitary matrix,  $\mathbf{D}$  is a  $n \times d$  diagonal matrix with non-negative real numbers on the diagonal, and  $\mathbf{V}^*$  denotes the complex conjugate transpose (Hermitian matrix) of the  $d \times d$  unitary matrix  $\mathbf{V}$ . Therefore, the main diagonal of  $\mathbf{D}$  contains the singular values  $\sigma_i \geq 0$ , the columns of  $\mathbf{U}$  are the left singular vectors, and the columns of  $\mathbf{V}$  are the right singular vectors. Assuming  $n > d$  and since there are at most  $d$  singular values, the last  $n - d$  columns of  $\mathbf{U}$  will be multiplied by 0, thus being irrelevant. In order to estimate the proportion of the variance (i.e. energy) that each eigenvector represents we could divide each eigenvalue of  $\mathbf{D}$  by the sum of all eigenvalues. Therefore, the rotation matrix  $\mathbf{R}$  can be obtained by truncating the size of  $\mathbf{U}$  according to the established energy value and the spectrum of eigenvectors in  $\mathbf{D}$ .

### 3.3.2 Expectation Maximization

Finding structure in a dataset is a fundamental problem of unsupervised learning. The main idea relies on organizing objects into groups whose members are similar according to some criteria. EM is an iterative algorithm that updates an initial estimation for a set of given parameters with  $k$  components  $\Theta = \{\alpha_1, \dots, \alpha_k, \theta_1, \dots, \theta_k\}$  by maximizing the likelihood within each cluster. Many variants of the algorithm can be found in the literature with different applications, for instance EM for Hidden Markov Models [150] or Variational

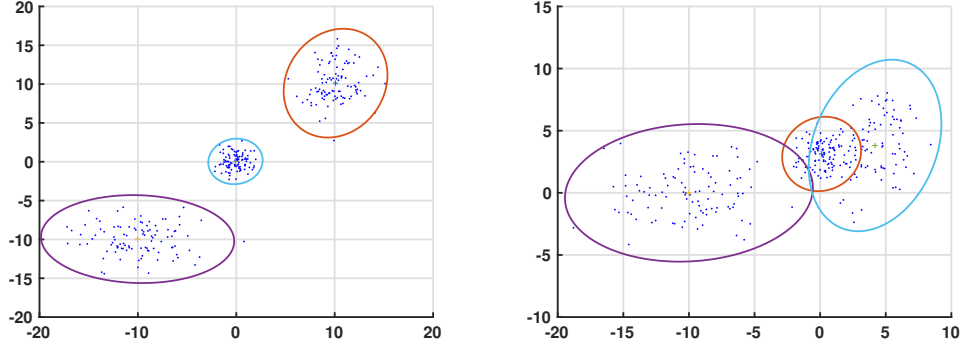


Figure 3.3: Examples of the EM algorithm running on different datasets. On the left the clusters are clearly separated while on the right the proximity of the clusters result in the Gaussians being overlapped.

Bayes EM [151]. In this section we focus on the EM for the estimation of Gaussian Mixture Models (GMMs). This method groups data into clusters that follow Normal probability distributions and finds the maximum likelihood of the parameters for a given dataset as shown in Figure 3.3. Hence, the parameters  $\theta$  for each Gaussian distribution in the mixture are defined by their mean  $\mu$  and covariance  $\Sigma$ . Each iteration performs an expectation (E) step and a maximization (M) step. The E-step creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters. The M-step computes parameters maximizing the expected log-likelihood found on the E-step, which will be used to determine the parameter values in the next iteration of the E-step.

For a given data set  $\mathbf{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  where  $\mathbf{x}_i$  is a  $d$  dimensional data vector i.i.d, from an underlying density  $p(\mathbf{x})$ . Then,  $p(\mathbf{x})$  is defined as a finite



mixture model with  $k$  components:

$$p(\mathbf{x}|\Theta) = \sum_{j=1}^k \alpha_j p(\mathbf{x}|\theta_j), \quad (3.3)$$

where  $p(\mathbf{x}|\theta_j)$  are the mixture components with parameters  $\theta_j$ , and  $\alpha_j$  are the mixture weights.

The EM algorithm can be initialized either by using some heuristic method to do an initial estimate [152] [150], such as k-means algorithm [153] [154], or choosing the initial parameters randomly [155]. In this work, better results were obtained using the second approach and selecting the initial parameters randomly from the data set  $\mathbf{D}$  instead of the vector space.

During the E-step, given the latest estimate of  $\Theta$ , the membership weights  $\omega_{ij}$  of all data points  $\mathbf{x}_i$  are computed for all mixture components such that  $\sum_{j=1}^k \omega_{ij} = 1$ . Specifically,

$$\omega_{ij} = \frac{p(\mathbf{x}_i|\theta_j)\alpha_j}{\sum_{m=1}^k p(\mathbf{x}_i|\theta_m)\alpha_m}, \quad 1 \leq j \leq k, \quad 1 \leq i \leq n. \quad (3.4)$$

The resultant  $n \times k$  matrix of membership weights  $\Omega$  is used during the M-step to update the current estimate of  $\Theta$ . For  $\alpha$  we have the sum of membership weights for each mixture component. Thus, the number of data points assigned to component  $k$  can be computed as the sum of each row in  $\Omega$  as:

$$n_j = \sum_{i=1}^N \omega_{ij}, \quad (3.5)$$

and used to obtain new mixture weights:

$$\alpha_j = \frac{n_j}{n}. \quad (3.6)$$

Additionally, we update the values of  $\theta_j$  and, since we aim to obtain a mixture of Gaussian distributions, new mean  $\mu_j$  and covariance  $\Sigma_j$  estimates

are computed for each component of the mixture using the weighted data as:

$$\boldsymbol{\mu}_j = \frac{1}{n_j} \sum_{i=1}^n \omega_{ij} \mathbf{x}_i, \quad (3.7)$$

$$\boldsymbol{\Sigma}_j = \frac{1}{n_j} \sum_{i=1}^n \omega_{ij} (\mathbf{x}_i - \boldsymbol{\mu}_j)(\mathbf{x}_i - \boldsymbol{\mu}_j)^T. \quad (3.8)$$

The M-step is completed when all new parameters are computed and the algorithm keeps iterating until convergence. Generally, convergence is defined using the value of the log-likelihood:

$$\log l(\boldsymbol{\Theta}) = \sum_{i=1}^n \left( \log \sum_{j=1}^k \omega_j p(\mathbf{x}_i | \boldsymbol{\theta}_j) \right), \quad (3.9)$$

where  $p(\mathbf{x}_i | \boldsymbol{\theta}_j)$  is the Gaussian density for the  $j$ -th mixture component.

### 3.3.3 Linear Regression

Regression techniques are used to model functional relationships among variables. More specifically, a given dataset can be used to estimate the parameters of a predictor function that models the relationship between a scalar dependent variable and one or more independent variables (i.e. explanatory variables). Thus, one could estimate the typical value of the dependent variable for any given value of the independent variable. A basic technique to compute the regression of a dataset is linear regression (see Figure 3.4).

In linear regression, given a data set  $\{y_i, x_{i1}, \dots, x_{ip}\}$  of  $n$  data points, the dependent variable  $y_i$  is a linear combination of the data points explanatory variables  $x_{ip}$ , where  $i = 1, \dots, n$ . Interestingly, despite the dependent variable being a linear combination of the  $p$ -vector of explanatory variables it does not need to be linear in the independent variables. A variable  $\epsilon_i$  captures the error between  $y_i$  and the explanatory variables. Therefore, we

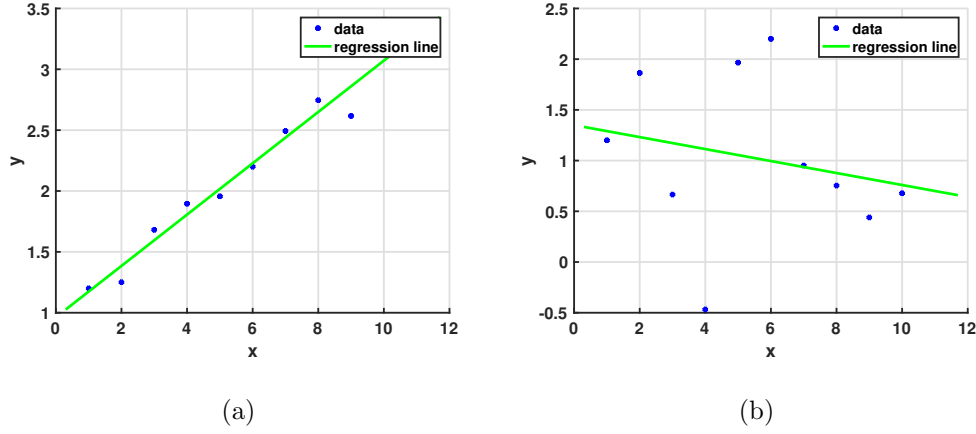


Figure 3.4: Examples of linear regression applied to different datasets. While data can be intuitively fitted into a line in the left figure, the figure on the right shows a sparser dataset and their corresponding regression line.

assume the response following a linear function of the inputs as:

$$y_i = \beta_0 1 + \beta_1 \mathbf{x}_{i1} + \dots + \beta_p \mathbf{x}_{ip} + \epsilon_i, \quad 1 \leq i \leq n. \quad (3.10)$$

or its vector form:

$$\mathbf{y} = \mathbf{X}\hat{\boldsymbol{\beta}} + \boldsymbol{\epsilon}, \quad (3.11)$$

where  $\mathbf{y}$  is the vector of dependent variables,  $\boldsymbol{\epsilon} = \{\epsilon_1, \dots, \epsilon_n\}$  are the regression errors and  $\mathbf{X}\hat{\boldsymbol{\beta}}$  is the matricial form of the product between the explanatory variables  $\mathbf{x}_i$  and the parameters  $\beta_i$ .

We can compute  $\hat{\boldsymbol{\beta}}$  in a straightforward manner using ordinary least-squares estimation. This method minimises the sum of squared residuals leading to a closed-form expression for the estimated value as:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}, \quad (3.12)$$

where  $(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top$  is the pseudo inverse of  $\mathbf{X}$ . Additionally, we can compute the regression error  $\boldsymbol{\epsilon}$  as the difference between the expected value of each

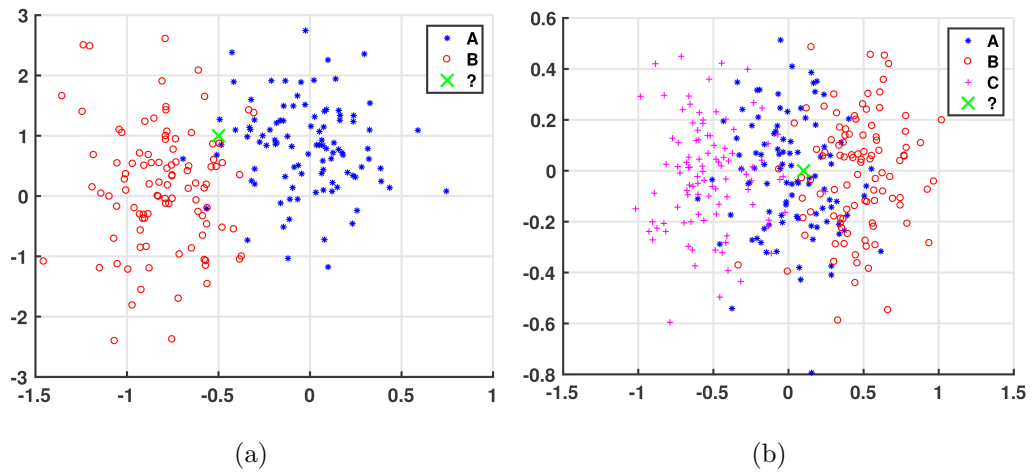


Figure 3.5: The classification problem. On the left, given 2 classes of training data, A and B, decide which of the two classes a new point (i.e. green  $\times$ ) belongs to. On the right, a multiclass ( $k = 3$ ) classification problem.

point in the dataset and their real values.

### 3.3.4 Classification Techniques

While regression statistical techniques model the relationship between variables by estimating a continuous output, classification could be understood as a similar approach for discrete values. Here the goal is to learn a mapping from inputs  $\mathbf{x}$  to outputs  $y$ , where  $y \in \{1, \dots, k\}$ , being  $k$  the number of classes. Figure 3.5 shows two examples of the classification problem statement. Given a labeled dataset, we aim to approximate some unknown function  $y = f(\mathbf{x})$  that makes accurate predictions of unseen data. A large number of classification techniques and variations exists in the machine learning literature. In this section we detail three of the most popular algorithms for data classification  $k$ -Nearest Neighbours, Artificial Neural Network and Support Vector Machine.

### 3.3.4.1 $k$ -Nearest Neighbours

$k$ -Nearest Neighbours ( $k$ NN) is a non-parametric method of data classification. The main idea is to find the  $k$  points in the training set that are nearest to the test input, i.e. count how many members of each class are in this set and return the empirical fraction as the estimate. Although a great variety of distances can be used, the most common distance is the Euclidean distance. More formally, we can define  $k$ NN as a probabilistic model with the form  $p(y|\mathbf{x})$  as follows:

$$p(y = c|\mathbf{x}, \mathbf{D}, k) = \frac{1}{k} \sum_{i \in N_k(\mathbf{x}, \mathbf{D})} \mathbb{1}(y_i = c), \quad (3.13)$$

where  $\mathbf{D}$  is the training set and  $\mathbf{x}$  is the test input,  $N_k(\mathbf{x}, \mathbf{D})$  are the  $k$  nearest points to  $\mathbf{x}$  in  $\mathbf{D}$  and  $\mathbb{1}(e)$  is the indicator function defined as:

$$\mathbb{1}(e) \begin{cases} 1 & \text{if } e \text{ is true} \\ 0 & \text{if } e \text{ is false} \end{cases}$$

Despite its simplicity, in many cases the  $k$ NN classifier outperforms other more sophisticated algorithms. However, the main problem is that it does not work well with high dimensional inputs as, due to the curse of dimensionality, long distances may not be accurate predictors of the input-output function at a given point.

### 3.3.4.2 Artificial Neural Networks

An Artificial Neural Network (ANN) is a learning classifier inspired by biological neural networks, i.e. nervous systems of animals and humans. According to its computational units, an ANN can be group into different generations, for instance, Spiking Neural Networks are considered the third generation

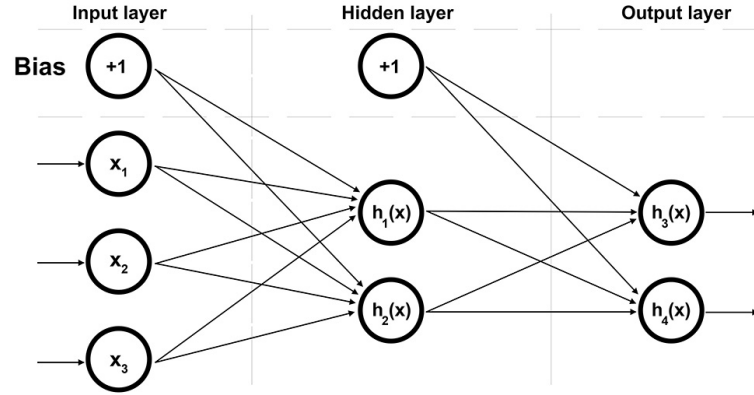


Figure 3.6: Artificial Neural Network. Each circular node represents a neuron and the arrows represent connections from the output of one neuron to the input of another.

of ANNs [156]. For simplicity, we will thereafter use the term ANN to refer to second generation of Artificial Neural Networks. ANNs approximate non-linear functions of the form  $f : \mathbf{x} \rightarrow \mathbf{y}$  as series of logistic regression models stacked on top of each other, with a final layer being another logistic regression. The synapses in the ANN store parameters called “weights” that manipulate the data during the calculations.

Typically, an ANN is defined by three types of parameters: the activation function that converts a neuron’s weighted input to its output activation, the interconnection pattern between the different layers and the learning process for updating the weights of the interconnections. Although a number of functions can be used as activation functions, such as the hyperbolic tangent or rectifier function, one of the most commonly used is the sigmoid function [157], defined as:

$$g(z) = \frac{1}{1 + e^{-z}}. \quad (3.14)$$

An ANN consists of a set of neurons typically arranged in layers, connec-

tions and weights, where each connection transfers the output  $x_i$  of a neuron  $i$  to the input of another neuron. Each connection is assigned a weight  $w_i$ , which is used by the neurons to compute their activation function:

$$a = \sum_i w_i x_i. \quad (3.15)$$

Thus, the output of a neuron is given by its propagation function as:

$$h(x) = g(a). \quad (3.16)$$

This can be conveniently represented as a network structure, with arrows depicting the dependencies between variables. For instance, Figure 3.6 shows an example interconnection pattern of an ANN with three input and two output neurons, using one hidden layer with two neurons. The neurons of the input layer provide an input interface for the network while the neurons in the output layer serve as output interface.

Additionally, the output and all hidden layers have a bias node, whose value is always 1 and connects to all the neurons of the next layer. If a neural network did not have a bias node in a given layer, the output of the propagation function in the neuron of the next layer would not be able to differ from 0 when their input values are 0.

The aim of the learning process is to find a function  $\mathbf{Y} = f(\mathbf{X})$  that matches a given set of example pairs  $(\mathbf{x}, \mathbf{y})$ , where  $\mathbf{x} \in \mathbf{X}$  and  $\mathbf{y} \in \mathbf{Y}$ . In other words, we wish to infer the mapping implied by the data. More formally, given a specific task to solve and a set of parameters  $\Theta = \{\theta_1, \theta_2, \dots, \theta_n\}$  (i.e. weights), we use a set of observations to find the values of  $\Theta^*$  that solve the task in some optimal sense. This entails defining a cost function  $J(\Theta) \in \mathbb{R}$  such that there is no solution with lower cost than the cost of the optimal solution  $\Theta^*$ . The cost function  $J(\Theta)$  is a measure of how far away a particular solution is from the optimal solution.

A commonly used cost is the mean-squared error [155], which tries to minimize the average squared error between the network's output,  $f(\mathbf{x})$ , and the target value  $\mathbf{y}$  over all the sample pairs. When one tries to minimize this cost using gradient descent for the class of neural networks called multilayer perceptrons (MLP), one obtains the common and well-known backpropagation algorithm [158] for training neural networks. This is done by simply taking the derivative of the cost function with respect to the network weights and then changing those parameters in a gradient-related direction.

In a nutshell, the learning algorithm of an ANN searches (i.e. through the solution space) for a function  $f(x)$  that has the smallest possible cost, as  $J(\Theta)$  is related to the mismatch between the mapping and the data.

#### 3.3.4.3 Support Vector Machine

Let's assume some given data points, each one belonging to one of two classes. The goal is to decide which class a new data point will be in as in Figure 3.5(a). In the case of a Support Vector Machine (SVM), a data point is viewed as a  $p$ -dimensional vector, and we want to know whether we can separate such points with a  $(p - 1)$ -dimensional hyperplane. This is called a linear classifier. There are many hyperplanes that might classify the data. One reasonable choice as the best hyperplane is the one that represents the largest separation, or margin, between the two classes, since in general the larger the margin the lower the generalization error of the classifier. So we choose the hyperplane so that the distance from it to the nearest data point on each side is maximized. If such a hyperplane exists, it is known as the maximum-margin hyperplane and the linear classifier it defines is known as a maximum margin classifier; or equivalently, the perceptron of optimal stability.



Whereas the original problem may be stated in a finite dimensional space, it often happens that the sets to discriminate are not linearly separable in that space, i.e. there is no hyperplane that perfectly separates the points. For this reason, we can opt for mapping the original finite-dimensional space into a much higher-dimensional space, potentially making the separation easier in that space.

“The kernel trick” consists of replacing all inner products of the form  $\mathbf{u} \cdot \mathbf{v}$  with a kernel function  $\kappa(\mathbf{u}, \mathbf{v})$ . It is worth noticing that  $\kappa(\mathbf{u}, \mathbf{v})$  needs to be a Mercer kernel for this trick to work [155]. The mappings used by SVM schemes are designed to ensure that inner products may be computed easily in terms of the variables in the original space, by defining them in terms of a kernel function  $\kappa(\mathbf{u}, \mathbf{v})$  selected to suit the problem. Thus, the computational load is kept reasonably low while the kernel function provides an indicator of similarity.

To train an SVM, the first step is to choose the kernel function. Some of the most common kernels are the hyperbolic tangent, polynomial kernels and the Radial Basis Function (i.e. Gaussian kernel) [159]. Given a dataset  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$  and their transformation  $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n\}$  using the kernel trick, we aim to maximize  $L(\boldsymbol{\alpha})$  defined as:

$$L(\boldsymbol{\alpha}) = \sum_{k=1}^n \alpha_k - \frac{1}{2} - \sum_{k,j} \alpha_k \alpha_j z_k z_j \mathbf{y}_j^t \mathbf{y}_k, \quad (3.17)$$

subject to the constraints:

$$\sum_{k=1}^n z_k \alpha_k = 0 \quad \alpha_k \geq 0, k = 1, \dots, n, \quad (3.18)$$

where  $n$  is the number of points,  $\alpha_i$  are undetermined Lagrange multipliers [149], and  $z_k = \pm 1$  is an indicator of the class point to which  $k$  belongs to.

Different to the  $k$ NN and ANN algorithms, which can perform classification of more than two classes (i.e. multiclass classification), SVM classifiers

are binary. However, some techniques can be used to take into account more than two classes when using binary classifiers. This is an active field of research nowadays and a variety of approaches has been proposed. We focus on two of the most common strategies: One-Vs-All and One-Vs-One.

In the One-Vs-All (OvA) strategy, i.e. also known as One-Vs-Rest or One-Against-All, a single binary classifier is trained for each class. Each classifier trains data of one class as positive samples and all other samples as negatives. To make a final decision of the predicted class  $\hat{z}$ , OvA applies all classifiers to an unseen sample  $\mathbf{x}$  and predicts the label  $i$  for which the corresponding classifier  $f_i(\mathbf{x})$  reports the highest confidence score:

$$\hat{z} = \operatorname{argmax}_{i \in \{1, \dots, k\}} f_i(\mathbf{x}), \quad (3.19)$$

where  $k$  is the number of classes in the multiclass problem. This can result in regions of the input space being predicted to be more than one class and, therefore, being ambiguously labeled. Additionally, the performance of each binary classifier in an OvA strategy may be affected by class imbalance if, for instance, there are significantly more negative examples than positive.

In the one-vs.-one (OvO), also known as all pairs,  $\frac{k(k-1)}{2}$  binary classifiers are trained, one for each pair of classes. At prediction time, each classifier evaluates the unseen sample and a majority voting procedure determines the final prediction. Although usually more accurate than OvA, OvO predictions are prone to ambiguities, as some classes may receive the same number of votes. It is worth noting that SVM's do not model uncertainty using probabilities and, therefore, their output scores are not comparable across classes. Ignoring this fact might result in inaccuracies when using the multiclass solutions.

## 3.4 Signal Processing

The efficiency of the classification methods described above can be improved by processing the raw data to discard unnecessary information and obtain descriptors that keeps the relevant one. The approach presented Chapter 4 converts the time-domain tactile microvibration signals into frequency domain using the Fast Fourier Transform algorithm and uses the resultant coefficients as descriptors for classifying the material of an object's surface. In order to visualise the datasets used in the experiments, a number of Fast Fourier Transform examples of tactile vibration signals for different materials are shown in Section 4.2.1. Moreover, signal processing techniques can be used to analyse a signal and, for instance, fit a mathematical model to a data set. A robust estimation algorithm (the Random Sample Consensus algorithm) is used in Chapter 5 to find shape primitives in a depth map obtained from a Microsoft Kinect sensor. Examples of the datasets (i.e. the depth maps) used for segmenting the experimental tabletop manipulation scenario are displayed in Section 5.4.1. This section provides some insight into these algorithms and shows a comparative example between the performance of robust line estimation and linear regression.

### 3.4.1 Fourier Transform

In the Fourier series, periodic functions are stated as a sum of sinusoidal wave functions. The properties of sine and cosine allow one to recover the amplitude of each wave in a Fourier series using an integral. We can do it by computing the Fourier Transform (FT). It is worth noting that in order to compute the FT, it is essential that the integral of the absolute value of the input function  $f(t)$  is a finite value, i.e.  $\int_{-\infty}^{\infty} |f(t)| dt < \infty$ .

The FT is the limit of the Fourier series when the period of the function approaches infinity [160]. Hence, if  $t$  is an independent variable representing time and the transform variable  $\omega$  represents frequency, we define the Fourier Transform (FT)  $F(\omega)$  as:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-2\pi i t \omega} dt, \quad \forall \omega \in \mathbb{R} \quad (3.20)$$

where the Euler's formula  $e^{-2\pi i t \omega} = \cos(2\pi \omega t) + i \sin(2\pi \omega t)$  is used to simplify the Fourier series in terms of basic waves. The representation of sines and cosines as complex exponentials ensures that the Fourier coefficients are complex valued. Therefore, the FT of a function is a complex-valued function of the frequency, whose absolute value represents the “amount” of that frequency existing in the original function, and whose complex argument is the phase offset of the basic sinusoid in that frequency. If the original input function is equally spaced in their input variables (i.e. equal time steps), then the FT is known as a discrete Fourier transform (DFT) and can be computed using the Fast Fourier Transform (FFT) algorithm [161].

The DFT converts the sequence of  $n$  numbers  $x_0, x_1, \dots, x_{n-1}$  into an  $n$ -periodic sequence of complex numbers:

$$y_k = \sum_{j=0}^{n-1} x_j e^{-2\pi i k j / n}, \quad k \in \mathbb{Z} \quad (3.21)$$

where the complex number  $y_k$  encodes amplitude and phase of a sinusoidal component of a sequence  $x_j$ . The sinusoid's frequency is  $k$  cycles per  $n$  samples.

Although discussing all of properties of the DFT goes beyond the scope of this section (see [162] [163] for in depth discussion), the DFT certainly has some relevant properties for the work presented in this thesis. For instance, the DFT is symmetric when the input values are real numbers i.e. the real-input property. Thus, we can split the DFT into  $\frac{N}{2} - 1$  Fourier components

while still carrying all relevant information. Other examples are periodicity and shift theorem properties, which ensure that splitting a time domain periodic signal into several windows with the same size generates similar DFTs.

### 3.4.2 Random Sample Consensus Algorithm

Random sample consensus [164] (RANSAC) is a robust estimation algorithm to fit a mathematical model to a set of observed data. The basic assumption is that the dataset contains a combination of data whose distribution can be explained by a set of model parameters and data that do not fit the model (i.e. outliers). The problem is the following: given a set of data points  $\mathbf{Z}$  and an error threshold  $\epsilon$ , find the subgroup of points  $\mathbf{X}$  that minimizes the sum of squared perpendicular distances while ensuring none of the points in  $\mathbf{X}$  deviates from a given model  $m(\mathbf{X}, \boldsymbol{\theta})$  by more than  $\epsilon$  units, i.e.  $\boldsymbol{\theta}$  are the model parameters.

Instead of obtaining an initial solution and then removing invalid data-points, RANSAC starts by selecting a random subset of  $k$  correspondences. Then, this subset is used to compute an initial estimate for  $\mathbf{X}$ . The support of  $\mathbf{X}$  is measured by the number of inliers computed as  $\|m(\mathbf{X}, \boldsymbol{\theta})\| \leq \epsilon$ . The random selection is repeated  $n$  times and the best estimate of  $\mathbf{X}$  is considered the robust fit.

Figures 3.7(a) and 3.7(b) show different approximations of a line for a given dataset using linear regression (see Section 3.3.3) and RANSAC respectively. While the first is significantly affected by the outliers, in the latter the support for lines is given by the number of points within a threshold distance of the lines. For instance, the line obtained using the points  $\{a, b\}$  has a support of 10 and the line given by  $\{c, d\}$  has a support of 2

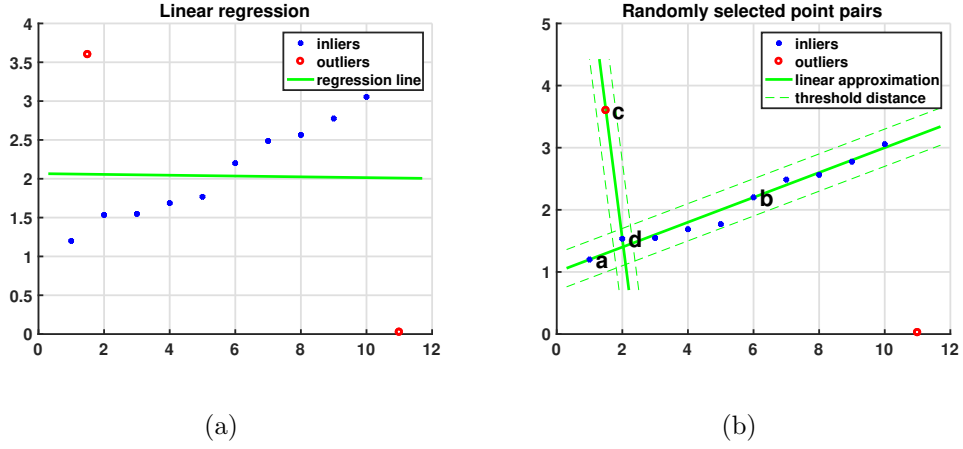


Figure 3.7: Comparison between linear regression and robust line estimation.

and, therefore, RANSAC would choose the line obtained using  $\{a, b\}$  as the robust estimation.

To ensure the algorithm has high probability of finding the true set of inliers, a sufficient number of trials needs to be computed. The likelihood that all  $k$  random samples are inliers (i.e. in one trial) is  $p^k$ , where  $p$  is the probability that any point in  $\mathbf{X}$  is valid. The likelihood that  $n$  trials will not find a subset  $\mathbf{X}$  in which all points are inliers is  $1 - P = (1 - p^k)^n$ , where  $P$  is the total probability of success after  $n$  trials. Hence, the required minimum number of trials is given by:

$$n = \frac{\log(1 - P)}{\log(1 - p^k)}. \quad (3.22)$$

In practice, one tries to keep the number of sample points  $k$  as low as possible since the value of  $n$  grows quickly for greater values of  $k$  [165].

## 3.5 Robotics and Control Systems

This section details the foundations used in Chapters 5 and 6 to explain the proposed approaches for object grasping and robot-human handovers. The techniques used in Chapters 5 and 6 to control the motion of the robot are also presented in this section. Bayesian filtering techniques are explored in this thesis for different applications. For instance, Bayesian filtering is used in Chapter 6 to smooth the trajectory of the angular deviation of the contact forces through time and to estimate their corresponding velocities. Another Bayesian filtering method (i.e. Recursive Bayesian Estimation) is used to develop a novel tactile data classification technique, which is explored in Chapters 4 and 6.

### 3.5.1 Robot Kinematics

A robotic mechanism is a system of rigid bodies with poses determined by their position and orientation in space. Those rigid bodies are connected by joints. Kinematics involve the design, analysis and control of a robotics mechanism and focusses on its motion regardless of the forces/torques that cause it. Thus, robot kinematics describe the rigid body poses (i.e. position and orientation), velocities, accelerations, and higher-order derivatives of the poses.

A fundamental concept to represent robot poses is the coordinate frame, also known as frame. A frame consists of an origin position vector and three mutually orthogonal basis vectors (defining an orientation) that are all fixed within a particular body. The pose of a body is always relative to another body and, thus, it can be understood as the pose of one coordinate frame relative to another. Therefore, in Cartesian coordinates a point can be

represented in space as the origin position  $\mathbf{o}$  of a frame relative to another reference frame

$${}^j\mathbf{o}_i = \begin{pmatrix} {}^j o_i^x \\ {}^j o_i^y \\ {}^j o_i^z \end{pmatrix}, \quad (3.23)$$

where the origin of the coordinate frame  $i$  is relative to the frame  $j$ .

Similarly, the motion of a rigid-body can be expressed as displacements between two frames where one of them stays fixed to the work space. A displacement in which no point of a rigid body remains in its initial position and all its straight lines remain parallel to their initial orientations is called translation. It can be represented by the combination of its position before and after the translation. Therefore, the position can be used to create representations of the translation and vice versa.

Although a point in space has only three degrees of freedom (i.e. position), the minimum number of coordinates required to locate a coordinate frame in Euclidean space and, therefore, a rigid body is six i.e. three for position and three for its orientation. While a point in space can only move along the  $x$ ,  $y$  and  $z$  axes, a coordinate frame can also rotate. A rotation is a displacement in which at least one point in the rigid body remains in its initial position and not all lines in the body remain parallel to their initial position. Similarly to position and translation, orientation can be used to create representations of the rotation and vice versa.

The orientation of a coordinate frame  $i$  relative to coordinate frame  $j$  can be denoted by

$${}^j\mathbf{R}_i = \begin{pmatrix} \hat{\mathbf{x}}_i \cdot \hat{\mathbf{x}}_j & \hat{\mathbf{y}}_i \cdot \hat{\mathbf{x}}_j & \hat{\mathbf{z}}_i \cdot \hat{\mathbf{x}}_j \\ \hat{\mathbf{x}}_i \cdot \hat{\mathbf{y}}_j & \hat{\mathbf{y}}_i \cdot \hat{\mathbf{y}}_j & \hat{\mathbf{z}}_i \cdot \hat{\mathbf{y}}_j \\ \hat{\mathbf{x}}_i \cdot \hat{\mathbf{z}}_j & \hat{\mathbf{y}}_i \cdot \hat{\mathbf{z}}_j & \hat{\mathbf{z}}_i \cdot \hat{\mathbf{z}}_j \end{pmatrix}, \quad (3.24)$$

where  $\{\hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i, \hat{\mathbf{z}}_i\}$  are the basis vectors representing the orientation for frame



$i$  and  $\{\hat{\mathbf{x}}_j, \hat{\mathbf{y}}_j, \hat{\mathbf{z}}_j\}$  denote the basis vectors of frame  $j$ . As those vectors are unit vectors and the dot product of two unit vectors is the cosine of the angle between them, we can find matrices for elementary rotation of frame  $i$  about the axis of frame  $j$  as:

$$\mathbf{R}^x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & C\theta & -S\theta \\ 0 & S\theta & C\theta \end{pmatrix}, \quad \mathbf{R}^y = \begin{pmatrix} C\theta & 0 & S\theta \\ 0 & 1 & 0 \\ -S\theta & 0 & C\theta \end{pmatrix}, \quad \mathbf{R}^z = \begin{pmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (3.25)$$

where  $C$  and  $S$  are the compact notations for  $\cos$  and  $\sin$  respectively, and  $\theta$  is the angle of rotation on the corresponding axis. To perform more complex rotations, we can combine rotation matrices by multiplying them. For instance, the orientation of frame  $i$  relative to frame  $k$  can be computed as:

$${}^k\mathbf{R}_i = {}^k\mathbf{R}_j {}^j\mathbf{R}_i. \quad (3.26)$$

### 3.5.1.1 Homogeneous Transformations

Sometimes, it is necessary to combine a number of successive translations and rotations about a reference frame. As already explained, any point  $\mathbf{p}$  expressed relative to coordinate frame  $i$  can be also expressed relative to the coordinate frame  $j$  if the position and orientations of  $i$  are known relative to  $j$  such as  ${}^j\mathbf{p} = {}^j\mathbf{R}_i {}^i\mathbf{p} + {}^j\mathbf{o}_i$  or its homogeneous representation form:

$$\begin{pmatrix} {}^j\mathbf{p} \\ 1 \end{pmatrix} = {}^j\mathbf{A}_i \begin{pmatrix} {}^i\mathbf{p} \\ 1 \end{pmatrix}, \quad (3.27)$$

where  $\begin{pmatrix} {}^j\mathbf{p} \\ 1 \end{pmatrix}^\top$  and  $\begin{pmatrix} {}^i\mathbf{p} \\ 1 \end{pmatrix}^\top$  are the homogeneous representations of the points  ${}^j\mathbf{p}$  and  ${}^i\mathbf{p}$  in the  $j$  and  $i$  reference frames, and

$${}^j\mathbf{A}_i = \begin{pmatrix} {}^j\mathbf{R}_i & {}^j\mathbf{o}_i \\ \mathbf{0}^T & 1 \end{pmatrix}, \quad (3.28)$$

is the 4 x 4 homogeneous transformation matrix, which transforms points from frame  $i$  to  $j$ , and  $\mathbf{0}$  is a 3-dimensional column vector of zeros. An interesting property of the homogeneous transformation matrix is that its inverse  ${}^j\mathbf{A}_i^{-1} = {}^i\mathbf{A}_j$  transforms points from coordinate frame  $j$  to  $i$ . As with rotations, we can compute compositions of transformations through matrix multiplication. For instance:

$${}^k\mathbf{A}_i = {}^k\mathbf{A}_j {}^j\mathbf{A}_i . \quad (3.29)$$

The homogeneous transformation of a simple rotation about an axis is:

$$\mathbf{R}(x, \theta) = \begin{pmatrix} \mathbf{R}^x & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{pmatrix}, \quad \mathbf{R}(y, \theta) = \begin{pmatrix} \mathbf{R}^y & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{pmatrix}, \quad \mathbf{R}(z, \theta) = \begin{pmatrix} \mathbf{R}^z & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{pmatrix}, \quad (3.30)$$

where  $\mathbf{R}^x$ ,  $\mathbf{R}^y$  and  $\mathbf{R}^z$  are the 3 x 3 rotation matrices described above and  $\theta$  is the angle of rotation. The homogeneous transformation of a simple translation along an axis is:

$$\mathbf{T}(x, d) = \begin{pmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{T}(y, d) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{T}(z, d) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (3.31)$$

Although this representation method is not as computationally efficient as other methods, for instance screw transformations [166] or matrix exponential parametrization [166], the compact notation used in homogeneous transformations simplifies the representation of robot poses.

### 3.5.1.2 Forward Kinematics

A fundamental problem in robot manipulation is to find the pose of the end-effector (i.e. the tool at the end of the arm) relative to the inertial reference system of the manipulator to which it is attached, given the positions of all

joints and the geometric properties of the links. This is known as the forward kinematics (FK) problem for a serial-chain manipulator. In practice, the solution for the FK problem is obtained computing the frame transformation between the end-effector and the base, i.e. the manipulator's joint that fixed to the world space. By convention, the coordinate frame attached to joint  $i$  is defined relative to the coordinate frame  $i-1$ . Therefore, FK can be calculated in a straightforward manner by concatenating homogeneous transformations between frames as explained in Section 3.5.1.1. For instance, in a six-degrees-of-freedom serial-chain manipulator the transformation is

$${}^B\mathbf{A}_E = {}^B\mathbf{A}_1 {}^1\mathbf{A}_2 {}^2\mathbf{A}_3 {}^3\mathbf{A}_4 {}^4\mathbf{A}_5 {}^5\mathbf{A}_E. \quad (3.32)$$

where  ${}^B\mathbf{A}_E$  is the 4x4 homogeneous transformation matrix from the end-effector's coordinate frame ( $E$ ) to the base frame  $B$ .

### 3.5.2 Differential Motions

Differential motions are, by definition, small movements of a mechanism. If a motion is measured in a small period of time, the velocity relationships between different parts of the mechanism can be derived. While solving the FK problem we can find the pose of a manipulator's end-effector given the joint positions (i.e. angles), but it is sometimes useful to compute the relationship between the joint and end-effector velocities.

#### 3.5.2.1 The manipulator Jacobian

The Jacobian matrix converts differential motions or velocities of the joints to differential motions or velocities of the end effector. It represents the differential geometry of the elements of a mechanism. As the joint positions vary in time, the Jacobian is time-related and the magnitude of its elements

also vary in time. One can calculate the Jacobian matrix by taking the derivatives of each position equation with respect to all variables, i.e. joint angles.

Given a set of equations  $y_i$  in terms of a set of variables  $x_j$  as:

$$y_i = f_i(x_1, x_2, \dots, x_j), \quad \forall \ 1 \leq i \leq n, \ 1 \leq j \leq m. \quad (3.33)$$

The differential change in  $y_i$  as the result of a differential change in  $x_j$  is:

$$\begin{cases} \delta y_1 &= \frac{\partial f_1}{\partial x_1} \delta x_1 + \frac{\partial f_1}{\partial x_2} \delta x_2 + \dots + \frac{\partial f_1}{\partial x_j} \delta x_m \\ \delta y_2 &= \frac{\partial f_2}{\partial x_1} \delta x_1 + \frac{\partial f_2}{\partial x_2} \delta x_2 + \dots + \frac{\partial f_2}{\partial x_j} \delta x_m \\ \vdots & \\ \delta y_i &= \frac{\partial f_i}{\partial x_1} \delta x_1 + \frac{\partial f_i}{\partial x_2} \delta x_2 + \dots + \frac{\partial f_i}{\partial x_j} \delta x_m \end{cases} \quad (3.34)$$

and its matricial form is:

$$\begin{pmatrix} \delta y_1 \\ \delta y_2 \\ \vdots \\ \delta y_n \end{pmatrix} = \mathbf{J} \begin{pmatrix} \delta x_1 \\ \delta x_2 \\ \vdots \\ \delta x_m \end{pmatrix}, \quad (3.35)$$

where  $\mathbf{J}$  is the Jacobian matrix defined as:

$$\mathbf{J} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_m} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_m} \\ \vdots & & \ddots & \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_m} \end{pmatrix}, \quad (3.36)$$

Applying this principle to a robot mechanism, we can use the following equation to relate the joint differential motions to the differential motions of its end-effector. For example, the corresponding equation for a six-degrees-of-freedom robotic arm manipulator can be written as:

$$\Delta \mathbf{x} = \mathbf{J}(\boldsymbol{\theta}) \ \Delta \boldsymbol{\theta}, \quad (3.37)$$

where  $\Delta \mathbf{x} = (\delta x \ \delta y \ \delta z \ \delta \alpha \ \delta \beta \ \delta \gamma)^\top$  is a vector representing the differential motions  $(\delta x, \delta y, \delta z)$  and differential rotations  $(\delta \alpha, \delta \beta, \delta \gamma)$  of the end effector,  $\Delta \boldsymbol{\theta} = (\delta \theta_1 \dots \delta \theta_m)^\top$  are the differential motion of the  $m$  joints in the kinematic chain and  $\mathbf{J}(\boldsymbol{\theta})$  is a  $6 \times m$  Jacobian matrix computed for  $\boldsymbol{\theta}$  as in equation 3.36. It is worth mentioning that, if  $\Delta \mathbf{x}$  and  $\Delta \boldsymbol{\theta}$  are divided by  $\Delta t$ , they represent the velocities instead of differential motions and equation 3.36 becomes  $\frac{d\mathbf{x}}{dt} = \mathbf{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$ , where  $\boldsymbol{\theta}$  is a vector of joint angle positions and  $\dot{\boldsymbol{\theta}}$  is a vector of joint angular velocities. Thereafter, we will refer to differential motions while, in fact, can be understood as differential motions or velocities.

Therefore, using the manipulator Jacobian one can obtain the differential motions of the end-effector given the joints differential motions. Additionally, we can calculate how fast each joint needs to move for the end-effector to yield a desired differential motion by calculating the inverse of the Jacobian matrix as  $\Delta \boldsymbol{\theta} = \mathbf{J}^{-1} \Delta \mathbf{x}$ . Moreover, the Jacobian can also be used to represent the relationships between wrenches  $\mathbf{f}$  applied at the end-effector and joint torques  $\boldsymbol{\Gamma}$  such as:

$$\boldsymbol{\Gamma} = \mathbf{J}^\top \mathbf{f}. \quad (3.38)$$

We can use this relation to find what joint torques are required to resist a force applied on the end-effector, i.e. applying a wrench. Similarly, like when using equation 3.37, we can use the inverse of the Jacobian to know what is the resultant end-effector wrench if we apply a set of joint torques.

### 3.5.3 Motion Control

When controlling the motion of a robotic manipulator, it can suffer from steady state errors when trying to get a certain configuration. In other words, when a controller sends a signal to one of the actuators, the joint may overshoot and go beyond the target value even if feedback is used to

stop the motion as soon as the joint reaches the target destination. This happens because of the linkage and the actuators have inertia, and may not stop immediately when the torque signal is turned off. Therefore, we need to control the motion of the robot so that we can achieve the desired configuration with an acceptable margin of error. We can differentiate the control strategies into two groups according to their objectives [166] such as trajectory tracking and regulation strategies. The goal of trajectory tracking is to follow a time-varying joint reference trajectory; which, obviously, must be set inside the manipulator workspace limits. Regulation, also called point-to-point, control aims to place and keep the joint at a desired configuration.

The selection of the control strategy may depend on the type of task to be performed. For instance, if the manipulator is only intended to move from one position to another without requiring significant precision during the motion, one could solve it using regulation strategies. On the other hand, tracking controllers are required when, in addition to the desired target configuration, we need to control the joint velocity and acceleration over time to perform a desired trajectory.

#### 3.5.3.1 PID Controllers

Proportional-integral-derivative (PID) is a type of feedback control mechanism designed to solve the regulation control problem. The PID controller continuously calculates an error  $e(t)$  in the desired configuration and applies a correction based on proportional ( $P$ ), integral ( $I$ ) and derivative ( $D$ ) terms. The error  $e(t)$  is calculated at each time step as the difference between the desired and measured values.

The PID controller is used for regulation as it tries to minimise the error over time by adjusting the control action  $u(t)$ . The general form of a PID

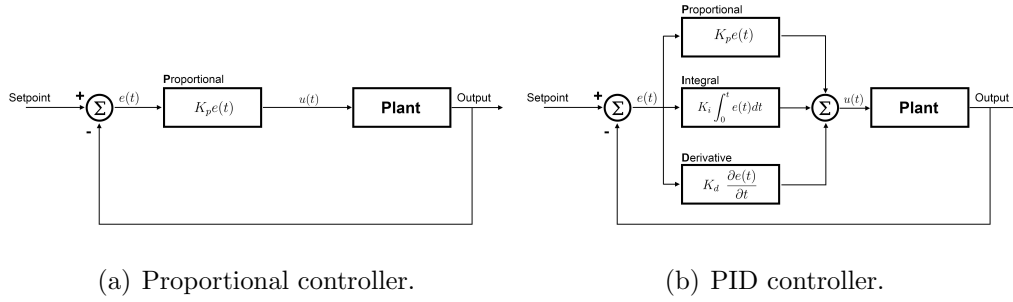


Figure 3.8: PID controller schemes. Figure (a) shows a proportional controller. Figure (b) shows a proportional-integral-derivative (PID) controller.

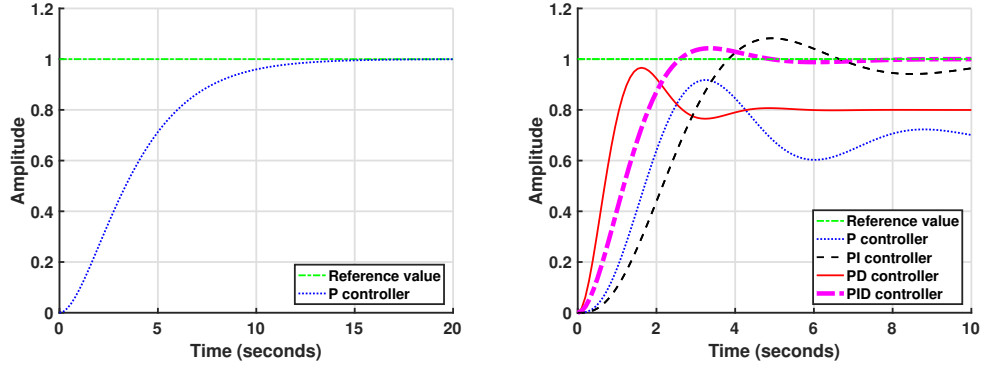
controller can be written as:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt}, \quad (3.39)$$

where  $K_p$ ,  $K_i$  and  $K_d$  are the coefficients of the  $P$ ,  $I$  and  $D$  terms respectively such as:

- $P$  accounts for present values of the error. For example if the error is large the proportional action term will be also large.
- $I$  accounts for past values of the error. For example, if the regulation value is not strong enough, the integral value will accumulate over time, and the controller will apply a stronger action signal.
- $D$  accounts for the expected change of the error based on its current rate of change.

Sometimes, one may want to set some of the parameters  $K_i$  or  $K_d$  to zero. This is acceptable if our control system does not require all of the three terms and, according to what parameters are zeroed, we refer to the controller as a  $P$ ,  $PI$  or  $PD$  controller.



(a) Response of a P controller with  $K_p = 0.25$  and plant  $G(s) = \frac{1}{(s+1)s}$ . (b) Tuned P, PD, PI and PID controller responses for a plant  $G(s) = \frac{1}{(s+1)^3}$ .

Figure 3.9: Response of different PID-type controllers for similar input signals.

Figure 3.8 shows two block diagrams of PID controllers. While Figure 3.8(b) represents an standard PID controller, Figure 3.8(a) represents a proportional (P) controller and, therefore, the integral and derivative are set to zero. Due to their simplicity,  $P$  controllers are very common. In many situations, a correct adjustment for the values of  $K_p$  is enough to make the system behave in a particular way. In others cases, it is not possible to achieve satisfactory results in terms of setting time or precision (i.e. how close to the desired configuration can the system reach). Figure 3.9(a) shows an example of a proportional controller response reaching the desired configuration for the plant (i.e. system to control)  $G(s) = \frac{1}{(s+1)s}$ . The value  $K_p = 0.25$  was selected to have optimal response without oscillations. Using a greater value of  $K_p$  would have faster response but it would oscillate about the reference value. Differently, a proportional controller will never reach the reference value for the plant  $G(s) = \frac{1}{(s+1)^3}$  (see Figure 3.9(b)) and, therefore, more sophisticated controllers are required.



While the setting time can be reduced using a *PD* controller, the precision can be improved by including an integral term (*PI* controller); notwithstanding there is a trade-off between setting time and precision. The *PID* controller provides optimal trade-off between setting time and precision for tuned  $K_p$ ,  $K_i$  and  $K_d$  values (see Figure 3.9). However, using the derivative term amplifies the process noise and, therefore, can cause the output to change abruptly. Discarding the derivative term also makes the system less responsive to real alternation (i.e. not only to noise) and, thus, slower on reaching the set-point, using a *PI* controller instead of a *PID* controller is a valid solution when fast setting time is not required.

### 3.5.4 Bayesian Filtering

In an ideal scenario where all information is available and the robot reacts in real time to the commands, robot control is a relatively simple problem. Unfortunately, having perfect control of mechanical structures is not a realistic assumption. A complete world model is almost never available and, additionally, the robot actuators can not execute perfect motion commands as they are discretised. To overcome this lack of information, sensing elements are used to supply information about the state of the robot and the environment. However, the sensed data may be noisy, not accurate, or insufficient to perform the desired task. Bayesian filtering methods can provide precise estimations of the states of a dynamical system relying on the sensed data. In this section, we will detail two of the most common estimation techniques: the Recursive Bayesian Estimation and the Kalman Filter.

### 3.5.4.1 Recursive Bayesian Estimation

Recursive Bayesian estimation (RBE), also called Bayes filter, is a probabilistic approach for estimating the unknown probability density function of a set of parameters recursively over time. Using the most recently acquired sensor data, this iterative algorithm allows for continuous updating of the probabilities of a set of multiple beliefs.

The aim of the RBE is to compute the marginal posterior distribution of the true state at each time step  $k$  given the history of measurements up to that time step. Assuming the true state  $\mathbf{x}$  is an unobserved Markov process, the next state is based exclusively on its present state (i.e. Markov property) such as:

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{x}_{k-2}, \dots, \mathbf{x}_0), \quad (3.40)$$

where  $\mathbf{x}_k$  is the true state of the Markov process at time step  $k$ .

Therefore, we can cast the problem to a HMM where the sensing data  $\mathbf{z}_k$  are the observed variables and, similarly to the true state, the measurement at time step  $k$  is dependent only upon the previous state:

$$p(\mathbf{z}_k|\mathbf{x}_{k-1}) = p(\mathbf{z}_k|\mathbf{x}_{k-1}, \mathbf{x}_{k-2}, \dots, \mathbf{x}_0). \quad (3.41)$$

For a sequence of  $k$  observed outputs  $\mathbf{z}_{1:k}$ , the RBE computes  $p(\mathbf{x}_k|\mathbf{z}_{1:k})$  in two steps. First, we compute the probability distribution associated with the predicted state  $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$  as:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1}, \quad (3.42)$$

where the probability distribution  $p(\mathbf{x}_k|\mathbf{x}_{k-1})$  is associated with the transition from time step  $k-1$  to  $k$  and  $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$  is the posterior distribution of the true state given the history of measurements at the previous time step.

Secondly, the probability distribution of the update is computed using the Bayes rule:

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k | \mathbf{z}_{1:k-1})}, \quad (3.43)$$

where the denominator is a normalization constant that is obtained using

$$p(\mathbf{z}_k | \mathbf{z}_{1:k-1}) = \int p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) d\mathbf{x}_k. \quad (3.44)$$

#### 3.5.4.2 Kalman Filter

The Kalman filter(KF), presented in [167], is a special case of RBE for the LQG (linear quadratic Gaussian) state space model [168]. This algorithm deals with the inaccuracies and noise of individual sensor measurements by computing the estimate of unknown variables based on series of observation measurements.

Let us consider a discrete-time linear time-invariant dynamical system model:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k \\ \mathbf{z}_k &= \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \end{aligned} \quad (3.45)$$

where  $\mathbf{x}$  is the state vector. The KF model assumes the true state  $\mathbf{x}_k$  at time  $k$  evolved from the state at  $k-1$ , given the state transition model  $\mathbf{F}_k$ , and the control-input vector  $\mathbf{u}_k$  applied through the action model  $\mathbf{B}_k$ . The process noise  $\mathbf{w}_k$  is assumed to follow a multivariate normal distribution with zero mean and covariance  $\mathbf{Q}_k$ , i.e.  $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ . At time  $k$  an observation  $\mathbf{z}_k$  (i.e. measurement) dependent on the true state  $\mathbf{x}_k$  is made according to the observation model  $\mathbf{H}_k$  that maps the state space into the observed space. Similar to the process noise, the observation noise  $\mathbf{v}_k$  is assumed to follow a Normal distribution with zero mean and covariance  $\mathbf{R}_k$ , i.e.  $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$

The general problem is to recursively estimate the state of the system  $\mathbf{x}_k$  given the above model of the system (equation 3.45), the known input  $\mathbf{u}_k$  and some noisy sensor measurements  $\mathbf{z}_k$ . The KF is a recursive estimator that performs two steps, predict and update, during each iteration. In the first step (i.e. predict), a prediction of the state is computed based on the previous state and the input applied. We first compute the *a priori* state estimate  $\hat{\mathbf{x}}_{k|k-1:1}$  at time  $k$  given all observations up to  $k-1$ :

$$\hat{\mathbf{x}}_{k|k-1:1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1:1} + \mathbf{B}_k \mathbf{u}_k, \quad (3.46)$$

where  $\hat{\mathbf{x}}_{k-1|k-1:1}$  is the posterior state estimate obtained during the update state at time  $k-1$ . During the predict state, the KF algorithm also computes the *a priori* estimate covariance:

$$\mathbf{P}_{k|k-1:1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1:1} \mathbf{F}_k^\top + \mathbf{Q}_k, \quad (3.47)$$

where  $\mathbf{P}_{k-1|k-1:1}$  is the posterior estimate covariance computed during the update state at time step  $k-1$ .

In the second step we update the estimate by including the new measurements. Firstly, we obtain the innovation as:

$$\boldsymbol{\nu}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1:1}, \quad (3.48)$$

and the innovation covariance matrix as:

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1:1} \mathbf{H}_k^\top + \mathbf{R}_k. \quad (3.49)$$

Then, we use the Kalman gain

$$\mathbf{K}_k = \mathbf{P}_{k|k-1:1} \mathbf{H}_k^\top \mathbf{S}_k^{-1}, \quad (3.50)$$

to map the innovation into a correction for the predicted state. Thus, we can update the estimate based on the observation by computing the posterior

estimates of the state  $\hat{\mathbf{x}}_{k|1:k}$  and covariance matrix  $\mathbf{P}_{k|1:k}$  as:

$$\begin{aligned}\hat{\mathbf{x}}_{k|1:k} &= \hat{\mathbf{x}}_{k|k-1:1} + \mathbf{K}_k \boldsymbol{\nu}_k \\ \mathbf{P}_{k|1:k} &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1:1}\end{aligned}\tag{3.51}$$

where  $\mathbf{I}$  is the identity matrix.

It is worth mentioning that the KF can be used asynchronously when necessary. If no sensor information is available at a particular iteration, we can compute the prediction step and finish the iteration without performing the update. However, its practical implementation is not always optimal due to the number of assumptions that the KF does, i.e. LQG. The algorithm requires a reasonable initialization of  $\hat{\mathbf{x}}_{0|0}$  and  $\mathbf{P}_{0|0}$  and fairly accurate estimations of the noise covariance matrices  $\mathbf{Q}_k$  and  $\mathbf{R}_k$ , which is still an active field of research nowadays.

## 3.6 Summary

This chapter provides a thorough revision of the methods and robotic platform used in this thesis. The methods used in this research can be grouped into four main areas: Machine Learning, Signal Processing, Robotics, and Control Systems. The Machine learning techniques detailed in this chapter include Principal Component Analysis (PCA), Expectation Maximization (EM) algorithm for Gaussian Mixture Models (GMM), linear regression, and a number of classification methods. Other techniques detailed in this chapter are the Fast Fourier Transform (FFT) algorithm, Random Sample Consensus algorithm (RANSAC), Recursive Bayesian Estimation (RBE) and the Kalman filter. Furthermore, this chapter details the foundations of Robotics and Control Systems used in Chapters 5 and 6 to control the motion of the robot.

The following three chapters present innovative approaches to material identification, object grasping and robot-human object handover. The approach presented in Chapter 4 uses Bayesian Filtering and Machine Learning methods to identify the surface material of an object through continuous tactile sensing. Robotic foundations, Control Systems, and Signal Processing methods are used in Chapter 5 to grasp objects of unknown shape. Finally, the approach to hand over objects between a robot and a human presented in Chapter 6 rely on Robotic foundations, Control Systems and Machine Learning methods to ensure the object transfer is performed in a reliable manner.

## Chapter 4

# Continuous Tactile Sensing for Robotic Material Identification

### 4.1 Introduction

Material identification is a fundamental problem for robot manipulation and grasping as it entails additional information about the properties of an object such as its friction coefficient or weight. Although tactile sensors are sequential, existing works on material identification through tactile sensing present episodic surface recognition approaches, i.e. a whole movement or a sequence of movements has to be completed for the identification to occur. Additionally, few existing approaches in the material identification literature are multimodal, as they usually rely on vibration or force sensing exclusively. However, human identification of surface materials is multimodal. For instance, one may identify a material because it feels smooth and cold, or rigid and warm.

We propose a novel multimodal approach to material identification, which accounts for the sequential nature of robot sensing. This research makes use

of vibration, temperature and deformation of the fingertip skin information from the BioTac tactile sensor. The proposed approach presented in this chapter can be applied online and allows for the integration of other information sources, either from tactile sensing or other modalities. In summary, the proposed approach processes the signals to obtain a set of features that represent the vibration and thermal conductivity responses upon contact with a surface. For each trained material, both input signals (vibration and thermal conductivity) are modelled as mixtures of Gaussian distributions. The Recursive Bayesian Estimation algorithm uses these models for continuous identification of the materials.

The contribution of this chapter is threefold. First, a continuous approach that allows fast and very accurate material identification using vibration signals, and can be extended to other sensing modalities, is presented. Second, it is shown that including temperature information significantly reduces the time needed to identify the material correctly. Finally, the proposed approach is compared with state-of-the-art tactile material identification techniques such as  $k$ NN, ANN and SVM, and with a set of descriptors found in the literature to characterise the vibration signal.

Part of the work presented in this chapter has been published in the following articles:

- Continuous material identification through tactile sensing. In *International Joint Conference on Neural Networks* (IJCNN 2016) [6].
- A multi-modal approach to continuous material identification through tactile sensing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS 2016) [7].

The remainder of the chapter is organised as follows. Section 4.2 presents



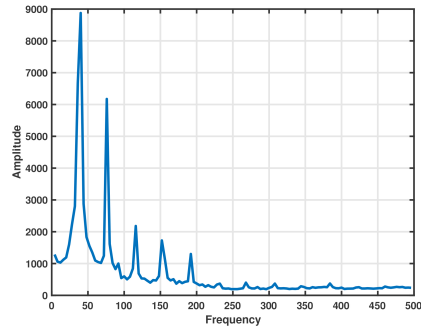
the methodology used for the proposed continuous material identification approach, including processing of the raw data, probabilistic modelling of the readings of each material, and the recursive classification technique applied. Additionally we detail the descriptors found in the literature to compare with the proposed approach and enhance continuous material identification by including thermal information. Section 4.3 presents experimental results and a comparison with state-of-the-art approaches. Finally, Section 4.4 summarises the findings of this chapter.

## 4.2 Continuous Material Identification

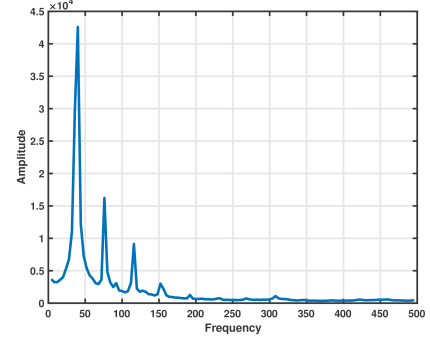
State-of-the-art methods achieve good material identification ratios using one or several batch readings and combinations of exploratory movements. However, episodic identification is not time efficient as some materials might have a characteristic texture or particular thermal properties that allow them to be identified faster than others. This section presents a material identification approach that takes into account the continuous nature of tactile sensing and identifies the material surface as fast as possible. Concisely, the PCA projected Fast Fourier Transform (FFT) of the vibration signal, and the thermal power transferred to the material are proposed as feature descriptors in a Recursive Bayesian Estimation algorithm to classify the materials.

### 4.2.1 Signal Processing and Vibration Features

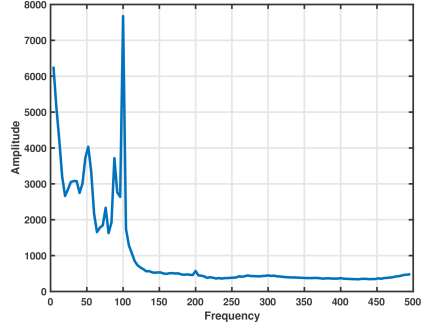
When the BioTAC fingertip interacts with a material, a vibration signal is induced in the liquid gel. Upon contact, this vibration signal is generated as a combination of oscillatory signals with a frequency spectrum dependent on the material. For this reason, one can use the Fourier Transform of the



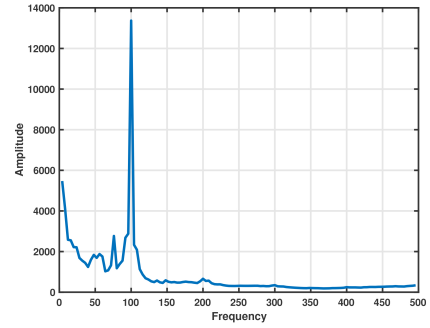
(a) Copper.



(b) Aluminium.



(c) Synthetic leather.



(d) Genuine leather.

Figure 4.1: FFT representation of four materials used in the experiments.

vibration signal to characterise the material texture. In particular, the Fast Fourier Transform (FFT) algorithm is proposed to convert the vibration signal  $\nu(t)$  into the frequency domain  $\rho(\omega)$ .

As this work aims to implement continuous material identification (instead of episodic), the FFT is computed for small non-overlapping windows of duration  $\Delta t$ . The existing literature [89] indicates that good discrimination results are obtained by using the frequency response in the range between 2 Hz and 500 Hz, therefore, for consistency this range was used to identify the materials.

For a  $\Delta t$  window, the FFT is a high dimensional vector of complex num-

bers  $\rho(\omega) \in \mathbb{C}^{d_{\Delta t}}$ , where the dimension  $d_{\Delta t}$  depends on the selected time interval  $\Delta t$  and the sampling period. Principal Component Analysis (PCA) is used over  $\rho(\omega)$  to obtain a lower dimensional feature vector  $\bar{\rho} \in \mathbb{R}^d$ , where  $\bar{\rho}$  is the result of projecting  $\rho(\omega)$  and  $d \ll d_{\Delta t}$ .

The complex nature of  $\rho(\omega)$  requires special care when processing the FFT through PCA. Despite the fact that PCA can be used with complex vectors, it was found that the modulus of the FFT carries the discriminative information for material identification. However, the centering process was performed over the complex vectors by subtracting the complex mean before computing the projection matrix of the FFT modulus. Thus, applying PCA to the centred vectors  $\rho - \mu_\rho$ , with mean complex values of the FFTs  $\mu_\rho \in \mathbb{C}^{d_{\Delta t}}$ , helps to cancel the random noise in the FFT, as white noise's phase is random while its amplitude is fixed.

Figure 4.1 shows the average values of the FFT frequencies modules for small periods ( $\Delta t = 0.25$  second) of the vibration signal upon contact with different materials. Clear differences can be observed in the FFT of the vibration signal when comparing materials, although some plots correspond to surfaces that are difficult to differentiate by human tactile sensing alone, such as copper and aluminium or synthetic and genuine leather.

Many existing works in the literature extract features from the vibration signal instead of using the Fourier coefficients. These features reduce the dimension for the vector used to characterise the material by encoding general properties of the vibration reading such as its total energy or statistical properties (e.g. average energy or temporal variance). However, this process discards relevant information for identifying the surface material. Moreover, if one is extracting frequency domain features, the process is not very efficient computationally as it is necessary to perform the FFT anyway. In Section

4.3, the filtered FFT projected through PCA  $\bar{\rho}$  will be compared with a vector  $\mathbf{x}$  of six features ( $\mathbf{x} \in \Re^6$ ) found in the related literature as discussed below, that combine independent information in the frequency and temporal domains. Therefore, the feature vector  $\mathbf{x} = \{x_1, x_2, \dots, x_6\}$  is defined using the following descriptors:

- The filtered average energy [79] is computed from the band pass filtered FFT as:

$$x_1 = \frac{1}{n} \sum_{i=1}^n |\rho_i|^2, \quad (4.1)$$

where  $\rho_i = \rho(\omega_i)$  is the Discrete Fourier Transform (DFT) of the vibration,  $\omega_1 = \omega_m$  and  $\omega_n = \omega_M$  are the limit frequencies of the band pass filter, and  $n$  is the number of samples within the frequency range  $[\omega_m, \omega_M]$ .

- The high frequency to low frequency energy ratio [96] is another sequential feature computed as:

$$x_2 = \frac{\sum_{i=1}^k |\rho_i|^2}{\sum_{i=k+1}^n |\rho_i|^2}, \quad (4.2)$$

where  $\rho_i = \rho(\omega_i)$ ,  $\omega_k \leq \omega_{th}$ ,  $\omega_{k+1} > \omega_{th}$  and  $\omega_{th}$  is a threshold that defines what low and high frequencies are (as in [96] we set  $\omega_{th} = 100Hz$ ).

- The spectral centroid [79] is computed from the FFT as the weighted frequency as follows:

$$x_3 = \frac{\sum_{i=1}^n |\rho_i|^2 \omega_i}{\sum_{i=1}^n |\rho_i|^2}, \quad (4.3)$$

where  $\rho_i$  is defined as above for the corresponding frequency  $\omega_i$ .

- Temporal variance of the signal was used as a feature in [96] and [95], and can be computed from the raw signal as:

$$x_4 = \frac{1}{n} \sum_{i=1}^n (\nu_i - \mu_\nu)^2, \quad (4.4)$$

where  $\nu_i = \nu(t_i)$  is the temporal signal of the vibration,  $\mu_\nu$  is its average value, and  $n$  is the number of vibration samples considered, which are also used to compute the temporal Skewness and Kurtosis of the vibration signal.

- Temporal Skewness of the signal [95] is another temporal statistic of the distribution of vibration values, computed as:

$$x_5 = \frac{\frac{1}{n} \sum_{i=1}^n (\nu_i - \mu_\nu)^3}{\left[ \frac{1}{n-1} \sum_{i=1}^n (\nu_i - \mu_\nu)^2 \right]^{\frac{3}{2}}}. \quad (4.5)$$

- Temporal Kurtosis of the signal, also used in [95], is a higher central moment of the vibration values computed as:

$$x_6 = \frac{\frac{1}{n} \sum_{i=1}^n (\nu_i - \mu_\nu)^4}{\left[ \frac{1}{n} \sum_{i=1}^n (\nu_i - \mu_\nu)^2 \right]^2} - 3. \quad (4.6)$$

The feature vector representing the vibration signal will be hereafter referred to as  $\bar{\rho}$ , the PCA projected FFT, although when performing the comparison with the features defined above,  $\bar{\rho}$  is substituted by  $\mathbf{x}$  as an input for the classifiers.

### 4.2.2 Statistical Modelling of Vibration

We will denote by  $\tilde{m}$  the discrete random variable encoding the  $n$  material textures to be identified, i.e.  $\{m^1, m^2, \dots, m^n\}$ , and  $P \in \mathfrak{R}^d$  the random

vector of features, FFT projected through PCA. Training sets were collected to estimate the conditional probability density function of the vibration signal for each material,  $p(\mathbf{P} = \bar{\boldsymbol{\rho}}|\tilde{m} = m^j)$ . For simplicity, we will use the compact notation  $p(\bar{\boldsymbol{\rho}}|m^j) = p(\mathbf{P} = \bar{\boldsymbol{\rho}}|\tilde{m} = m^j)$ . The conditional probability of each material were modelled as a mixture of Gaussian distributions, i.e. Gaussian Mixture Model (GMM), of the form:

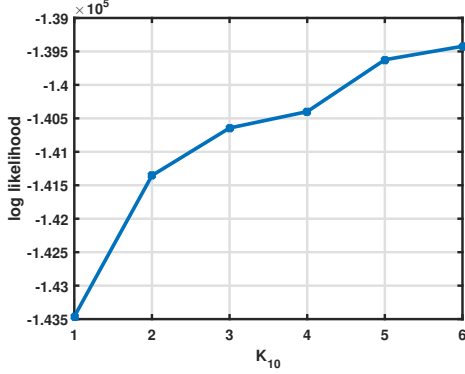
$$p(\bar{\boldsymbol{\rho}}|m^j) = \sum_{i=1}^{K_j} \alpha_i^j \mathcal{N}(\boldsymbol{\mu}_i^j, \boldsymbol{\Sigma}_i^j) \quad (4.7)$$

where  $K_j$  is the number of Gaussians in the mixture for material  $m^j$ ,  $\alpha_i^j$  is the prior probability (i.e. weight) of the  $i^{th}$  Gaussian, and  $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  represents a normal distribution with mean  $\boldsymbol{\mu}$  and covariance  $\boldsymbol{\Sigma}$ . Therefore, for each material  $j$ , we obtained  $K_j$  normals with different means ( $\boldsymbol{\mu}_i^j$ ) and covariances ( $\boldsymbol{\Sigma}_i^j$ ), and their corresponding weights  $\alpha_i^j$ .

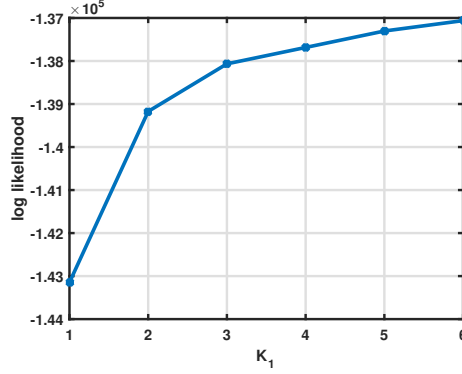
A GMM was estimated for each material and their parameters were obtained using the Expectation-Maximisation (EM) algorithm (see Chapter 3). The number of Gaussian functions for each material was empirically chosen observing the increase of the maximum likelihood of the model. Figure 4.2 shows the log-likelihood of the model as a function of  $K_j$  for two materials, copper and a synthetic fabric. The slope of the curve is large for small values of  $K_j$  while it decreases and become more stable as  $K_j$  increases. Therefore, one can choose the value so that a greater number of clusters does not significantly increase the likelihood of the model.

### 4.2.3 Material Recursive Bayesian Estimation

In contrast with the approaches used in the literature where material identification is treated as a batch classification problem, a Recursive Bayesian



(a) Copper



(b) Synthetic fabric

Figure 4.2: Log-likelihood of a GMM as a function of the number of clusters  $k_j$  for three different materials.

Estimation algorithm was implemented to estimate over time an identification probability of each of the materials to be identified. Although recursive estimation algorithms are widely used in robotics they have not been used for material identification. At this stage we have the conditional probability density function  $p(\bar{\rho}|m^j)$  for each of the tested materials, which would already allow identification of the materials using a maximum likelihood (ML) classifier. Alternatively, one could classify according to the maximum *a posteriori* (MAP) probability of a material by using the Bayes rule for a set of prior probabilities  $p(m^j)$  to estimate the posterior material probabilities  $p(m^j|\bar{\rho})$ . The classification results of both ML and MAP approaches are identical under the assumption of uninformative priors  $p(m^j) = \frac{1}{n}$  for all  $j$ .

Nevertheless, this work aims at identifying materials continuously without the need to collect a long sequence of readings. The vibration signal  $\bar{\rho}$  is used to iteratively obtain new probability posteriors for each material. Before the robot hand starts the identification of a material, the initial prior probabilities

are distributed evenly among all materials. Then, the material posterior at each time step  $p(m_k^j|\bar{\rho}_k)$  will be used as prior probability  $p(m_{k-1}^j|\bar{\rho}_{k-1})$  for the next iteration. In summary, at  $\Delta t$  intervals the feature vector  $\bar{\rho}_k$  was obtained sequentially from the BioTAC data stream by computing the PCA projection of the filtered FFT, and update the material probabilities using:

$$p(m_k^j|\bar{\rho}_k) = \frac{p(\bar{\rho}_k|m_k^j)p(m_{k-1}^j|\bar{\rho}_{k-1})}{p(\bar{\rho}_k)}, \quad (4.8)$$

where the normalisation constant  $p(\bar{\rho}_k)$  can be obtained as:

$$p(\bar{\rho}_k) = \sum_i^n p(\bar{\rho}_k|m_k^i)p(m_{k-1}^i|\bar{\rho}_{k-1}). \quad (4.9)$$

In each iteration the algorithm updates the posterior probability  $p(m_k^j|\bar{\rho}_k)$  for all materials,  $j = 1, \dots, n$ , and the material with the highest posterior can be considered to be the one the robot is touching. Alternatively, one could define a confidence level to decide on a material only if the posterior probability is above some threshold. Instead of predicting the perceived texture in one episode, the Recursive Bayesian Estimation algorithm incrementally updates the probability estimate of the material.

#### 4.2.4 Multimodal Continuous Material Classification Using Thermal Features

A multimodal (i.e. vibration and thermal properties) tactile approach is proposed to enhance continuous material identification. Most of the state of the art methods for material identification, as well as the approach presented in Section 4.2.3, rely on the vibrations generated during the contact between the fingertip and the surface. However, the proposed approach (Section 4.2.3) allows a straightforward inclusion of other modalities. Combining



vibration with thermal information might improve the identification accuracy, for instance, for materials with similar textures but different thermal properties. As detailed in Chapter 2, a heating device maintains the core of the BioTAC higher than room temperature. Thus, the BioTAC can measure the temperature at the core, which typically decreases during contact with a material because of the heat flux leaving the finger through the contact surface, which is at room temperature. This can be used to discern a surface as each material has a different thermal conductivity. In addition to the thermal conductivity of the material, the thermal energy lost also depends on the contact area and the temperature difference between the finger and the external material. In particular, the thermal power lost (thermal energy per unit of time  $\frac{\partial E}{\partial t}$ ) is defined as the integral of the heat flux over the contact surface:

$$\frac{\partial E}{\partial t} = \oint_S \vec{\phi} \cdot d\vec{S} \quad (4.10)$$

where  $\vec{\phi}$  is the heat flux, and the integral is computed on the contact area  $S$  between the two materials. We can assume the flux  $\vec{\phi}$  always leaves the finger, and its modulus – measured by the BioTAC sensor – increases with the temperature difference and the thermal conductivity of the material (although the thermal conductivity of the finger might play a role). The thermal flux is assumed to be directed towards the surface normal as the thermal conductivity of the air compared with the material is small, and the friction is assumed to be too weak to generate thermal energy. Since the contact area is typically small, the power loss can be approximated as the product of the average flux modulus  $\bar{\phi}$  by the contact area. Considering all objects to be identified are at room temperature, the temperature difference depends only on the temperature of the core, and therefore the thermal conductivity of the material can be approximated as  $\frac{\bar{\phi}A}{T}$ , where  $A$  is the contact area, and  $T$

is the finger core temperature.

The contact area is estimated from the readings of the 19 electrodes placed in the core under the BioTAC's skin. These electrodes measure the impedance, which is related to the thickness of the fluid between the core and the finger skin at their corresponding locations. When the fingertip is in contact with a surface, it causes a deformation in the rubber skin and, thus, the value of the readings in the surrounding electrodes decreases. Electrodes with negative value with respect to their resting level indicate contact. The contact area for each electrode  $i$  is approximated as a circle of radius  $r_i$  equal to half the distance between the electrode and its nearest neighbour. Therefore, the contact area of the fingertip and the material can be approximated as a weighted sum of small circular areas:

$$A = \sum_i \lambda_i \pi r_i^2, \quad (4.11)$$

where  $\lambda_i \in [0, 1]$  is a scale factor that depends on the value returned by electrode  $i$ . At the resting level (or above)  $\lambda_i$  is zero, and it increases to 1 for decreasing impedances down to a fixed minimum threshold  $e_m$  (in this case  $e_m = -400$ ), and is 1 for values below that threshold. To calculate the scale factor  $\lambda_i$  we define a piece-wise linear function of the change of the impedance value  $e_i$  of each electrode relative to the resting level  $\bar{e}_i$  as:

$$\lambda_i = \begin{cases} 1 & \text{if } e_i \leq e_m \\ 1 - \frac{e_i - e_m}{\bar{e}_i - e_m} & \text{if } e_m < e_i < \bar{e}_i \\ 0 & \text{if } e_i \geq \bar{e}_i \end{cases} \quad (4.12)$$

Additionally, another two thermal features were experimentally found to improve the material identification accuracy. A linear regression of the thermal flux as a function of time was performed for each sequence of  $\Delta t$

readings, and used the slope and the regression error as additional features. Therefore, in addition to the vibration feature vector  $\bar{\rho}_k$ , a three dimensional heat based feature vector  $\theta_k$  was obtained for each  $\Delta t$  time window.

Similarly to the modelling of the vibration signal, the conditional probabilities were computed for training each material  $p(\theta|m^j)$  using the technique presented in Section 4.2.2. As for the vibration only approach (see Section 4.2.3), the material probabilities can be updated at every time step  $k$  using:

$$p(m_k^j|\bar{\rho}_k, \theta_k) = \frac{p(\bar{\rho}_k, \theta_k|m_k^j)p(m_{k-1}^j|\bar{\rho}_{k-1}, \theta_{k-1})}{p(\bar{\rho}_k, \theta_k)}, \quad (4.13)$$

where the normalisation constant  $p(\bar{\rho}_k, \theta_k)$  is:

$$p(\bar{\rho}_k, \theta_k) = \sum_{i=1}^N p(\bar{\rho}_k, \theta_k|m_k^i)p(m_{k-1}^i|\bar{\rho}_{k-1}, \theta_{k-1}), \quad (4.14)$$

and, the vibration and thermal features are assumed to be conditionally independent with each individual likelihood function given by the corresponding GMM model for material  $m^j$ , i.e.  $p(\bar{\rho}_k, \theta_k|m_k^j) = p(\bar{\rho}_k|m_k^j)p(\theta_k|m_k^j)$ . At each iteration, the algorithm updates the estimated probability  $p(m_k^j|\bar{\rho}_k, \theta_k)$  for all materials,  $j = 1, \dots, n$ , and the perceived material is considered to be the one with the highest posterior probability.

#### 4.2.4.1 A note on sensor fusion

In Section 4.3 the proposed approach is compared with other state of the art techniques for material identification such as  $k$ NN, ANN and SVM. The multimodal recursive material identification framework combines vibration and thermal information in a straightforward manner by assuming both modalities are conditionally independent. Therefore, one can compute the likelihood function of the combined data as the product of the individual likelihood functions. However, this approach is not possible in the case of  $k$ NN, ANN

and SVM. There are several ways in which one can combine the data for those classifiers, for instance, one could classify each modality independently and then use another classification level to make a final decision on the material. We opted instead for performing the sensor fusion at the feature level by defining a single feature vector  $\hat{\mathbf{x}}$  joining together the vibration and thermal features  $\hat{\mathbf{x}} = [\bar{\boldsymbol{\rho}} \ \boldsymbol{\theta}]$  (alternatively  $\hat{\mathbf{x}} = [\mathbf{x} \ \boldsymbol{\theta}]$ ). This is simpler than a combination of classifiers as there is no need to evaluate several classifiers to make the final decision. However, the components of the vector provided as an input to the classifier are obtained from diverse sources and, therefore, they have different numerical scales. To avoid negative effects of the scaling on the classifier's performance, data whitening was implemented as a pre-processing step for the learning techniques used. Hence, the training and testing datasets were transformed to have zero mean and covariance equal to the identity matrix.

### 4.3 Experimental Results

To evaluate the continuous material identification approach explained above, a series of experiments in which 34 materials were used for classification was performed. In this section, the results for continuous material identification using the two different vibration feature sets,  $\bar{\boldsymbol{\rho}}$  and  $\mathbf{x}$ , detailed in Section 4.2.1 are shown. Moreover, results of the multimodal approach presented in Section 4.2.4 are shown in this section. As continuous material identification aims to identify the material as fast as possible, the average identification accuracy and the time needed for successful recognition were used as performance measures. Figure 4.3 shows the classification pipeline followed in our experiments, including a comparison with other state-of-the-art techniques.

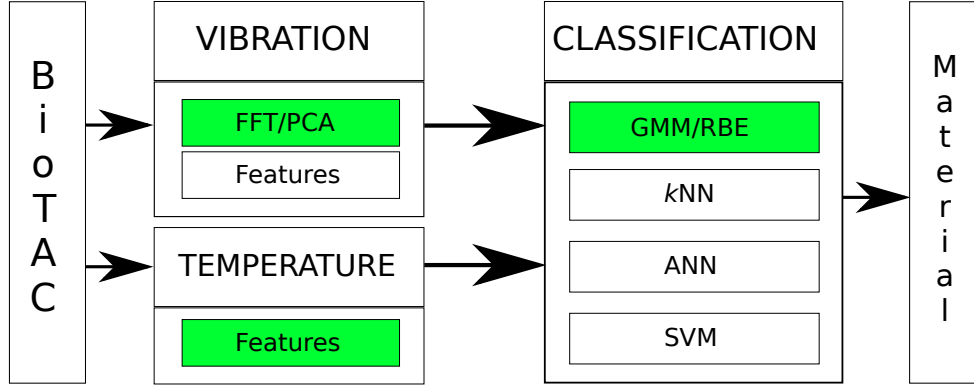


Figure 4.3: Material identification pipeline scheme.

The colour boxes represent the proposed approach while the plain boxes show features and classifiers commonly found in the literature. In addition to thoroughly evaluating the presented continuous material identification approach (i.e. GMM/RBE) and its multimodal extension, the identification accuracy was compared with other classification approaches such as  $k$ -Nearest Neighbour ( $kNN$ ), Artificial Neural Network (ANN) and Support Vector Machine (SVM).

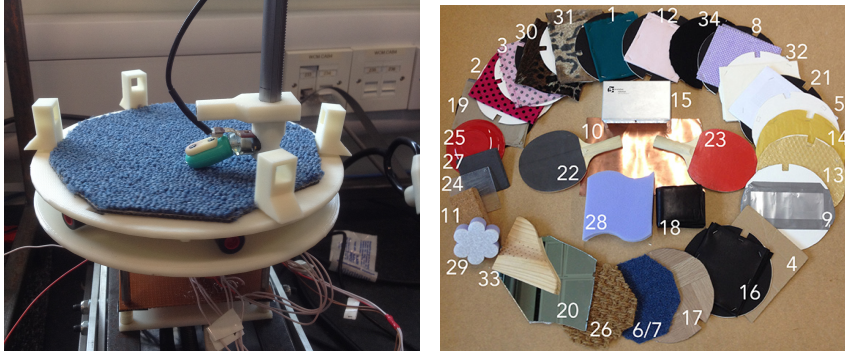
The main advantage of our algorithm is that it minimises the time needed to identify a material, as the classification is performed over several small data windows. Conversely, all batch approaches existing in the material identification literature decide on a single, typically larger window. Therefore, the time interval required for successful material identification for the proposed approach and the assessed machine learning techniques cannot be compared. However, in the first set of experiments, the average number of iterations that the recursive Bayesian estimation takes to successfully classify a material was measured, and this information was used to perform a fair comparison between the recursive identification and the  $kNN$ , ANN, and SVMs algorithms.

### 4.3.1 Experimental Set-Up

The experimental setup is shown in Figure 4.4(a) and consists of a turntable moved by a step motor through a set of reduction gears and a BioTAC sensor attached to a worm drive bar. The motor was controlled by an Arduino board, which sets the turning speed of the motor to rotate the turntable at 4 secs per lap. A second motor is also controlled by the Arduino board to move up and down the worm drive bar. This additional motor allows the BioTAC to touch the material on the turn table when the bar is driven down. Friction between the surface and the fingertip is proportional to the contact force. A high contact force (i.e. large friction) stops the turntable, as the motor has not enough power. Therefore, the contact force has to be kept low which also implies the thermal energy created by the friction (i.e.  $E = F_f d$ , where  $F_f$  is the friction force and  $d$  the displacement of the turntable) is low.

It is worth noting that the temperature of the finger takes time to reach a steady state. The finger was allowed to warm up until the temperature was stable before gathering any of the data-sets, either for training or testing. Then the speed of the turntable was set and then the fingertip was moved down until it touched the material surface. The readings were collected continuously storing all the information provided by the BioTAC sensor running at 4.4  $kHz$ , the maximum communication rate. Thus, the absolute temperature, impedance and thermal flux are collected at a sampling frequency of 100  $Hz$ , while the vibration signal has a maximum frequency of 2.2  $kHz$ .

In the experiments a set of materials were purposely selected to include different groups and several materials in each group. Figure 4.4 shows and labels a total of 34 materials used in the identification experiments, which are listed in Table 4.1. The selection contains materials with very different



(a) Experimental setup.

(b) Materials used in the experiments.

Figure 4.4: Experimental setup for material identification.

textures such as the glass from a mirror and wood, but also a number of fabrics and materials which are dissimilar. As some of the fabrics have different textures on both sides of the cloth, for instance materials 2 and 3, and materials 30 and 31, these fabrics were used as separate materials and thus increased the number of materials used on the experiments. Genuine and synthetic leather, materials 16 and 18 respectively, were also included and materials 13 and 14 were obtained using both sides of a padded envelope, material 13 being the side of the bubbles, i.e. plastic, and material 14 the paper side. Other pairs correspond to both sides of the same object such as materials 6-7, 22-23 and 28-29, although their surfaces were clearly different.

### 4.3.2 Microvibrations-based Material Identification

In the first set of experiments the proposed approach for tactile material identification was evaluated using vibration information only. As it is explained in Section 4.2, the FFT of the vibration signal was computed, band pass filtered and projected into a lower dimensional space. In all experiments, the

Table 4.1: List of materials used in all our classification experiments.

Idx	Material	Idx	Material
1	Synthetic Green fabric	18	Genuine leather
2	Synthetic Pink fabric 1	19	Linen
3	Synthetic Pink fabric 2	20	Mirror
4	Cardboard box	21	Normal paper
5	Cardboard disk	22	Ping pong paddle 1
6	Carpet	23	Ping pong paddle 2
7	Rubber	24	Plastic
8	Baize	25	Plastic dish
9	Can of drink	26	Rough fabric
10	Copper	27	Slate stone
11	Cork	28	Sponge 1
12	100% Cotton	29	Sponge 2
13	Padded envelope 1	30	Leopard fabric 1
14	Padded envelope 2	31	Leopard fabric 2
15	Aluminium	32	Watercolour paper
16	Synthetic leather	33	Wood
17	Floor tiles	34	Peach skin fabric



frame rate for continuous material classification was set to 4  $Hz$  by splitting the time domain signal into non-overlapping windows of  $\Delta t = 0.25$  secs. The corresponding window is a sequence of 550 pressure samples that is transformed into the frequency domain by computing the FFT. The resultant FFT is band pass filtered by selecting only the 2 – 500  $Hz$  frequency range and, therefore, reduced to 124 FFT spectral components. It is worth noting that, given the short time interval selected, the frequency interval between spectral components is approximately 4  $Hz$ , that is the actual minimum frequency in the band pass filter. The 124 spectral components were projected to a 16 dimensional space, computing the complex average with the training dataset for all 34 materials, while projecting their amplitude vectors. The number of reduced dimensions was chosen to keep 97% of the total variance of the original vectors and, therefore, the input to the recursive classification will be  $\bar{\rho} \in \Re^{16}$ .

The resultant feature vectors, the filtered FFT components projected through PCA, were used for training the models of the likelihood functions for each material as GMMs using the EM algorithm. To evaluate the proposed approach 10-fold cross-validation was performed on the 10 minutes of data sequences recorded for each material. The datasets were split into 10 groups of 60 secs sequential readings using alternatively 9 of the groups for training and one for testing. It was found that the RBE approach always identifies the correct material if enough evidence is presented and, once it is detected, the rest of the sequence maintains a high probability estimate of the correct material. The evaluation procedure used 60 secs of test time series of every material starting from the first sample and stopping once the recognition was successful (i.e. correct material detected), storing the number of required iterations. The first sample was then discarded (creating a

shorter test sequence) and the process was repeated until the sequence was too short for the algorithm to identify the material (typically sequences of 0.25 secs). This process was repeated for each testing sequence in the 10-fold cross-validation to compute the average number of iterations needed for successful identification and its standard deviation. Because the RBE algorithm always identifies correctly the material, the ratio of successful material classification cannot be considered as a performance measure. Therefore, this experiment evaluates how long the approach takes to detect a material based on the maximum posterior probability  $p(m_k^j|\bar{\rho}_k)$ , alternatively, using the feature vector  $p(m_k^j|\mathbf{x}_k)$  (see Section 4.2.1). The average number of iterations and its standard deviation was computed, and translated into seconds multiplying by  $\Delta t$  (0.25 secs). These timings will be used in Section 4.3.2.1 to compare the proposed RBE approach to other state-of-the-art classifiers.

Table 4.2 shows the average time (in secs) required to correctly identify each material over the 10 trials of the 10-fold cross-validation. Most of the time, the algorithm was found to identify the materials within less than 0.5 secs when using the FFT-PCA components to model the material vibrations. The results illustrate an average recognition time of 0.36 secs, which is just greater than one iteration, with a standard deviation of 0.2 secs (i.e. just less than one iteration). Although, obviously the recognition time is a discrete random variable with positive skewness, attending to the probabilities of a normal distribution, one can conclude that, with nearly 99% probability, the time required for successful identification is less than one second, i.e. average time plus three standard deviations.

Similar performance measures are applied to the results shown in Table 4.2 to evaluate the presented approach when using the six dimensional feature vector  $\mathbf{x} = (x_1, x_2, \dots, x_6)$  instead of  $\bar{\rho}$  (see Section 4.2.1). The

Table 4.2: Average time (in seconds) needed for material classification using the FFT of the vibration signals and 10-fold cross-validation. First number is the average time and second is the standard deviation.

Material	1	2	3	4	5
<b>FFT+PCA</b>	0.39+0.21	0.35+0.19	0.35+0.19	0.43+0.28	0.39+0.25
<b>Features</b>	0.74+0.62	0.64+0.56	0.56+0.42	0.68+0.41	0.72+0.64
Material	6	7	8	9	10
<b>FFT+PCA</b>	0.3+0.12	0.33+0.14	0.44+0.26	0.34+0.16	0.3+0.2
<b>Features</b>	0.34+0.20	0.64+0.51	0.90+0.46	0.79+0.59	1.58+1.29
Material	11	12	13	14	15
<b>FFT+PCA</b>	0.3+0.11	0.35+0.18	0.33+0.16	0.34+0.17	0.3+0.16
<b>Features</b>	0.48+0.34	0.54+0.53	0.76+0.60	0.49+0.28	0.56+0.48
Material	16	17	18	19	20
<b>FFT+PCA</b>	0.44+0.25	0.46+0.34	0.32+0.14	0.49+0.35	0.29+0.11
<b>Features</b>	0.74+0.59	1.24+1.29	0.41+0.28	0.77+0.63	0.41+0.28
Material	21	22	23	24	25
<b>FFT+PCA</b>	0.39+0.25	0.38+0.22	0.36+0.19	0.38+0.22	0.38+0.21
<b>Features</b>	0.42+0.28	0.55+0.41	1.06+0.70	1.01+0.83	1.15+0.79
Material	26	27	28	29	30
<b>FFT+PCA</b>	0.27+0.07	0.42+0.25	0.29+0.13	0.32+0.14	0.27+0.08
<b>Features</b>	0.33+0.15	0.64+0.46	0.47+0.39	0.60+0.44	0.30+0.16
Material	31	32	33	34	<b>Avg.</b>
<b>FFT+PCA</b>	0.27+0.08	0.46+0.3	0.52+0.34	0.4+0.26	<b>0.36+0.2</b>
<b>Features</b>	0.35+0.25	0.29+0.12	0.98+0.70	0.65+0.63	<b>0.67+0.51</b>

10-fold cross-validation results show that the time needed to recognise the correct material is on average 0.67 secs (more than two iterations), while the standard deviation is also slightly greater than two iterations (0.51 secs). Following the same reasoning, high probability of successful material identification is above 2 secs, i.e.  $\mu + 3\sigma$ , when modelling the vibration response through the feature vector  $\mathbf{x}$ . Hence, one can interpret that Table 4.2 quantifies the (expected) benefit of using the whole frequency spectrum instead of a set of features. Furthermore, it shows why previous material identification works based on features require over one second long vibration readings. The improvement in the average detection time when using PCA-FFT is above 45%, with improvement of 81% for material 10, and over 60% for materials 17, 23, 24, and 25. One can conclude that using the PCA-FFT features reduces both average identification time and standard deviation, making this input vector less sensitive to variations in the signal. Moreover, using the whole frequency spectrum information (from 2 Hz to 500 Hz) makes material identification more than twice as fast for these materials. Interestingly, the watercolour paper (material 32) is the only one identified faster using the state-of-the-art features than using the FFT.

#### 4.3.2.1 Evaluating the classification accuracy against standard Machine Learning methods

A comparison between the proposed RBE approach for continuous material identification and the most commonly used algorithms in the machine learning literature ( $k$ NN, ANN, and SVM) was performed. In order to conduct a fair comparison, the same amount of information was provided to each classification technique. The classification time results presented above showed that the RBE typically takes approximately one second to identify the cor-

rect material ( $\mu + 3\sigma$  secs) when using the vibration model based on  $\bar{\rho}$ , and more than two secs when using the vibration feature vector  $\mathbf{x}$ . New FFT-PCA vectors from one second data streams, and new feature vectors  $\mathbf{x}$  from 2.25 secs length data streams, were generated to train and evaluate the  $k$ NN, ANN and SVM classifiers, from the same datasets of the 34 materials. In the case of the RBE approach for material identification this means that, using  $\bar{\rho}$ , a classification failure is considered when the online algorithm takes more than four iterations (i.e. 1 second) to identify the material. Similarly, in the case of the feature vector  $\mathbf{x}$ , a classification failure was considered if the algorithm takes longer than nine iterations (i.e. 2.25 secs). This provides a fair comparison between the two methods as they are all required to classify the surfaces in the same amount of time with the same amount of information (although in the case of the RBE it can be observed in shorter time windows). Although the same amount of variance (97% of the original signal) was kept when projecting the FFT vector through PCA, taking longer temporal sequences of vibration readings changes the dimension of  $\rho(\omega)$ . Therefore, the training vectors for these algorithms have a higher dimension, specifically the  $\bar{\rho}$  used to train and evaluate the  $k$ NN, ANN, and SVM have dimension 49, after band pass filtering (between 2Hz and 500Hz) and PCA projection.

In order to determine the optimal value of  $k$  for the  $k$ NN algorithm, a systematic search procedure was implemented using Euclidean distance and 10-fold cross-validation. Test values from  $k = 1$  to  $k = 21$  were used to find adequate values of  $k$  and  $k = 5$  was concluded to perform best for both the FFT-PCA vector and the feature vector  $\mathbf{x}$ . In a similar manner, the topology of the two hidden layer ANNs was set by running 10-fold cross-validation tests for different numbers of units on each of the layers. The number of outputs corresponds to the number of materials to be identified,

i.e. 34 outputs encoding the materials as a binary output vector, while the number of inputs was 49 for the FFT-PCA input vectors and 6 for the feature vector  $\mathbf{x}$ . The best trade-off between the network size and the identification accuracy was achieved for networks with 34 units in both hidden layers for the feature vector  $\mathbf{x}$ . As a reflection of its higher dimensionality, 65 units in both hidden layers were used for the FFT-PCA input vector. The input vectors of the training and testing datasets were pre-processed through whitening (i.e. scaled to have zero mean and identity covariance matrix), and the networks were trained using the Levenberg-Marquardt algorithm. Finally, a One-Vs-One strategy multiclass approach was designed for the classification using SVMs, as it was found experimentally that it performs better than the One-Vs-All alternative. Although it provided better identification accuracy, the One-Vs-One approach needs to perform  $\frac{n(n-1)}{2}$  binary classifications for each input pattern, which in this case (since there is 34 materials) corresponds to 561 binary classifications. Therefore, this classification approach is the most computationally expensive, and difficult to implement in an online set-up. The kernel for all the SVMs was the standard Radial Basis Function (RBF), and its scale parameter was chosen through a grid search.

These three classification techniques (i.e.  $k$ NN, ANN, and SVM) are typically used in tactile material identification literature. Their classification accuracies were compared, using both FFT-PCA input vectors and the feature vector  $\mathbf{x}$ , with the proposed RBE approach “with a recognition timeout”. As expected, all techniques worked better when the PCA projected Fourier transform was used as an input ( $k$ NN 98%, ANN 85.5%, SVM 92.2%, and RBE 96.6%), instead of the feature vector ( $k$ NN 73.2%, ANN 54.2%, SVM 53.7%, and RBE 81.7%). It is worth noting that besides the performance decrease, the feature vectors were collected over more than twice the time.

Hence, there is a reduced identification rate and much slower identification time. Interestingly, the recursive material identification is the best performing classifier when the six dimensional feature vector  $\mathbf{x}$  is used (81.7%), since it actually performs nine different classifications with much lower certainties, but combines the material probabilities in an optimal way. All in all, when using the FFT-PCA input vector, the  $k$ NN algorithm outperforms the other approaches with a 98% average accuracy over the 10-fold cross-validation, followed by the RBE 96.6%. Nevertheless, the latter can typically identify the materials faster, meaning that the material is often identified in less than one second.

### 4.3.3 Multimodal Material Identification

This section presents experimental results of the material identification approach described in Section 4.2.4 using vibration information and thermal features ( $\theta_k$ ). Similar to Section 4.3.2, the 34 materials listed in Table 4.1 were classified using the proposed RBE and, for comparison purposes, the same three machine learning techniques (i.e.  $k$ NN, ANN and SVM). Similarly to the vibration only approach, the proposed recursive multimodal approach always identified the materials correctly if enough evidence was obtained, i.e. in a variable number of iterations. This occurs for both multimodal approaches, either using the FFT-PCA or the feature vector  $\mathbf{x}$  as input descriptors of the vibration. It is worth mentioning that processing, training, and testing procedures are identical to those presented in the vibration only section, i.e.  $\Delta t = 0.25$  secs.

Although the thermal features are also computed in 0.25 secs intervals, only 25 impedance and heat flux readings are obtained on each time window because of the differences in sampling frequencies. To estimate the thermal

power loss, the heat flux and impedance readings are averaged during the measuring interval, while the heat flux slope and error are computed with all the samples. The number of Gaussian functions in the mixture model of the thermal features is estimated using the change in the training data likelihood, which was typically 2, except for materials 4, 8, 12 and 23 which was estimated as 3.

A 10-fold cross-validation procedure was performed on 10 minutes of data sequences to identify the average time required by the RBE to successfully identify the materials. The average identification times across the ten iterations for both sets of input vectors (i.e. FFT-PCA vs. features both enhanced with thermal information) are shown in Table 4.3. Similarly to the vibration only approach, faster material classification is achieved using the PCA projected Fourier coefficients. However, in this case the time difference is smaller than before as the improvement of the classification using features is much greater when thermal information is included. Thus, a 40% reduction in the average identification time is achieved when thermal information is combined with the feature vector  $\mathbf{x}$  (0.4 secs vs. 0.67 secs in Table 4.2), and a 22% improvement is achieved when combined with the FFT-PCA vector  $\bar{\rho}$  (0.28 secs vs. 0.36 secs in Table 4.2) was achieved. The multimodal average identification time (and its standard deviation) for the feature sets  $\mathbf{x}$  is close to the time of the vibration only case when using FFT-PCA, nonetheless faster identification is achieved when using the Fourier coefficients to characterise vibration. Therefore, the average identification time when combining the FFT-PCA with thermal information is slightly above one single iteration, i.e. 0.28 secs. Attending to the standard deviations of the identification times, one could say that, if combined with thermal features, to have good identification accuracies ( $\mu + 3\sigma$ ) the multimodal approaches need to see data



Table 4.3: Average time (in secs) needed for multimodal material classification using 10-fold cross-validation. First number is the average time and second is the standard deviation.

Material	1	2	3	4	5
<b>FFT+PCA</b>	0.28+0.08	0.28+0.10	0.29+0.11	0.28+0.10	0.31+0.14
<b>Features</b>	0.32+0.16	0.39+0.29	0.39+0.28	0.29+0.11	0.59+0.54
Material	6	7	8	9	10
<b>FFT+PCA</b>	0.26+0.06	0.27+0.08	0.33+0.18	0.32+0.17	0.26+0.03
<b>Features</b>	0.29+0.12	0.30+0.14	0.50+0.31	0.64+0.50	0.41+0.35
Material	11	12	13	14	15
<b>FFT+PCA</b>	0.25+0.02	0.28+0.09	0.29+0.13	0.26+0.06	0.25+0.01
<b>Features</b>	0.26+0.04	0.31+0.15	0.71+0.60	0.35+0.17	0.39+0.34
Material	16	17	18	19	20
<b>FFT+PCA</b>	0.38+0.25	0.32+0.17	0.30+0.14	0.30+0.14	0.26+0.50
<b>Features</b>	1.11+1.55	0.50+0.48	0.39+0.17	0.53+0.47	0.31+0.15
Material	21	22	23	24	25
<b>FFT+PCA</b>	0.26+0.04	0.27+0.07	0.29+0.11	0.27+0.07	0.31+0.14
<b>Features</b>	0.27+0.09	0.31+0.17	0.41+0.27	0.34+0.19	0.53+0.44
Material	26	27	28	29	30
<b>FFT+PCA</b>	0.25+0.01	0.32+0.17	0.25+0.01	0.27+0.07	0.26+0.06
<b>Features</b>	0.25+0.01	0.48+0.43	0.27+0.07	0.32+0.18	0.29+0.12
Material	31	32	33	34	<b>Avg.</b>
<b>FFT+PCA</b>	0.25+0.01	0.29+0.13	0.30+0.12	0.26+0.06	<b>0.28+0.09</b>
<b>Features</b>	0.25+0.01	0.26+0.05	0.50+0.40	0.27+0.08	<b>0.40+0.28</b>

for approximately 0.5 secs (two iterations) in the case of the Fourier coefficients, and 1 second (four iterations) for the vector  $\mathbf{x}$ . These values will be used as the baseline for comparison with other learning approaches in the following section.

#### 4.3.3.1 Comparison of the multimodal recognition against commonly used machine learning approaches

As previously discussed, the multimodal RBE approach was compared with three machine learning techniques:  $k$ NN, ANN, and SVM. A 10-fold cross-validation was performed for each of the four algorithms using 10 minutes of data sequences from the 34 material dataset. As stated in Section 4.2.4.1, the datasets were whitened to have zero mean and identity covariance when evaluating  $k$ NN, ANN and SVM and their optimal parameters were obtained using the same criteria as in Section 4.3.2.1. Similarly to the vibration only approach, the  $k$ NN algorithm achieved optimal accuracy for  $k = 5$  in both two tested multimodal combinations (FFT-PCA and feature vector combined with thermal features). The topology of the ANN was selected to have two hidden layers with 65 units on each layer for the Fourier coefficients combined with the thermal features, and 30 hidden units for the combined vibration-thermal feature vector. Finally, SVMs multiclass classification was performed using One-Vs-One strategy and RBF kernels, for which parameters were tuned through a grid search. The input data streams for all classifiers were chosen to have a length according to the same principle used in Section 4.3.2.1. Therefore, relying on the results shown in see Table 4.2 the input data is classified for three iterations long (0.75 secs) when using the Fourier coefficients, and five (1.25 secs) for the vibration features. To provide a fair comparison with the presented approach the same number of

Table 4.4: Summary of material identification results.

	<i>k</i> NN	ANN	SVM	RBE
<b>Vibration - Features</b>	73.2%	54.2%	53.7%	81.7%
<b>Vibration - FFT+PCA</b>	98.0%	85.5%	92.0%	96.6%
<b>Multimodal - Features</b>	76.6%	82.9%	82.4%	93.8%
<b>Multimodal - FFT+PCA</b>	89.4%	94.6%	97.5%	98.6%

iterations was set as a deadline for successfully identifying the material using the RBE algorithm.

Table 4.4 summarises the average identification accuracy of all 4 classification techniques (i.e. RBE, *k*NN, ANN and SVM) using vibration only and the multimodal approaches. While the first two rows show the classification accuracy of vibration only material identification (i.e. either using FFT-PCA or the feature vector) presented in Section 4.3.2.1, the last two rows correspond to the multimodal cases. The selected criteria of the average time taken by the RBE to identify the materials was also followed in the multimodal cases. Therefore, despite the fact that the time window sizes used as input for some columns are different (i.e. 0.25 secs for the RBE and 1 second for the state-of-the-art classifiers), the same amount of information is fed into all classification techniques when evaluating the performance. Table 4.4 shows that, regardless of the classification technique and the way of characterising the vibration (i.e. Fourier coefficients or features), including thermal information improves the identification accuracy and shows that the proposed feature vector provides an excellent description of the thermal properties of the materials. An exception occurs for the *k*NN classifier when combining the Fourier coefficients with thermal features. In that case, the identification accuracy decreases from 98% to 89.4% as the data whitening

Table 4.5: Iterative estimation of material 17 using vibration only approach with FFT-PCA.

Iteration	0	1	2	3	4	5	6	7	8
$P(m_{17})$	0.03	0.03	0.01	0.01	0.05	0.61	0.98	0.99	1.00
$1 - P(m_{17})$	0.97	0.97	0.99	0.99	0.95	0.39	0.02	0.01	0.00
$\max P(m_i), i \neq 17$	0.03	0.88	0.95	0.99	0.95	0.39	0.02	0.01	0.00

preprocessing step (necessary to perform the combination) changes the distances between clusters. Moreover, despite the fact that the performance does not decrease when using vibration features for the  $k$ NN classifier its increase is relatively low (from 73.2% to 76.6%) compared to the other approaches evaluated. Table 4.4 also shows that using the full spectral range always results in better classification. Therefore, these results pinpoint the difficulty of identifying materials using the vibration features found in the literature as they typically extract only statistical measures of the vibration signal, completely ignoring the spectral components which actually characterise the interaction between the BioTAC fingertip and the material. In short, the best identification results were obtained when using the multimodal recursive Bayesian estimation (98.6%). It is worth noticing that this identification accuracy corresponds for a classification deadline (i.e. within the specified time range), notwithstanding the RBE algorithm could achieve perfect identification accuracy if longer time sequences are used.

To further illustrate the improvement of the RBE algorithm when multimodal information is provided, Tables 4.5 and 4.6 show the evolution of the posterior probabilities for the same test sequence (i.e. material 17) using only vibration and multimodal approaches. In both sequences the FFT-PCA was used to characterise the vibration and the initial prior probabil-

Table 4.6: Multimodal iterative estimation of material 17.

Iteration	0	1	2	3	4
$P(m_{17})$	0.03	0.98	0.99	0.99	1.00
$1 - P(m_{17})$	0.97	0.02	0.01	0.01	0.00
$\max P(m_i), i \neq 17$	0.03	0.01	0.01	0.01	0.00

ities (i.e. iteration 0) were uninformative. Both tables show the probability of the correct material  $P(m_{17})$ , the combined probability of all the wrong materials  $1 - P(m_{17})$ , and the highest probability of a wrong material  $\max P(m_i), i \neq 17$ . While Table 4.5 shows that it takes 5 iterations to identify material 17 with a probability of  $P(m_{17}) = 0.61$  using vibration only, Table 4.6 shows material 17 is identified in the first iteration with a probability of  $P(m_{17}) = 0.98$ . As it can be clearly seen, including thermal information makes the identification correct from the first iteration on, reducing in this case the identification time from 1.25 secs to 0.25 secs, i.e. from 5 to 1 iteration. It is worth mentioning that once the algorithm has detected the correct material with probability 1.00 it maintains that probability value as the prior for any other material in the next iteration will be zero.

## 4.4 Summary

This chapter presents a multimodal approach to tactile based continuous material identification. Despite the fact that tactile sensors provide temporal sequences of readings, state-of-the-art material recognition approaches are episodic, i.e. a whole sequence of readings is processed to identify the material. This work presents a continuous identification technique using recursive estimation of the probability of identifying a set of materials. This allows faster identification of most materials and does not require several

exploratory movements. The experimental results show that when enough evidence is gathered, the system achieves perfect recognition using the experimental set of 34 materials.

While existing approaches for robotic material identification mostly rely on vibration information, including thermal features was shown to enhance the recognition accuracy and reduce the time required for successful identification. Moreover, standard tactile identification techniques typically require a sequence of at least one second to classify materials. Using the proposed continuous material identification approach, a robot endowed with tactile sensors can identify materials in an average of 0.28 seconds with a very small deviation from that time lapse. Although the proposed approach allows the integration of one or more sensing modalities, this high detection speed is due to the use of multimodal information. Hence thermal sensing enables a faster identification than a vibration only approach, i.e. 0.36 seconds. Additionally, including thermal information brings the average material identification time very close to the window size. Faster identification could be achieved by reducing the size of the window, yet the selection of  $\Delta t = 0.25$  was empirically found to provide an excellent recognition time vs. accuracy trade-off. The proposed approach was shown to perform better than any of the existing alternatives when using the same data set.

Material identification can be used to enhance robotic grasping and manipulation of objects. For instance, grasping an object made of a fragile material requires a precise estimation of the contact forces, i.e. to grasp the object without breaking it. Chapter 5 presents an innovative technique to grasp objects of unknown shape by adapting the fingers to reach an adequate set of contact forces.

## Chapter 5

# Precision Grasp control through Tactile Sensing

### 5.1 Introduction

The identification of the material from which an object is made allows one to infer some properties such as the coefficient of friction or the object compliance. Such properties are important to endow robots with grasping capabilities. Robotic grasping of objects is a core skill to allow robots to manipulate objects in real environments. Robotic manipulation of objects requires grasping systems capable of ensuring that the contact between the robot and the object is never lost, as this would result in the object being dropped and damaged if the object is fragile.

A grasp is defined as a set of contacts on the surface of an object that restricts the potential movements of the object when external perturbation forces occur [169]. Humans use precision grasps to increase control on the object (i.e. using only the fingertips) as opposed to power grasp where the aim is to increase strength [170, 171]. While grasping strategies for robotic

grippers are typically limited to using only the thumb and index finger facing each other, or to hook grip the object (e.g. holding a suitcase), state-of-the-art robotic dexterous hands have enough degrees-of-freedom (DoF) to perform different types of grasps [172]. Finding a suitable set of contacts to grasp an object is usually addressed by either using planning and execution strategies (i.e. estimating the contact points for a given object) or using sensory feedback control. While the first uses a model of the object to plan a set of contacts that will result in a stable grasp, the latter adapts the joints configuration until finding a suitable grasp, for instance, performing a sequence of exploratory movements. However, most of the state-of-the-art approaches to robotic grasping rely on previous known information about the object geometry and surface material.

This chapter presents an approach for grasping objects using a Shadow dexterous hand when information about the object geometry is not available, and ensuring neither the robot or the object are damaged. The contributions of this chapter are three-fold. Firstly, the contact forces of the BioTAC tactile sensor are modelled using the impedance and pressure signals. Secondly, a technique to approximately detect the position of an object and find the homogeneous transformation between a Shadow Robotic hand and a Microsoft Kinect sensors reference frames is presented. Finally, a precision grasp controller is implemented in a Shadow Robotic hand combining Cartesian position and contact force control.

The work presented in this chapter has been partially published in the following article:

- Reliable object handover through tactile force sensing and effort control in the Shadow Robot hand. In *IEEE International Conference on Robotics and Automation* (ICRA 2017) [8].



The remainder of the chapter is structured as follows. Section 5.2 presents the precision grasp controller used to grasp objects with unknown geometry, and the technique used to estimate the contact force using the BioTAC tactile sensor. Section 5.3 proposes a method to obtain the centre position of an object in a tabletop manipulation scenario using a Microsoft Kinect sensor. Section 5.3 also presents a technique to obtain the homogeneous transformation between the reference frames attached to the Kinect sensor and the Shadow robot forearm, and provides insights to the practical implementation of the proposed approach. Experimental results are presented in 5.4. Finally, Section 5.5 concludes the chapter and summarises the findings.

## 5.2 Precision Grasp Control through Continuous Tactile Sensing

Existing reports in the grasping literature approach the grasping problem using either planning and execution, or grasp control strategies (see Chapter 2). However, they often rely on information about the objects geometry, which it is not always accurate. It is worth noting that inaccurate contact forces applied during the grasp may damage the object or the robot. For instance, applying a large wrench on a rigid object might damage the robot or applying too much force on a fragile object might break it.

This section presents a grasp control strategy to grasp objects of unknown shape while ensuring the integrity of the robot and the object is maintained in a stable grasp. Using the surface material classification approach presented in Chapter 4 the robot could identify the material from which the object is made and use the material properties to find a suitable contact force configuration that ensures neither the robot nor the object are damaged. The

proposed approach drives the fingers towards the object and adapts the contact configuration to grasp it using a Shadow robotic hand equipped with BioTAC tactile sensors. Section 5.2.1 presents the technique used to estimate the contact force using the BioTAC tactile sensor, and Section 5.2.2 presents the proposed controller to perform precision grasps of unknown shaped objects.

### 5.2.1 Cartesian Force Estimation using the BioTAC

If the hand and finger joints have torque sensors attached, the wrench applied to an object can be easily computed from the measured torques  $\mathbf{\Gamma}$  using the pseudo-inverse Jacobian matrix for the corresponding configuration  $J(\mathbf{q})^\dagger$  since  ${}^B\mathbf{f} = J(\theta)^\dagger \mathbf{\Gamma}$ . However, the sensors included in the joints of the Shadow hand measure the differential tension on the tendons [5], not the applied torque in the joints. Therefore, this section presents an alternative approach using the SynTouch BioTAC [4] multimodal tactile sensors to estimate the contact forces in the fingertips  ${}^E\mathbf{f}$  instead of the applied wrench.

Although the BioTAC is not a force sensor, we can estimate the contact force from its measurements. Specifically, the force applied in the direction of the normal to the object surface can be approximated by using the pressure, the contact area, and the normal vector to the object surface. While the pressure measurements are obtained directly from the BioTAC's raw data, the contact area and the normal vector to the object surface can be estimated from the skin deformation. Upon contact, the increase in the pressure ( $P$ ) measured by the BioTAC can be converted into the norm of a contact force ( $|\mathbf{f}|$ ) by using the relationship  $|\mathbf{f}| = Pa$ , where  $a$  is the contact area with the object. Although the pressure is obtained from the sensor, the contact area can only be estimated through impedance measurements from 19 electrodes

located across the finger core. The measured impedances are directly related to the distance between the core and the rubber skin at the electrode's corresponding locations. Thus, the impedance value with respect to the resting level (i.e. no contact) decreases when the rubber skin is deformed.

The BioTAC sensor and the approach detailed in Chapter 4 are used to approximate the contact area corresponding to each electrode ' $i$ ' as a circle of radius  $r_i$  equal to half the distance between the electrode and its nearest neighbour. Although the fingertip of the BioTAC also applies a torque at the contact point, there is no way to estimate it, nor to compute the component of the force tangential to the object without knowing the friction coefficient between the skin and the object. Therefore, it is assumed that the full length of the force is applied in the direction perpendicular to the contact surface. The technique presented in [173] is used to estimate the contact direction based on the unit vectors normal to the BioTAC fingertip at each electrode position. Similar to the approach used to compute the area (see Chapter 4), a weighted average of the normal vectors using the change in the corresponding impedances relative to the resting levels is used to compute the contact direction. Given the normal vectors for the BioTAC electrodes with respect to the reference frame at its core  $\hat{\mathbf{n}}_i$ ,  $i = 1, \dots, 19$ , the total estimate of the contact force can be computed as:

$${}^E\mathbf{f} = \frac{|\mathbf{f}|}{|\sum_i \lambda_i \hat{\mathbf{n}}_i|} \sum_i \lambda_i \hat{\mathbf{n}}_i, \quad (5.1)$$

where the force norm  $|\mathbf{f}|$  is as defined above and  $\lambda_i \in [0, 1]$  is an impedance dependent scale factor. The scaling factor  $\lambda_i$  is a piecewise linear function of the average impedance value  $e_i$  of each electrode (as described in Section 4.2.4). Thus,  $\lambda_i$  is zero at the resting level (or above), meaning no contact at the electrode position, and it linearly increases to 1 for decreasing impedances

up to a minimum threshold  $e_m$  (experimentally set to  $e_m = -400$ ) and is 1 for values below the threshold.

It is worth noting that Equation 5.1 corresponds to the contact force in the reference frame of the fingertip. The proposed precision grasp controller uses  ${}^E\mathbf{f}$  as an input to adapt the hand joints upon contact to reach a desired contact force configuration.

### 5.2.2 Precision Grasp Controller

In this section, a strategy to perform precision grasps using three fingers and a Cartesian position/contact force controller is presented. The proposed approach has been proven to effectively grasp a large range of objects, without having *a priori* information about their shape and size, by driving the fingers towards a Cartesian position (typically inside the object) and, once the fingers are in contact with the object, adapting the robot configuration to reach a given set of contact forces. Upon contact, the transition between Cartesian position and force control was designed to be sufficiently smooth to avoid contact loss and sudden changes in the trajectory of the fingers.

The controller has two inputs: a desired Cartesian position  $\bar{\mathbf{x}}$  relative to the coordinate frame attached to the palm of the robot, and a contact force  $\bar{\mathbf{f}}$  relative to the coordinate frame of the BioTAC tactile sensor equipped as the finger's end-effector. Assuming the set of contact forces and the reference Cartesian positions for all fingers are known, the precision grasp controller will drive the fingers to reach their input positions or contact forces depending on whether there is contact or not. The transition between driving the finger towards  $\bar{\mathbf{x}}$  and adjusting to  $\bar{\mathbf{f}}$  (i.e. from position to contact force control) needs to be sufficiently smooth to prevent the contact from being lost. The proposed precision grasp controller ensures that the transition is smooth

using a piece-wise linear function of the modulus of the contact force defined as:

$$g(\mathbf{f}) = \begin{cases} 0 & \text{if } |\mathbf{f}| \leq a \\ \frac{|\mathbf{f}| - a}{|\bar{\mathbf{f}}| - a} & \text{if } a < |\mathbf{f}| < |\bar{\mathbf{f}}| \\ 1 & \text{if } |\bar{\mathbf{f}}| \leq |\mathbf{f}| \end{cases} \quad (5.2)$$

where  $a$  is a threshold that defines the force magnitude of  $|\mathbf{f}|$  at which  $g(\mathbf{f})$  becomes greater than 0. The threshold value  $a = 0.005$  was found to provide smooth transitions between position and contact force control.

When the BioTAC does not detect any contact the proposed approach behaves as a Cartesian position controller. As the fingers of the Shadow robotic hand only have 3 joints (4 in the thumb) the workspace of each finger is limited and, therefore, reaching a desired pose might not be possible. Sometimes a given position could be reached but only for a small number of orientations. This issue was addressed by only controlling the position of fingertip, i.e. the orientation of the finger is not controlled. Thus, the proposed approach finds a suitable pose that guarantees the finger's end-effector to reach the desired Cartesian position.

Although the plant (i.e. the model of the system and the actuator) of the robotic hand is unknown, a first-order transfer function with an integrator of the form:

$$Y(s) = \frac{\kappa}{(\tau s + 1)s}, \quad (5.3)$$

is assumed, where the parameters  $\kappa$  and  $\tau$  define the first-order system. Since the aim is to always reach the desired position when the finger is not in contact but no time constraints are considered, a proportional controller was used to drive the fingers towards  $\bar{\mathbf{x}} = (o_x, o_y, o_z, r_x, r_y, r_z)$ . It is worth

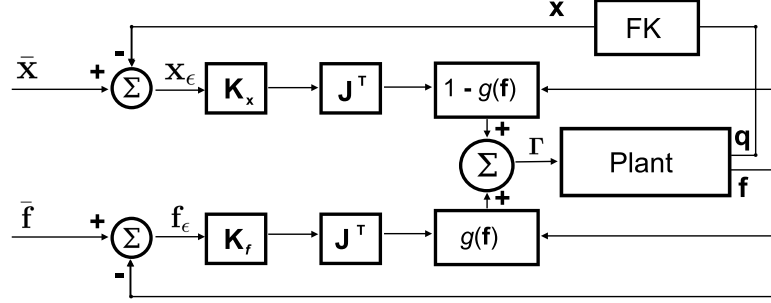


Figure 5.1: Precision grasp control diagram.

mentioning that although the orientation values  $r_x$ ,  $r_y$ , and  $r_z$  are different to zero, the proposed approach does not control the orientation of the finger as the gain  $K_x$  zeroes the orientation components of the pose error  $\mathbf{x}_e = \bar{\mathbf{x}} - \mathbf{x}$ . Thus, the Cartesian position controller feeds back the joint angles of the finger's kinematic chain  $\mathbf{q}$  and computes the position of the end-effector  $\mathbf{x}$  using Forward Kinematics (see Section 3.5.1).

While the proposed controller only considers  $\bar{\mathbf{x}}$  when there is no contact, it solely relies on  $\bar{\mathbf{f}}$  when  $|\mathbf{f}| \geq |\bar{\mathbf{f}}|$ . Another proportional controller was used with the contact force  $\mathbf{f}$  estimated using BioTAC tactile sensors (see Section 5.2.1). Figure 5.1 shows a block diagram of the proposed control system used for each finger. The torques  $\mathbf{\Gamma} \in \mathbb{R}^n$  applied in each iteration for each of the  $n$  joints of the kinematic chain are computed as:

$$\mathbf{\Gamma} = (1 - g(\mathbf{f}))\mathbf{J}^T\mathbf{K}_x(\bar{\mathbf{x}} - \mathbf{x}) + g(\mathbf{f})\mathbf{J}^T\mathbf{K}_f(\bar{\mathbf{f}} - \mathbf{f}), \quad (5.4)$$

where  $\mathbf{J}$  is the  $n \times 6$  Jacobian matrix for the corresponding finger,  $\mathbf{K}_x \in \mathbb{R}^6$  are the gains of the Cartesian position controller, and  $\mathbf{K}_f \in \mathbb{R}^6$  are the gains of the force controller.

The geometrical description of the hand is given by the Denavit-Hartenberg

Table 5.1: Thumb D-H parameters of a standard Shadow Dexterous hand [174], where  $d_i$ ,  $a_i$ ,  $\alpha_i$ , are respectively the offset, length and twist of the link  $i$ , and  $\theta_i$  is the angle of the  $i^{th}$  joint.

Joint	$d_i$	$a_{i-1}$	$\alpha_{i-1}$	$\theta_i$		Max.	Min.
1	0	0	0	$\theta_1$	$\theta_1$	$60^\circ$	$-60^\circ$
2	0	0	$90^\circ$	$\theta_2$	$\theta_2$	$75^\circ$	$75^\circ$
3	0	$b_2$	0	$\theta_3$	$\theta_3$	$15^\circ$	$-15^\circ$
4	0	0	$90^\circ$	$\theta_4$	$\theta_4$	$30^\circ$	$-30^\circ$
5	0	$b_3$	0	$\theta_5$	$\theta_5$	$90^\circ$	$-10^\circ$

(D-H) parameters [175]. An example of the D-H parameters of the thumb in a standard Shadow robot hand is shown in Table 5.1. Therefore, a kinematic model of the robot can be obtained using the D-H parameters to obtain the homogeneous transformations as detailed in Section 3.5.1. Additionally, the mathematical description of the hand motion can be obtained using the equation of motion [175]. For each chain of links (i.e. fingers) the motion of the robot can be represented in matrix form as a set of coupled differential equations:

$$\mathbf{\Lambda} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}), \quad (5.5)$$

where  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$  and  $\ddot{\mathbf{q}}$  are the joint coordinates, velocities and accelerations,  $\mathbf{M}$  is the joint-space inertia matrix,  $\mathbf{C}$  is the Coriolis and centripetal coupling matrix,  $\mathbf{G}$  is the gravity loading, and  $\mathbf{\Lambda}$  is the vector of actuator forces associated with the joint coordinates  $\mathbf{q}$ . However, the dynamic model of the hand was not used in this thesis as the precision grasp control was not implemented on simulation and the controller parameters were adjusted empirically.

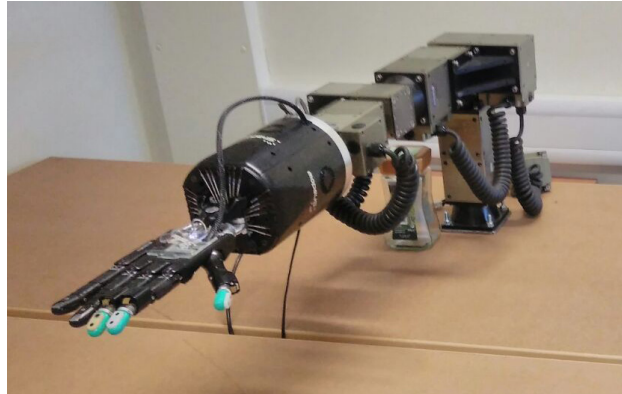


Figure 5.2: The Shadow robot hand attached to a Schunk arm.

### 5.3 Precision Grasp Control for Tabletop Manipulation

In order to perform a grasp, a Cartesian position inside the object is passed to the proposed grasp controller, which will result in the fingers eventually being in contact with the object surface. Upon contact, the robot adapts to reach a desired set of contact forces that was previously selected to suit a point contact with friction model (i.e. Coulomb friction model) for a three finger precision sphere grasp. Although the generated configuration of the finger poses and contact forces does not imply the system will always perform a stable grasp, high friction between the rubber skin and the object is assumed and, therefore, the object grasp will display a stable behaviour in spite of the inaccuracies.

This section presents an approach to finding the Cartesian position inputs so as to use the proposed precision grasp controller in a tabletop manipulation scenario. The proposed approach is implemented in a Shadow robotic hand attached to a Schunk arm (see Figure 5.2). Although the precision grasp



control detailed in this chapter assumes the arm stays fixed and the object is reachable by the hand, it can be extended to drive the arm so the object is within the hand workspace. A Kinect sensor facing the robot from a lateral position, from a distance of approximately 70 cm was used to find the object and the homogeneous transformation between the reference frames was defined by the Kinect sensor and the robot forearm poses.

### 5.3.1 Scene Segmentation using Microsoft Kinect Sensor

Although the Kinect sensor also features an RGB camera and a multi-array microphone, this work only uses the embedded depth sensor consisting of an infra-red laser projector combined with a monochrome CMOS sensor [176]. Therefore, the Kinect provides a depth map  $\mathbf{Z}$  of the scene in front of the sensor, which is assumed to be a tabletop manipulation scenario for one object. The depth map  $\mathbf{Z}$  was processed using the Point Cloud Library (PCL) [177] and the RANSAC algorithm [164] was used repeatedly to extract the objects in the scene.

The first stage of the scene segmentation is removing all points in  $\mathbf{Z}$  that are at a distance greater than 1 metre from the Kinect sensor as the object is at a distance of approximately 70 cm. Removing far points allows subsequent segmentation of objects to be performed faster than using all points in  $\mathbf{Z}$  as it reduces the number of trials required by the RANSAC algorithm to ensure certain probability of success (see Section 3.4.2). Then, the resultant depth map  $\mathbf{Z}_1$  was used to find the table plane using the RANSAC algorithm. Using a point  $\mathbf{x}$ , a normal vector  $\hat{\mathbf{n}} = (n_x, n_y, n_z)$  and a distance  $d$  of the plane from the origin, the equation of the plane can be defined by its Hessian normal form  $\mathbf{x} \cdot \hat{\mathbf{n}} - d = 0$ . Thus, a model of the plane  $\mathbf{m}_\pi = \{n_x, n_y, n_z, d\}$  was used

in RANSAC to find the points in  $\mathbf{Z}_1$  that best match the shape of a plane. Upon completion, the points matching the plane  $\mathbf{\Pi}$  were extracted from  $\mathbf{Z}_1$  and the resultant depth map  $\mathbf{Z}_2$  was then used to find the forearm of the Shadow robotic hand.

The forearm of the Shadow robotic hand was modelled as a parametric cylinder with a radius between 3 and 10 cm. A cylinder in 3D space can be defined as  $\|\mathbf{x} - \mathbf{p} - (\mathbf{x} - \mathbf{p}) \cdot \hat{\mathbf{v}} \cdot \hat{\mathbf{v}}\|^2 = r^2$ , where  $\mathbf{p} = (p_x, p_y, p_z)$  is a point on its axis,  $\hat{\mathbf{v}} = (v_x, v_y, v_z)$  is the axis direction, and  $r$  is the radius of the cylinder. Therefore, the cylinder model was defined by seven coefficients  $\mathbf{m}_c = \{p_x, p_y, p_z, v_x, v_y, v_z, r\}$  that the RANSAC algorithm used to find the best match in  $\mathbf{Z}_2$ . The points estimated to fit in the cylinder model  $\mathbf{C}$  were extracted from  $\mathbf{Z}_2$  to obtain another depth map  $\mathbf{Z}_3$ .

$\mathbf{Z}_3$  contains the points corresponding to the robotic hand (except the forearm) and the object. A  $k$ -means clustering approach [155] was used to partition the points in  $\mathbf{Z}_3$  into two clusters. In order to find the cluster that is closer to the table, the euclidean distances from the centre of each cluster  $\boldsymbol{\mu}_i$  to the plane  $\mathbf{\Pi}$  were computed as:

$$d_i(\boldsymbol{\mu}_i, \mathbf{\Pi}) = |\boldsymbol{\mu}_i \hat{\mathbf{n}} - d|, \quad (5.6)$$

where  $i \in \{1, 2\}$  denotes the index of the cluster. The cluster that is closer to the plane  $\arg\min_{i \in \{1, 2\}} d_i(\boldsymbol{\mu}_i, \mathbf{\Pi})$  was selected as the depth map of the object  $\boldsymbol{\Omega}$  placed on the tabletop. It is worth noting that  $\boldsymbol{\Omega}$  (similarly to other depth maps detailed in this section) only reflects points from the point cloud that can be seen from the sensor viewpoint and, as some faces of the object are totally or partially occluded, it does not represent the whole surface of the object.

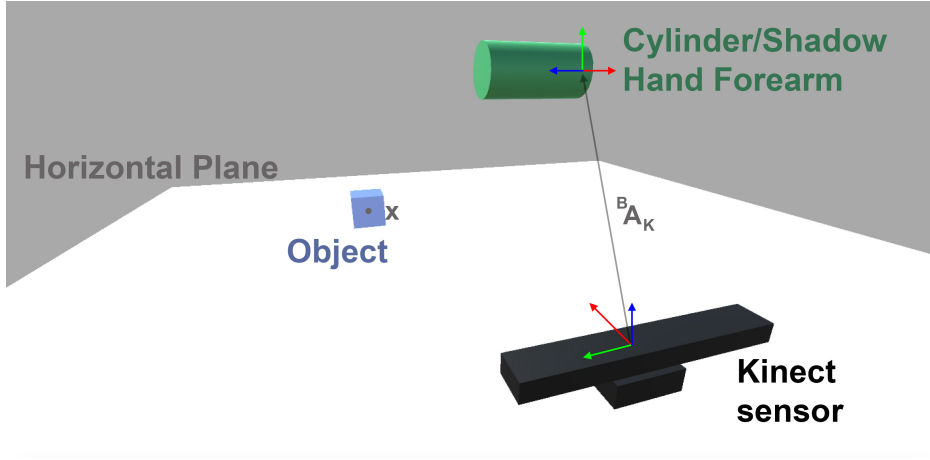


Figure 5.3: Tabletop manipulation scene diagram.

### 5.3.2 Kinect-Robot Frame Transformations

All points in the depth maps, extracted as described in Section 5.3.1, are relative to the reference frame attached to the Kinect sensor. However, the Cartesian position input  $\bar{\mathbf{x}}$  of the precision grasp controller (see Section 5.2.2) is relative to the reference frame defined by the palm joint pose (i.e. position and orientation) of the Shadow robotic hand. For simplicity we will refer thereafter to the coordinate frames defined by the palm and forearm joint position and orientations as the palm reference frame  $P$  and the base (or forearm) frame  $B$ . As the transformation between the Shadow robot hand's forearm and palm reference frames can be obtained using the robot's kinematic chain and forward kinematics, the problem is now to find the homogeneous transformations between the Kinect sensor and the forearm frame of the Shadow hand.

As detailed in Chapter 3, the homogeneous transformations  ${}^j\mathbf{A}_i$  allow any point to be expressed in the coordinate frame  $i$  relative to the frame  $j$ . Therefore, the homogeneous transformation between the Kinect sensor and

the base frame of the Shadow hand is defined as:

$${}^B\mathbf{A}_K = \begin{pmatrix} {}^B\mathbf{R}_K & {}^B\mathbf{o}_K \\ \mathbf{0}^T & 1 \end{pmatrix}, \quad (5.7)$$

where  ${}^B\mathbf{o}_K$  is the origin of the coordinate frame  $B$  relative to the frame  $K$ ,  ${}^B\mathbf{R}_K$  is the matrix representing their relative orientation and  $\mathbf{0}$  is a 3-dimensional column vector of zeros. Figure 5.3 shows a diagram of the tabletop manipulation scenario including the reference frames attached at the Kinect sensor and the robot forearm. The  $x$ ,  $y$  and  $z$  orientation axis of the reference frames (i.e. the Kinect and the forearm frames) are represented using a red, a green and a blue arrow respectively, and the transformation between the Kinect frame and the forearm frame  ${}^B\mathbf{A}_K$  is represented using a grey arrow. Although  ${}^B\mathbf{A}_K$  is unknown as it depends on the location of the Kinect and the robot, an approach to estimate it using the depth maps of the cylinder  $\mathbf{C}$  and the plane  $\mathbf{\Pi}$  (see Section 5.3.1) is presented below.

The matrix  ${}^B\mathbf{R}_K$  can be built using the normalised vectors that define each axis of the coordinate frame  $B$  relative to frame  $K$ . The  $z$ -axis of frame  $B$  is computed as  $\hat{\mathbf{v}}_z = \frac{\mathbf{v}}{|\mathbf{v}|}$ , where  $\mathbf{v}$  is the axis vector of the cylinder obtained during its segmentation using the Kinect. The palm of the robot is assumed to be aligned with the  $z$ -axis of the base frame  $B$  and facing the table. Considering the cross product of two linearly independent vectors is a vector perpendicular to both, it is possible to obtain the vector that defines the  $x$ -axis as the cross product of  $\hat{\mathbf{v}}_z$  and the normal vector to the tabletop as  $\hat{\mathbf{v}}_x = \hat{\mathbf{n}} \times \hat{\mathbf{v}}_z$ , where  $\hat{\mathbf{n}}$  is the normal vector to the tabletop surface obtained during the segmentation of the plane. This approach works under the assumption of the palm facing the table, which suits the experimental set up used in this work, i.e. the robotic arm is fixed. However, this restriction could be overcome through deeper analysis of the arm configuration to suit

any relative pose of the palm joint. In a similar manner it is possible to obtain the vector defining the  $y$ -axis  $\hat{\mathbf{v}}_y = \hat{\mathbf{v}}_z \times \hat{\mathbf{v}}_x$  and, therefore, define the orientation of the reference frame  $B$  relative to  $K$  as:

$${}^B\mathbf{R}_K = [\hat{\mathbf{v}}_x \hat{\mathbf{v}}_y \hat{\mathbf{v}}_z]^T. \quad (5.8)$$

To obtain the origin  ${}^B\mathbf{o}_K$ , we first compute the position of the mean point in the depth map of the cylinder  $\mathbf{C}$ . The mean point  $\bar{\mathbf{c}}$  is approximately centred in the  $y$  and  $z$  axis of the cylinder. To find the centre of the cylinder, the point  $\bar{\mathbf{c}}$  is projected on the axis  $\hat{\mathbf{v}}_z$  as:

$$\bar{\mathbf{c}}' = \mathbf{p} + [(\bar{\mathbf{c}} - \mathbf{p}) \cdot \hat{\mathbf{v}}_z] \hat{\mathbf{v}}_z. \quad (5.9)$$

The robot's forearm reference frame is attached to the base of the cylinder but the average of the points detected by the Kinect is in the middle of the cylinder and, therefore, the point  $\bar{\mathbf{c}}'$  needs to be translated along the  $z$ -axis to match the base frame origin. Using the height of the forearm cylinder  $h = 18.3$  cm. the origin of the frame can be obtained as:

$${}^B\mathbf{o}_K = \bar{\mathbf{c}}' - \frac{h}{2} \hat{\mathbf{v}}_z. \quad (5.10)$$

### 5.3.3 Grasping Objects with Unknown Shape

Instead of performing a detailed grasp planning strategy, the proposed approach moves the fingers towards the object and adjusts the contact forces to find a suitable configuration for grasping when the fingertips touch the object. To perform the grasp, a force vector relative to the fingertip reference frame  ${}^E\mathbf{f}$  and a Cartesian position relative to the frame defined by the robot's palm position and rotation  ${}^P\mathbf{x}$  are passed as inputs to the precision grasp controller presented in Section 5.2.2. While the  ${}^E\mathbf{f}$  vector is assumed

to provide a successful object grasp configuration,  ${}^P\mathbf{x}$  is a Cartesian position inside the object. Thus, the position control always achieves contact between the finger and the surface of the object leading smoothly to contact force control.

An approximation of the object position can be computed using the depth map of the object  $\Omega$  described in Section 5.3.1. As the object is unknown and the depth map only represents the surface of the object that is visible from the viewpoint of the Kinect sensor, it is not possible to estimate the centre of the object accurately. The object centre is assumed to lie at some point in the vector  $\hat{\mathbf{v}}_z$ , i.e. is roughly aligned with the arm. Therefore, the centre of the object is computed by projecting the average position of the depth map  $\Omega$  to the vector  $\hat{\mathbf{v}}_z$  as:

$${}^K\mathbf{x} = \bar{\mathbf{o}} - [(\bar{\mathbf{o}} - \mathbf{p}) \cdot \hat{\mathbf{v}}_x] \hat{\mathbf{v}}_x, \quad (5.11)$$

where  $\bar{\mathbf{o}}$  is the average point position of  $\Omega$ .

If the homogeneous transformation matrix  ${}^B\mathbf{A}_K$  and the kinematic chain of the robot hand are known, it is possible to transform any point relative to the coordinate frame  $K$  to be relative to the palm frame  ${}^P\mathbf{x}$  using Forward Kinematics (see Section 3.5.1). As the object position is (approximately) known, one option is to compute  ${}^P\mathbf{x}$  and command all three fingers to move towards that position. Instead, a virtual sphere is set inside the object and, therefore, the grasp is approached as a precision sphere grasp. Figure 5.4 shows a flow chart of the proposed grasp operation to be performed by the shadow robot hand.

Thus, a virtual sphere with radius  $r_1$  and centred at the object position  ${}^K\mathbf{x}$  is defined and its grasp locations denoted as  ${}^K\mathbf{x}_j$ , where  $r_1 < r_0$  is the radius of the virtual sphere,  $r_0$  is the radius of the depth map  $\Omega$ ,  $j = 1$  is the thumb,  $j = 2$  is the first finger, and  $j = 3$  is the middle finger. The contact

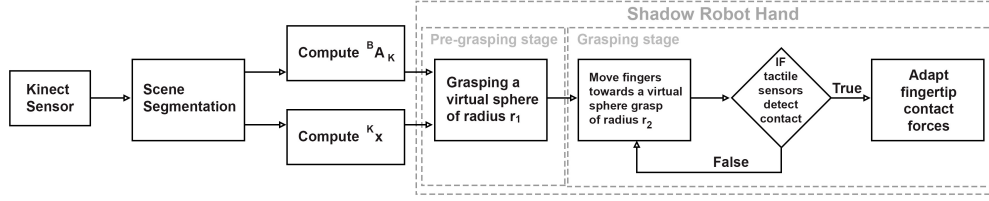


Figure 5.4: Flow chart of the proposed precision sphere grasp operation.

location of the thumb is the intersection of the vector  $\hat{\mathbf{v}}$  and the virtual sphere that is closest to the hand and is computed as  ${}^K\mathbf{x}_1 = {}^K\mathbf{x} - \hat{\mathbf{v}}_z r_1$ . The contact locations of the first  ${}^K\mathbf{x}_2$  and middle  ${}^K\mathbf{x}_3$  fingers were computed by converting  ${}^K\mathbf{x}_1$  to spherical coordinates and finding the points in the sphere that lie at azimuth angles of  $\pm 170^\circ$  relative to  ${}^K\mathbf{x}_1$ .

When trying to reach  ${}^K\mathbf{x}$  the trajectory of the finger is discontinued upon collision as the target position is inside the object. Depending on the initial configuration of the joints, the size and shape of the object, the grasp could fail if the trajectories are interrupted in a joint configuration that prevents the precision grasp controller adapting to a suitable contact force configuration. For instance, if the initial position of all fingers is facing the same side of the object the fingers might push the object instead of grasp it. To deal with this issue, the proposed grasp approach was split into two stages: pre-grasping and grasping. During the pre-grasping stage, the fingertips are driven to find contact locations with a virtual sphere of  $r_2 > r_0$  (typically  $r_2 = 6$  cm), which is assumed to be bigger than the object, i.e. the pre-grasping stage will result in no contact with the object. The pre-grasping stage could be finished once the controller has reached the desired configuration within a distance threshold. Instead, after 5 seconds the grasping stage is triggered by passing new contact locations for a virtual sphere of  $r_1 = 1$  cm (i.e. inside

the object) to the precision grasp controller. As the motion required during the pre-grasping stage is relatively small, 5 seconds is sufficient to reach any hand configuration if the precision grasp controller is tuned, i.e.  $\mathbf{K}_x$  selected to provide optimal response without oscillations.

During the grasping stage, the fingertips are driven towards the object in order to grasp the virtual sphere of radius  $r_1 = 1$  cm placed inside the object. However, the fingertips will always contact the object surface before reaching the commanded position. The BioTAC tactile sensors detect arising contact forces and the precision grasp controller will adapt the joint angles to reach the desired force contact configuration. As detailed in Section 5.2.2, the transition between driving the fingers towards the contact locations of the virtual sphere and adapting to the desired grasp force configuration is set progressively as a function of the magnitude of the contact forces.

When the contact force of a finger is small the target contact locations of the virtual sphere will still play a significant role on the commanded torques to the finger so that the contact force increases. Once the measured contact force reaches the magnitude of the desired contact force, the precision grasp controller will adapt relying only on contact force control. If the contact force magnitude increases above the commanded contact force magnitude, the force control module would reduce the pressure applied on the object. Hence, the precision grasp controller adapts the finger's joint positions to find a suitable grasp force configuration and guarantees the contact with the object is maintained constant.





(a) Tabletop manipulation set-up.

(b) Alternative set-up.

Figure 5.5: Tabletop manipulation experimental set-ups.

## 5.4 Experimental Results

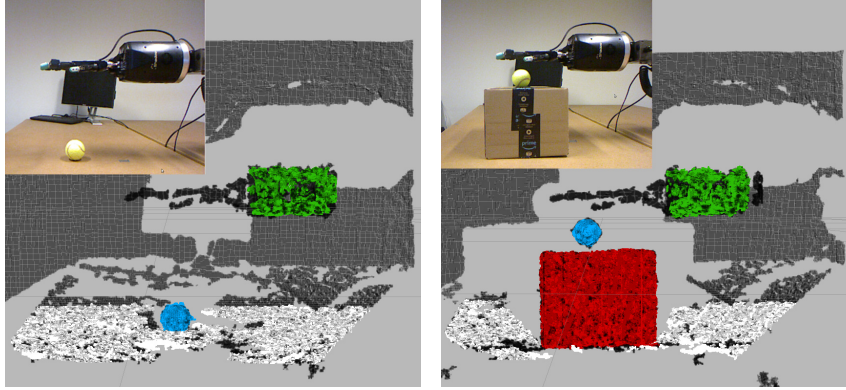
The proposed approach was evaluated on a tabletop manipulation scenario using a Microsoft Kinect sensor and a Shadow Robot Hand equipped with BioTAC tactile sensors. The hand was attached to a Schunk arm, which stays fixed in all of the experiments as shown in Figure 5.5. The initial configuration of the robot manipulator was set in stretch position (i.e. the arm and the hand are straight and horizontal). The Kinect sensor was placed on a box that raised it 19 cm from the table plane facing the robot from a lateral position at a distance of approximately 70 cm from the object and the robot palm. It is worth noting that the proposed approach would detect the table, the object and the forearm for any configuration of the Kinect and, therefore, it is not limited to the experimental set-up described.

The approach presented in Section 5.3 finds an object in a tabletop manipulation scenario and its homogeneous transformation to the reference frame attached at the palm joint position and orientation. Although the object position could be reached by driving the manipulator arm to place the palm

of the robot close to the object, the experimental set up used to grasp objects raised them using a box so the object is in the hand workspace, i.e. all fingertips can reach the object position. This implies that an additional plane must be segmented during the scene processing. Section 5.4.1 details the additional steps required when the object is placed on a box and presents experimental performance results in comparison with the approach described in Section 5.3.1. Section 5.4.2 evaluates the proposed scene segmentation approach and Section 5.4.3 presents experimental results from grasping a number of objects with different shapes and materials using the proposed precision grasp controller.

#### 5.4.1 Scene Segmentation

In this experiment the depth maps from two different tabletop manipulation scenarios were segmented. First, the scene segmentation method described in Section 5.3.1 was used to extract the object, the robot’s forearm, and the table in a typical tabletop manipulation scenario as shown in Figure 5.5(a). Additionally, an experimental set-up where the object was placed on a box that, in turn, is on the table (see Figure 5.5(b)) to raise it closer to the hand was segmented. The scene segmentation method described in Section 5.3.1 was adapted to suit the additional box in the experimental set-up by extracting a vertical plane (i.e. the face of the box that looks at the Kinect sensor) using the RANSAC algorithm. A similar model of a plane  $\mathbf{m}_\pi$  to that used in Section 5.3.1 to extract the tabletop was used to obtain the parameters of the vertical plane. Note that, due to the relative angle of the plane and the sensor, the Kinect sensor detects more points from the vertical plane than the horizontal plane. Thus, the RANSAC algorithm will detect the vertical plane first if the data points representing both (i.e. the vertical



(a) Object lays on the tabletop.

(b) The object is raised.

Figure 5.6: Segmentation of a tabletop manipulation scene.

and the horizontal) planes are in the depth map used as an input. Therefore, the plane of the box is extracted from  $\Pi$  immediately after the points that are far away from the sensor are cleaned, and subsequent segmentation steps extract the tabletop and the robot's forearm from the depth map resulting from the extraction of the box.

Figure 5.6 shows the scene segmentation for the two tabletop manipulation experiments described above. The data points of the depth map obtained from the Kinect data (i.e. before any signal processing) are shown in black and the cylinder corresponding to the robot forearm is coloured in green. The horizontal plane representing the tabletop, the vertical plane corresponding to one side of the box, and the object are coloured in white, red, and blue, respectively. Considering the size of  $\mathbf{Z}_1$  in the experiments, running  $k = 22$  iterations of the RANSAC algorithm is sufficient for detecting the plane with 99 % of total probabilities of success (see Section 3.4.2). Similarly, extracting the cylinder using  $k = 52$  iterations provides 99% chance of finding the forearm. In this experiment,  $k = 300$  was used for both, the plane and the cylinder, as it provides a very low probability of missing the

Table 5.2: Standard deviation  $\sigma_1$  of the position estimation for three different objects.

	Cube	Cylinder	Prism	<b>Average</b>
$\sigma_1$ (cm.)	0.25	0.69	0.49	<b>0.48</b>

object, i.e.  $3 \times 10^{-25}$  % for the plane and  $1.9 \times 10^{-12}$  % for the cylinder. It was found that the proposed approach can accurately extract the object from the original depth map, which is a pre-requisite to continue with the process of finding the object position relative to the robot's coordinate frame.

#### 5.4.2 Finding the Shadow Robotic Hand and the Object

In this section a number of object segmentation experiments were performed on a tabletop manipulation scenario to further obtain the object position relative to the coordinate frame attached to the Kinect sensor  ${}^K\mathbf{x}$ , and the homogeneous transformation between the Kinect sensor and the Shadow robot forearm  ${}^B\mathbf{A}_K$ . This experiment aims at evaluating the accuracy of the method presented in Section 5.3 to estimate the values of  ${}^K\mathbf{x}$  and  ${}^B\mathbf{A}_K$ . The objects used in this experiment were a cube of size  $4.5 \times 3.9 \times 3.25$  cm, a cylinder with radius 2.45 cm and height 5 cm, and an octagon prism with height 4.8 cm and radius 2.45 cm. The objects were raised to be within the workspace of the robot hand using a box as shown in Figure 5.5(b).

This experiment evaluates the robustness of the proposed approach when estimating the Shadow forearm frame and the object position. Depth maps of the tabletop manipulation scene were collected using the Kinect sensor for 100 iterations for each object, and the estimate of the values of  ${}^K\mathbf{x}$ ,  ${}^B\mathbf{o}_K$ , and  ${}^B\mathbf{R}_K$  was computed for each iteration. To evaluate the accuracy of the

proposed approach in finding the object in the scene, a  $100 \times 3$  data matrix  $\mathbf{X}$  for each object was computed including all values of  ${}^K\mathbf{x}_i$  for  $i = \{1 \dots 100\}$ , and its covariance matrix  $\Sigma_{\mathbf{X}}$ . The eigenvalues  $\lambda_j$  of  $\Sigma_{\mathbf{X}}$  for  $j = \{1, 2, 3\}$  represent the variance of the object position  ${}^K\mathbf{x}$  in the directions defined by the eigenvectors. Table 5.2 shows the square root of the larger eigenvalue (i.e. standard deviation of  ${}^K\mathbf{x}$ ) for all three objects evaluated in this experiment, which on average is 0.48 cm. Assuming the object position error follows a normal distribution, one can conclude that the error is less than three times the standard deviation of  ${}^K\mathbf{x}$  with 99.7% accuracy, i.e.  $\epsilon = 3\sigma_1$ . It is worth noting that, given the size of the objects used in this experiment, the proposed approach guarantees the point  ${}^K\mathbf{x}$  is always inside the object and, therefore, the control mechanism will drive the fingers towards contact when used as input to the proposed precision grasp controller.

The error of  ${}^B\mathbf{o}_K$  was also evaluated using a similar procedure. A  $100 \times 3$  matrix  $\mathbf{O}$  was defined using the values of  ${}^B\mathbf{o}_K$  collected during all the iterations in this experiment. The covariance matrix of  $\mathbf{O}$  was computed and the square root of the larger eigenvalue was taken as standard deviation  $\sigma_o = 0.22$  cm of the forearm position. Similarly to the case when the object position was evaluated, the cylinder position error was assumed to follow a normal distribution and, therefore, 99.7% of the time the error of  ${}^B\mathbf{o}_K$  lie within three standard deviations, i.e.  $\epsilon = 3\sigma_o$ . Hence, the proposed approach can estimate  ${}^B\mathbf{o}_K$  with an error of less than 0.66 cm.

This experiment also evaluates the accuracy of the estimate of the robot forearm orientation. The azimuth  $\theta$  and elevation  $\phi$  angles of the forearm reference frame with respect to the Kinect reference frame were computed for each of the 100 trials and their average and standard deviation were obtained. The results in Table 5.3 show the average  $\mu$  and standard deviation  $\sigma$  of the

Table 5.3: Mean  $\mu$  and standard deviation  $\sigma$  (in degrees) of the robot forearm orientation relative to the Kinect reference frame.

	$\mu$	$\sigma$
$\theta$	$-178.90^\circ$	$0.07^\circ$
$\phi$	$-58.84^\circ$	$0.06^\circ$

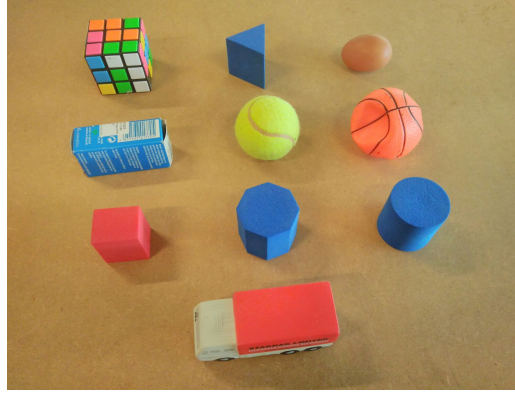


Figure 5.7: Objects used in the experiment presented in Section 5.4.3.

rotation angles ( $\theta$  and  $\phi$ ) of the forearm reference frame relative to the Kinect reference frame. The values of  $\sigma$  are  $0.07^\circ$  for the azimuth and  $0.06^\circ$  for the elevation angle and, therefore, the variations in the estimate of the robot forearm orientation relative to the Kinect coordinate frame are very small. More precisely, assuming the error of  ${}^B\mathbf{A}_K$  follows a normal distribution, one can conclude that 99.7% of the time the proposed approach can find  ${}^B\mathbf{A}_K$  with an error less than  $0.21^\circ$ , i.e.  $3\sigma$ .

### 5.4.3 Precision Grasp Control through Continuous Tactile Sensing

In this section, the method to grasp objects of unknown shape detailed in Section 5.2 was evaluated for 10 different objects. The objects are shown in Figure 5.7 and were purposely selected to have different shape, size, and surface material. The set-up used in this experiment is a tabletop manipulation scenario where the object is raised 19 cm above the table to place it in a location within the robot hand workspace such as in Figure 5.5(b). However, this assumption can be relaxed if the hand is attached to a movable arm. The proposed approach segments the scene using the RANSAC and the  $k$ -means algorithms to obtain the reference Cartesian position command for each finger's precision grasp controller. Thus, the grasp of the object was executed in two stages (i.e. pre-grasping and grasping), and then the object was lifted.

All executions of the RANSAC algorithm used in this experiment set the maximum number of iterations to  $k = 300$  although, as detailed in Section 5.3.1, a smaller number of iterations could provide sufficient certainty of finding all objects. Despite the fact that the performance of the proposed approach can be enhanced by providing a specific set of contact forces for each object (i.e. considering the object material) the same contact force configuration was used for all objects, which was selected empirically. This assumption can also be relaxed if a set of contact forces specific to the detected object material are obtained a priori. Nonetheless, the proposed approach was capable of grasping and lifting all objects used in the experiment.

Figure 5.8 shows sequences of trials using the proposed approach to grasp 4 different objects: a boiled egg, a foam prism, a tennis ball, and a foam cube.

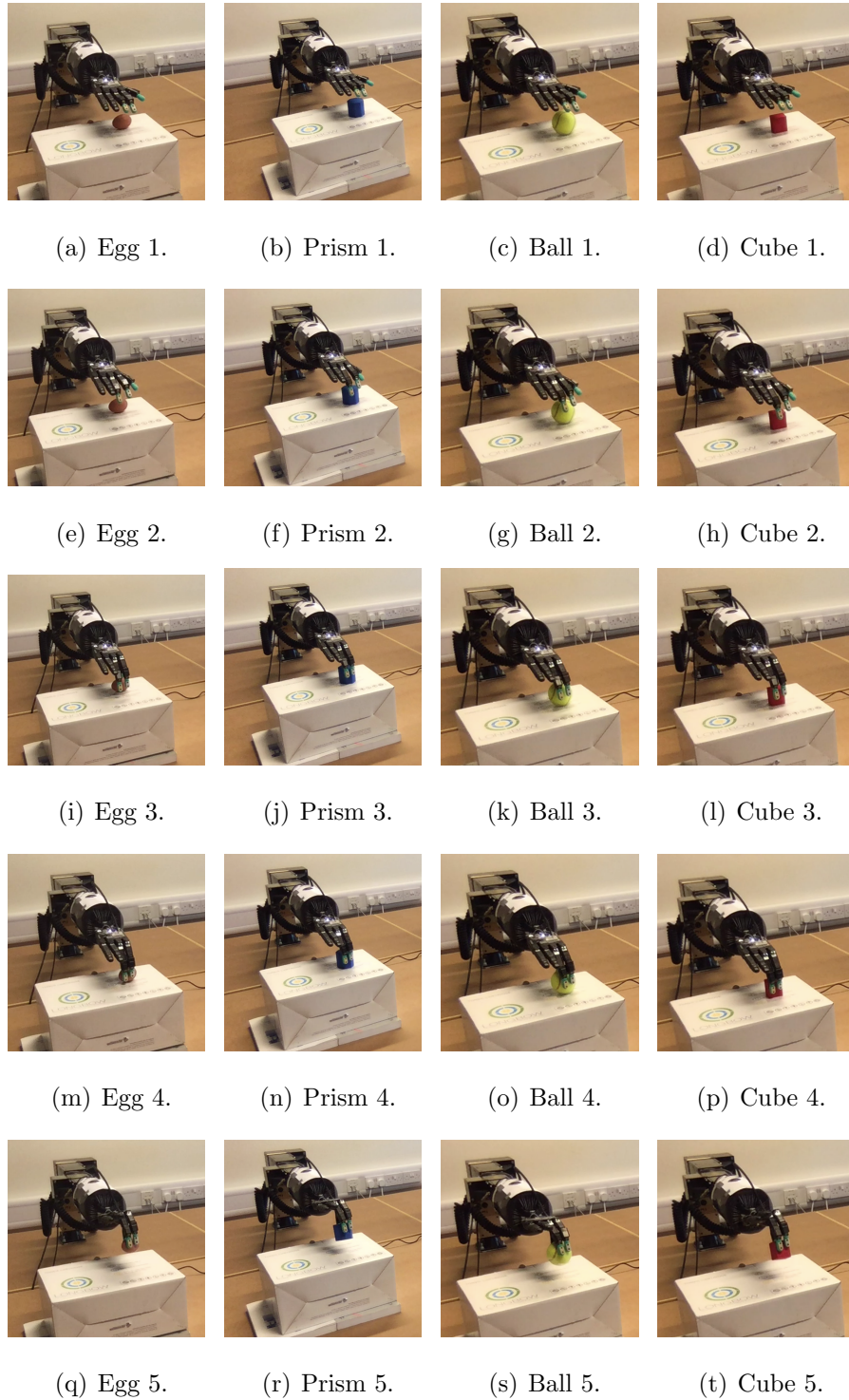


Figure 5.8: Four sequences of grasping different objects. An egg in the first column, a foam prism in the second column, a tennis ball in the third column, and a foam cube in the fourth column.



Although no information of the object shape and size was provided, the hand always grasped and lifted the objects. While the whole grasping process was performed using the approach presented in Section 5.2, the lift was performed manually by applying a torque on the wrist joint of the Shadow robotic hand using the default user interface and, therefore, varied from trial to trial. In order to support the case that the proposed controller can keep the object grasped during manipulation in spite of arising perturbations, more complex object manipulation tasks are necessary.

To illustrate the behaviour of the proposed approach the magnitude of the contact forces for each finger was analysed. Figure 5.9 shows the evolution over time of the force magnitude for the thumb, first and middle fingers during the grasp of an octagon prism as shown in the sequence of the second column in Figure 5.8. The magnitude of the thumb's reference force was set to be twice as large as the the reference forces of the first and middle fingers since it has to compensate the forces applied by the other two fingers. Thus, the reference force magnitude  $|\bar{\mathbf{f}}_i|$  for each finger ' $i$ ' is represented with a horizontal blue line in Figure 5.9, and the force magnitude at every time step  $|\mathbf{f}_i|$  is represented by a red line. The green vertical dashed line shows the approximate moment in which the object lift begins. The results show that the proposed approach adapts to keep the contact force magnitudes below the reference threshold and increase the pressure when a variation in the contact forces is detected during the lift.

There is a small chance of the proposed approach failing to grasp any of the objects used in this experiment. For that to happen, the estimation error of the object position and the robot reference frame would have to match their maximum values simultaneously, which would result in some of the Cartesian positions passed as inputs to the precision grasp controller

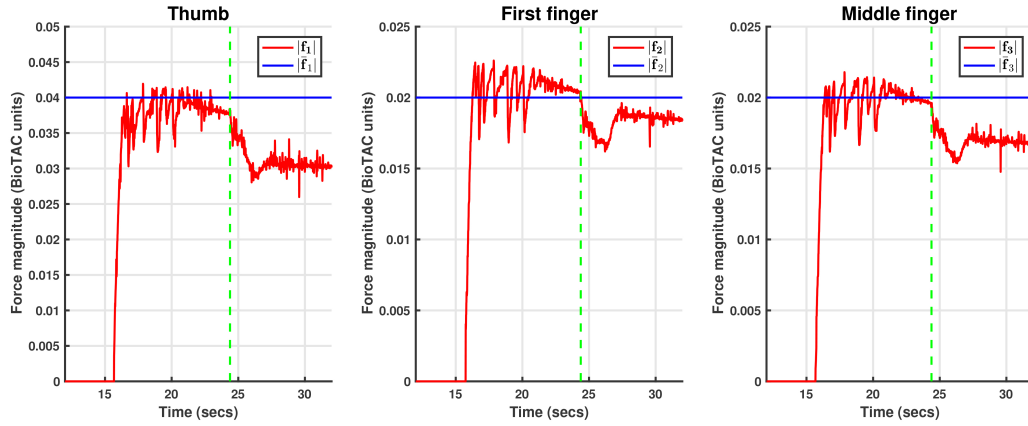


Figure 5.9: Force magnitude evolution over time for grasping and lifting an octagon prism.

being outside of the object. This is a very unlikely situation and it does not necessarily result in the object grasping failing as the object would be pushed by some of the fingers. To deal with this issue, the robot could reset the system without completing the grasp and, therefore, segment the scene again if any of the fingers reaches the input Cartesian position within a safety threshold. This approach to solve the potential issue of missing the object is grounded on the results presented in this work although, in practice, the system never encountered this situation during the experiments ran.

## 5.5 Summary

This chapter presents a control strategy to grasp objects of unknown shape. Although a number of approaches for robotic grasping exists, they often require an accurate model of the objects or they perform several exploratory procedures. This work proposes a precision grasp controller using tactile sensing that adapts the grasp configuration upon contact with the object.

The approach drives the fingers to grasp a virtual sphere inside the object and, when contact is detected, the joint positions adapt to a contact force configuration that was set *a priori* to provide a suitable grasp. Therefore, the proposed controller applies joint torques on each finger according to two inputs: the contact force vector for that finger, and a Cartesian position for that finger corresponding to the grasp of the virtual sphere. While the set of contact forces are assumed to be known (i.e. computed according to the surface material of the object), the reference Cartesian positions for each finger were obtained using a Microsoft Kinect sensor. Additionally, a method for segmenting the tabletop manipulation scenario, finding the object, and obtaining the homogeneous transformation matrix between the reference frames attached to the Kinect sensor and the Shadow robot's forearm was presented. The experimental results showed that the proposed approach could successfully grasp 10 objects with different shapes and sizes, despite the fact that no information about the object geometry was provided.

All objects tested in this experiment were successfully grasped and lifted, but a number of other objects could not be grasped due to their shape, weight, compliance or the surface material being too slippery for the given contact forces (e.g. a hammer, a jelly block or an ice cube). While the low friction of the object surface could be addressed by using an adequate contact force configuration, objects with certain shapes, weights, or compliances require alternative grasp approaches or previous knowledge of the shape of the object (i.e. to compute grasp planning). However, a large range of objects that are found in everyday scenarios can be grasped using this approach. Hence, the grasping approach presented in this chapter contributes towards the integration of robots in human environments.

In a typical robot-human scenario a common collaboration task to be

performed by the robot, after an object has been successfully grasped, is to hand the object to a human. Chapter 6 presents an innovative approach for reliable object handover between robots and humans.

## Chapter 6

# Continuous Tactile Sensing for Reliable Robot-Human Handovers

### 6.1 Introduction

The exchange of objects between humans is an everyday occurrence, although such interactions are trivial tasks for humans, they are still very challenging for robots. Existing approaches to robot-to-human object handover assume that no potential problems occur during the object transfer. However, unintentional perturbation forces can be occasionally applied to the object while it is being grasped. In that case, the robot and the object could be damaged if the robot maintains the current grasp configuration; for instance, being the object dropped. Instead of trying to maintain the current grasp, the system could move to a different configuration to reach another stable grasp; nonetheless, this would entail losing contact with the object at some point in time resulting in a high chance of the object dropping. Hence, maintaining a

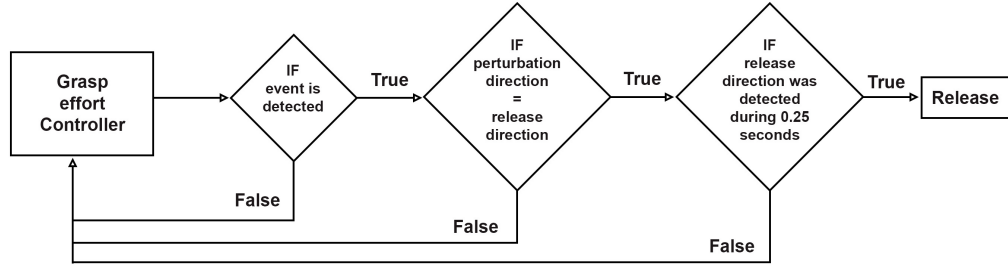


Figure 6.1: Flow chart of the proposed reliable object handover algorithm.

stable grasp could turn out not to be possible when the object is perturbed by external forces.

This chapter presents a novel approach to reliable object handover that ensures the safety of the robot and the object. A flow chart of the proposed approach is shown in Figure 6.1. Relying on tactile sensing, the system uses an effort controller to adapt the grasp forces in the presence of perturbations. Moreover, the proposed approach identifies perturbation forces being applied to the object. When a perturbation event is detected, the algorithm classifies the direction of the pulling forces to decide whether to release it or not. The reliable handover system was implemented and evaluated using a Shadow Robot hand equipped with BioTAC tactile sensors. The results show that the system correctly adapts to the forces applied on the object, and maintains the grasp, releasing the object only if the human receiver pulls in the correct direction.

The work presented in this chapter has been partially published in the following articles:

- Reliable object handover through tactile force sensing and effort control in the Shadow Robot hand. In *IEEE International Conference on Robotics and Automation* (ICRA 2017) [8].

- Towards Robot-Human Reliable Hand-over: Continuous Detection of Object Perturbation Force Direction. In 26th *IEEE International Symposium on Robot and Human Interactive Communication* (RO-MAN 2017) [9].

The rest of the chapter is organised as follows. Section 6.2 presents the grasping effort controller used to keep the object grasped while perturbations act on the object. Section 6.3 presents a release detection approach based on the findings of a preliminary study with naïve users. Both grasping effort controller and release detection approach are core functionalities implemented in the reliable handover algorithm, which is detailed in Section 6.4. Section 6.5 presents experimental results and guidelines to tune the algorithm parameters. Finally, Section 6.6 summarises the findings and concludes the chapter.

## 6.2 Grasping Effort Controller

Ensuring the robotic hand and the grasped object are not damaged during the handover process needs a system capable of adapting the hand configuration when a perturbation is applied on the object. The proposed approach relies on tactile sensing to obtain contact information used by an effort controller that keeps the grasp forces constant. The proposed approach assumes that the object being handed over is rigid, the initial configuration of the hand is ready for the handover, and a stable grasp using three fingers is set. Using a fixed position control entails contact loss or increased efforts in the joints, which could result in the robot being damaged. To illustrate the effect of forces when controlling the position of the fingers, a perturbation force was applied to a single finger of the Shadow robotic hand equipped with BioTAC

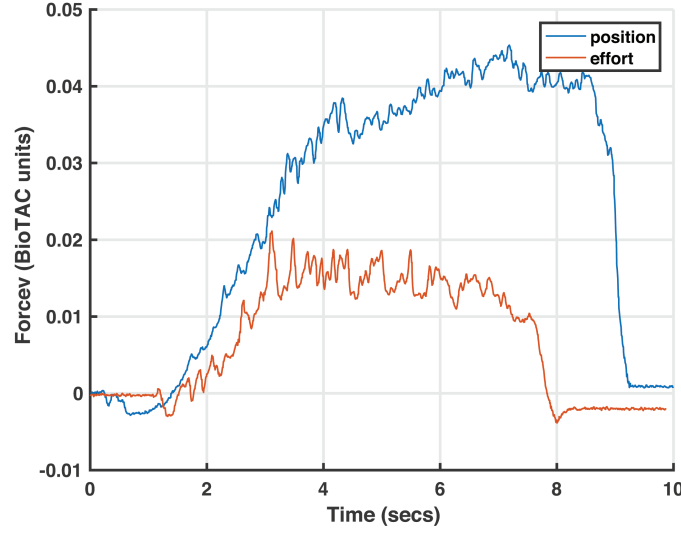


Figure 6.2: Changes in the fingertip force length for effort vs. position control.

tactile sensors. Figure 6.2 shows a comparison of the response in the norm of the force of the middle finger over time for the proposed effort controller and a position control (see Chapter 5 for how to compute the contact force).

Although the Shadow hand fingers provide some compliance through their mechanical design with tendons and springs, the force sensed for a small perturbation using a position control is more than twice the force sensed when the effort controller is running, which significantly reduces the risk of damaging the hand.

The fingers used for grasping the object will be labelled  $j = 1, 2, 3$ , where  $j = 1$  is the thumb,  $j = 2$  is the first finger, and  $j = 3$  is the middle finger, and refer to the coordinate frame defined by the forearm joint position and orientation as the base reference frame. The effort controller changes the hand configuration to maintain an initial wrench (i.e. force and torque) while adapting to perturbation forces on the object. The wrench  ${}^B\bar{\mathbf{f}}_j$  set on



the hand for the stable grasping in the robot base reference can be obtained as:

$${}^B\bar{\mathbf{f}}_j = J_j(\mathbf{q}_j)^\dagger \mathbf{\Gamma}_j, \quad (6.1)$$

where  $\mathbf{q}_j$  is the configuration of the finger joints,  $\mathbf{\Gamma}_j$  their corresponding torques,  $J_j(\mathbf{q}_j)^\dagger$  is the pseudo-inverse of the Jacobian for finger  $j$ , and the superscript  $B$  states the wrench is in the base reference frame. The necessary conditions to maintain a stable grasp are assumed to be obtained by converting the forces and torques of the initial wrench to the object reference frame and taking into account the friction coefficients, and the normals at the contact points.

In order to maintain the stability of the grasp in the presence of a perturbation while ensuring that neither the hand nor the object are damaged, one could keep the wrenches in the object reference frame  ${}^O\bar{\mathbf{f}}_j$  constant, since the palm reference suffers only small changes. Instead of maintaining the wrenches constant in the object reference frame, the contact forces and torques were kept and restored as computed in the base frame  ${}^B\bar{\mathbf{f}}_j$  for a stable grasp for each finger. The proposed effort joint control considers the fingers individually and uses the stable grasp wrench  ${}^B\bar{\mathbf{f}}_j$  as a reference while the perturbed measured wrench  ${}^B\mathbf{f}_j$  is fed back to the controller. Therefore, given the difference between the  $j$ -th finger contact wrench  ${}^B\mathbf{f}_j$  and the one for the stable grasp the effort to be applied is computed as:

$$\mathbf{\Gamma}_j = K_j J_j(\mathbf{q}_j)^T ({}^B\bar{\mathbf{f}}_j - {}^B\mathbf{f}_j), \quad (6.2)$$

where  $J_j(\mathbf{q}_j)$  is the Jacobian of finger  $j$  at the joint position  $\mathbf{q}_j$ , and  $K_j$  is a square gain matrix of size equal to the number of joints used to restore the stable grasping wrench. Thus, for instance, the finger will move backwards to keep the force constant if a perturbation increases the contact force while

maintaining the direction and torque. Generally, the product of a perturbed wrench and the Jacobian results in motion of the finger to compensate for external forces and torques. When a perturbation force is applied (since the grasped object is rigid), a change in the contact force and torques is perceived by all three fingers. All fingers will move individually to maintain the stable grasping wrench in the base frame. Experimentally it was found that this control mechanism kept a stable grasp while the object moved due to external perturbations. In this manner the proposed control mechanism implements compliance in the tactile force.

## 6.3 Release Detection

While adapting to perturbations allows to maintain the object grasp and avoid the hand to be damaged, releasing the object in a timely manner is vital to ensure the safety of the object, i.e. it should never be dropped. In order to find how humans behave when receiving an object from a robot Section 6.3.1 presents a preliminary study with naïve users. Relying on the results of that study, the proposed approach triggers the object release based on an event detection system and a method to identify the perturbation force direction. Thus, Section 6.3.2 presents a system to detect perturbations on the object during grasping. Section 6.3.3 defines the features used in Section 6.3.4 to classify the direction in which a perturbation force is applied.

### 6.3.1 Robot-Human Handovers: A Naïve User Study

A study with naïve human participants was proposed to determine when to release an object. In the experiment, the Shadow Robot hand (installed on a Schunk arm) was presenting the object (see Figure 6.3) and the naïve

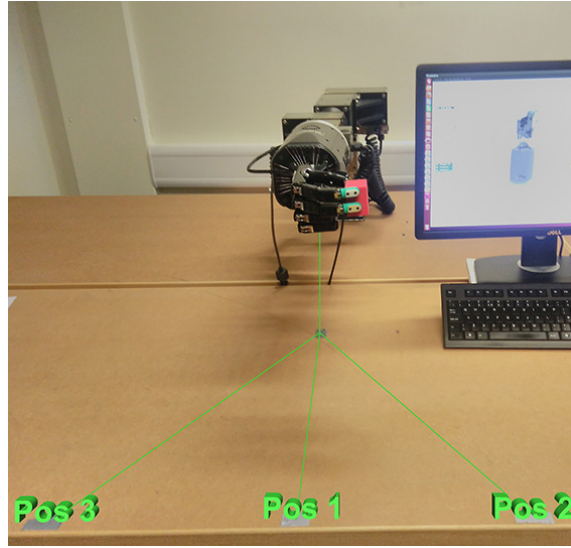


Figure 6.3: Receiver positions.

human participants were the receivers of the handover. The robotic hand was equipped with three SynTouch BioTAC tactile sensors mounted on the thumb, index and middle fingers, which were used to hold the object while the tactile sensors estimated the force applied by the fingertips. In this experiment, 10 subjects (2 females and 8 males) were asked to receive an object from the robotic hand, while keeping the Schunk arm fixed. The effort controller presented in Section 6.2 was used to adapt the grasp of the fingers preventing damage to the robot by the forces and torques applied over the object while picking it up. A cube was the object used for the handover in all experiments, grasped by the robot hand in approximately the same position and rotation for all experiments. To perform the analysis, the contact forces obtained from the BioTAC (see Chapter 5) for each finger were stored during the whole process.

For this experiment, the subjects were placed in front of the robot hand (position 1 in Figure 6.3), at a distance of approximately 35 cm from the

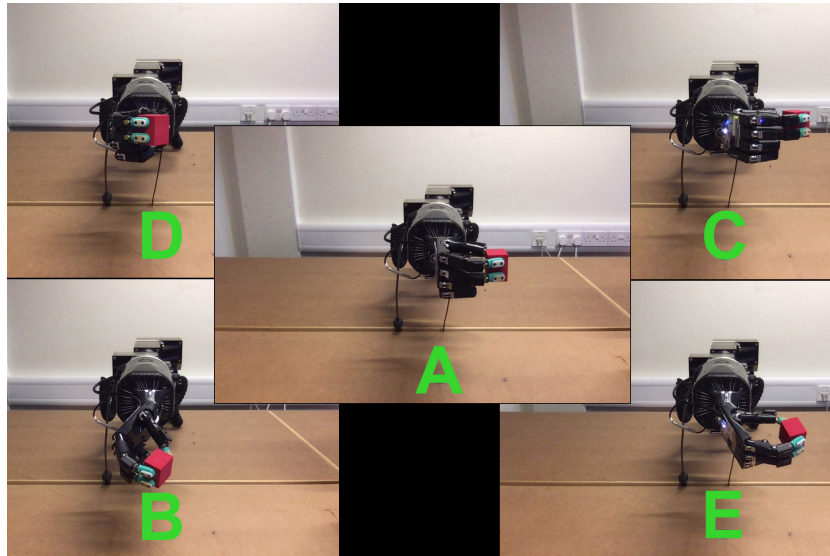


Figure 6.4: Initial configurations.

object. The subjects were explained that the purpose of the study was to evaluate robot-human handover, and they were instructed to use their dominant hand to approach the object, but no further indication of how to act during the handover was given. The goal of the first trial was to determine whether a naïve user would pull the object or expect the robot hand to release it as soon as the human grasps it. An additional dataset of a steady grasp, in which the object was grasped but no pulling force was applied in any direction, was collected as a comparison baseline. Although with different timings, all of the participants pulled the object in different directions and with variable strengths, but in some cases the subjects clearly expected the robot to release the object. The force change was quantitatively measured through the angular changes of the individual finger forces relative to the static grasping forces in the reference system of each finger-tip (more details are given in Section 6.3.3). An analysis of the forces, and a comparison with the grasp only task, clearly showed that the sum of the angular variations of

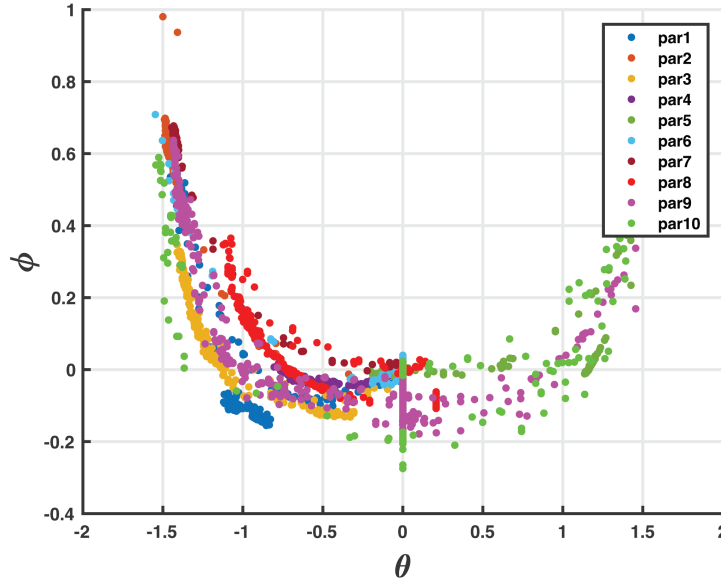


Figure 6.5: Azimuth  $\theta$  and elevation  $\phi$  angle variations for all participant first fingers.

the finger forces for all participants during pulling was significantly greater than the values during a grasp-only state. This showed that it is possible to detect the object being pulled by a naïve user before it is taken away from the robot hand.

After the initial experimental trial, the subjects were asked to perform a robot-human handover in the five different grasp configuration scenarios (A to E) shown in Figure 6.4 and three orientations relative to the robot arm,  $\pm 30^\circ$  deviation from the middle position (see Figure 6.3). A total of 150 different handover trials (i.e. 15 trials per participant) were recorded and analysed off-line. The force analysis showed that the perturbation force applied to the object was dependent on variables such as the receivers position, whether he/she was left/right handed, and the grasp configuration. Figure 6.5 shows the variations in the azimuth  $\theta$  and elevation  $\phi$  angle, of the force detected in

the first finger for all participants in this trial, i.e. see Section 6.3.3 for how to compute  $\theta$  and  $\phi$ . The plot in Figure 6.5 shows how participants pulled from the object in different directions (cf. for example participant 8 and 10), therefore showing the need for accurate pulling direction identification.

### 6.3.2 Load Force Variations for Event Detection

Based on the observations in Section 6.3.1, the proposed handover algorithm was designed to release the object based on two events: the detection of a perturbation event and the pulling force direction. The change in the perceived load force was used to identify when a perturbation has been applied on the object and trigger a classification process (see Section 6.3.4) to identify the direction of the pulling force. Therefore, at every time step, a fixed size sliding window of duration  $\Delta t$  seconds ( $\Delta t = 0.05$ ) is updated to include the latest forces  ${}^B\mathbf{f}$  measured by the BioTAC sensors. The window  ${}^B\mathcal{W}$  is divided in two equal sized sequences  ${}^B\mathcal{W}^1$  and  ${}^B\mathcal{W}^2$ , where  ${}^B\mathcal{W}^1$  denotes the oldest data and  ${}^B\mathcal{W}^2$  the most recent. The averages of the force  ${}^B\mathbf{f}$  over the windows are computed and used for detecting load force variations.

Most of the object handover approaches in the literature rely on changes in the load force to control grasping forces. Here, the estimate of the load force in the base frame  ${}^B\mathbf{f}_L$  corresponds to the sum of all the contact forces  ${}^B\mathbf{f}_L = \sum_j ({}^B\mathbf{f}_j)$ . Therefore, a change in the norm of  ${}^B\mathbf{f}_L$ ,  $\Delta {}^B\mathbf{f}_L$  can be detected when an external action is being carried out over the object. In the context of the reliable object handover algorithm, this norm change event triggers a classification process to identify the type of event as described below. When a perturbation force is applied to the object, the range in which the contact force varies is different for the  $x$ ,  $y$  and  $z$  coordinates (i.e. with respect to the BioTAC's frame). A plausible assumption is that this is caused by

the conductive fluid being displaced more easily in some directions than in others due to the design of the BioTAC, which shape is not homogeneous. Because of the ranges of  ${}^B\mathbf{f}_L$ ,  $\Delta^B\mathbf{f}_L$  is more sensitive to perturbations in some directions than in others. For instance, despite being very sensitive to lateral movement perturbations with respect to the fingertips, the change in the norm was not as responsive for frontal movements due to contact force variations having a smaller range than the lateral ones. In order to solve this issue, an approach that takes into account the scale of the contact force variations for each axis is proposed. Instead of using the norm of  ${}^B\mathbf{f}_L$ , the proposed approach computes the variation in the load forces using the Mahalanobis distance [149] between  $E[{}^B\mathcal{W}^1]$  and  $E[{}^B\mathcal{W}^2]$ :

$$\Delta^B\mathbf{f}_L = E[{}^B\mathcal{W}^*]^T {}^B\Sigma^{-1} E[{}^B\mathcal{W}^*], \quad (6.3)$$

where  $E[{}^B\mathcal{W}^*] = E[{}^B\mathcal{W}^1] - E[{}^B\mathcal{W}^2]$ ,  $E[{}^B\mathcal{W}^i]$  denotes the expected value of the corresponding sub-window  ${}^B\mathcal{W}^i$ , and  ${}^B\Sigma$  is the covariance matrix of  ${}^B\mathcal{W}$ . Therefore, Equation 6.3 provides a direction independent distance measure between two vectors i.e. contact forces. The values for  ${}^B\Sigma^{-1}$  were empirically found by computing the covariance of the load force  ${}^B\mathbf{f}_L$  during a steady state grasp of the object by the hand, i.e. without perturbation forces applied. If  $\Delta^B\mathbf{f}_L$  exceeds a fixed threshold the proposed approach determine that an external force is acting on the object. The threshold  $f_{th} = 0.002$  was experimentally chosen while perturbing the object with the grasping effort controller running. It is worth noting that selecting  $f_{th}$  without using the grasping effort control system will generate a threshold value that is too large as a consequence of the hand not adapting to perturbations, i.e. resulting in larger contact forces.

### 6.3.3 Features for force direction detection

A perturbation force  ${}^B\mathbf{f}_L$  being applied to the object can be detected if exceeding a given threshold. However, as not all perturbations are expected to result in a handover, identification of the direction of perturbation force that will trigger release if and only if it is safe to do so is also needed.

The variations of the contact force  ${}^E\mathbf{f}_j$  were estimated using the BioTAC for each individual finger  $j = 1, 2, 3$  with respect to their corresponding resting forces  ${}^E\bar{\mathbf{f}}_j$ . Therefore, the variations of the contact force  ${}^E\mathbf{f}_j$  were modelled to classify the direction of a perturbation force over the object. Given the initial grasp of the object the resting forces  ${}^E\bar{\mathbf{f}}_j$  were computed as the average response within a window of  $\Delta t$  seconds. A sliding window, which is updated at the sensor sampling interval, was used to retain all  ${}^E\mathbf{f}_j$  estimates obtained during the last  $\Delta t$  seconds. As the human touches the object, potentially starting a handover, the robot computes the perturbed forces of each finger as the average of the forces in the sliding window  ${}^E\mathbf{f}_j = E[{}^E\mathbf{f}_j(t_k)]$ . It is worth noting that both the resting and the perturbed forces are in the fingertip reference system.

The azimuth and elevation angles of the contact (perturbed) force for each finger ‘ $j$ ’ were denoted as  $\theta_j^{\mathbf{f}}$  and  $\phi_j^{\mathbf{f}}$  respectively, i.e. the spherical coordinates of the force vector in the reference system of the fingertip. Similarly,  $\theta_j^{\bar{\mathbf{f}}}$  and  $\phi_j^{\bar{\mathbf{f}}}$  are respectively the azimuth and elevation angles of the forces at the resting position. To identify the pulling force direction, a feature vector containing the differences between the azimuth  $\theta_j = \theta_j^{\mathbf{f}} - \theta_j^{\bar{\mathbf{f}}}$  and elevation  $\phi_j = \phi_j^{\mathbf{f}} - \phi_j^{\bar{\mathbf{f}}}$  angles of the contact forces and those corresponding to the resting position forces of each finger in their corresponding fingertip reference frame was defined, where  $\theta_j^{\mathbf{f}} = \arctan\left[\frac{\mathbf{f}_y^j}{\mathbf{f}_x^j}\right]$  and  $\phi_j^{\mathbf{f}} = \arctan\left[\frac{\mathbf{f}_z^j}{\sqrt{(\mathbf{f}_y^j)^2 + (\mathbf{f}_x^j)^2}}\right]$ . The resultant



feature vector  $\boldsymbol{\vartheta} = (\theta_1, \phi_1, \theta_2, \phi_2, \theta_3, \phi_3)$  is the angular deviation of the contact forces, which is an invariant descriptor against changes on the hand position, object geometry and size.

Finally, a Kalman Filter was used to estimate the angular deviation velocities of the contact forces  $\dot{\boldsymbol{\vartheta}}$ . The posterior state estimate  $\hat{\mathbf{x}}_{k|k}$  will be used as the feature vector for detecting perturbation force directions (see Section 6.3.4). A constant velocity model was used to compute the state estimate  $\hat{\mathbf{x}}_{k+1}$  at time  $k + 1$  from the true state  $\mathbf{x}_k$  at time  $k$ . The true state  $\mathbf{x}_k$  was obtained by concatenating the angular deviation of the contact forces  $\boldsymbol{\vartheta}$  and their velocities  $\dot{\boldsymbol{\vartheta}}$  at time  $k$ . Constant process and observation noise in the model were assumed and appropriate initialization for the covariance matrix of the process noise was empirically found. Before detecting any event or pulling direction readings 5 seconds of reading were collected and used to compute values of  $\boldsymbol{\vartheta}$  and their covariance. The observation noise was set by using the covariance of  $\boldsymbol{\vartheta}$  in the calibration data, which was assumed to be obtained without exerting any perturbation on the contact forces. Therefore, at each time step  $k$  the state  $\hat{\mathbf{x}}_{k|k-1}$  and covariance matrix  $\mathbf{P}_{k|k-1}$  using the latest posterior estimate  $\hat{\mathbf{x}}_{k-1|k-1}$  were predicted. Then, the posterior estimate for both the state  $\hat{\mathbf{x}}_{k|k}$  and covariance  $\mathbf{P}_{k|k}$  were updated, which will be used for estimating *a priori* parameters in the next time step  $k + 1$ .

Filtering the feature vector  $\boldsymbol{\vartheta}$  using a Kalman filter before classifying the perturbation force directions enables to overcome two issues. First, the feature vector  $\boldsymbol{\vartheta}$  is often very noisy and leads to some misclassifications [9]. More importantly, using  $\boldsymbol{\vartheta}$  as a descriptor for perturbation force directions results in occasional false positives which trigger the hand to release the grasp, resulting in the object falling [9]. By filtering the angular deviation of the contact forces abrupt changes in the trajectory estimation of the an-

gles' deviation that put the safety of the object at risk will not occur; for instance, when the object perturbation force has ended a bounce effect might happen after sudden release. Secondly, by computing the posterior state of the observation the angular deviation velocity  $\dot{\boldsymbol{\vartheta}}$  of the contact forces can be estimated, which provides additional information to enhance classification accuracy with respect to using the contact force angular deviations  $\boldsymbol{\vartheta}$  as the feature vector [9].

#### 6.3.4 Statistical learning of perturbation force directions

Object release detection is based on two events: the change in the perceived load force (see Section 6.3.2), and the issue of a pulling force by the receiver with a predetermined direction. The classification of the pulling force direction applied to the object is characterised as follows. The discrete random variable representing the different perturbation force directions was denoted as  $H$ , i.e. the  $n$  events to be identified  $\{h^1, h^2, \dots, h^n\}$ , and  $\hat{\mathbf{x}}_k$  the 12-dimensional random vector of features encoding the posterior state estimate through the Kalman filter at time step  $k$ . In this case the number of object perturbation forces is  $n = 5$ , corresponding to forward, backward, up and down pulling directions (with respect to the robot palm) and a receiver steady grasp event (i.e. no pulling). Training sets were obtained to estimate the likelihood functions of the feature vectors for each event,  $p(\hat{\mathbf{x}}|h^j)$ , which are modelled as normal distributions  $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  with mean  $\boldsymbol{\mu}$  and covariance  $\boldsymbol{\Sigma}$ . Therefore, for each event  $j$ , a normal distribution with mean  $\boldsymbol{\mu}$  and covariance  $\boldsymbol{\Sigma}$  is obtained.

Having the models of the likelihood function  $p(\hat{x}|h^j)$  for all events and given a set of prior probabilities  $p(h^j)$ , one can estimate, through the Bayes

rule, the posterior probabilities  $p(h^j|\hat{\mathbf{x}})$ , and classify input data according to the maximum *a posteriori* (MAP) probability. For simplicity, the Kalman filter's state estimate at time  $k$  will be denoted thereafter as  $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k}$ . Under the assumption of initial uninformative priors,  $p(h^j) = \frac{1}{n}$  for all pulling directions  $j$ , the current estimate is updated when the change of the load force  $\Delta^B \mathbf{f}_L$  exceeds a threshold  $f_{th} = 0.002$  (see Section 6.3.2). Therefore the feature vector  $\hat{\mathbf{x}}$  will be used to iteratively obtain new posteriors for each event, and the posterior probability  $p(h_k^j|\hat{\mathbf{x}}_k)$  at step  $k$  will be the prior for obtaining the next estimate  $p(h_{k+1}^j|\hat{\mathbf{x}}_{k+1})$ . When a significant change in the load force is first detected, the initial prior probabilities are distributed evenly among all events. The contact forces were estimated from the BioTAC data stream and the feature vector  $\hat{\mathbf{x}}_k$  was computed in windows of time length  $\Delta t$ . The posterior probabilities were updated when a significant variation in the load force is detected, using:

$$p(h_k^j|\hat{\mathbf{x}}_k) = \frac{p(\hat{\mathbf{x}}_k|h_k^j)p(h_k^j|\hat{\mathbf{x}}_{k-1})}{p(\hat{\mathbf{x}}_k|\hat{\mathbf{x}}_{k-1})}, \quad (6.4)$$

where  $p(\hat{\mathbf{x}}_k|h_k^j)$  is given by the likelihood function of perturbation force direction  $h^j$ , and the normalization constant  $p(\hat{\mathbf{x}}_k|\hat{\mathbf{x}}_{k-1})$  can be obtained as:

$$p(\hat{\mathbf{x}}_k|\hat{\mathbf{x}}_{k-1}) = \sum_i^N p(\hat{\mathbf{x}}_k|h_k^i)p(h_k^i|\hat{\mathbf{x}}_{k-1}). \quad (6.5)$$

## 6.4 The Reliable Object Handover Algorithm

The pseudocode for the proposed approach for reliable object handover is shown in Algorithm 1. In each iteration the algorithm estimates the load force variations (line 5) and, if required, updates the conditional probability  $p(h_k^j|\hat{\mathbf{x}}_k)$  for all perturbation events,  $j = 1, \dots, n$  (line 15). The object perturbation force direction which has the highest posterior probability can

be considered the one that the robot is perceiving at time step  $k$ . However, the aim of this chapter is to endow robots with a method that guarantees reliable handovers. As mentioned before, a single false positive would result in the object falling. Hence, the proposed algorithm only releases the object if a pre-set pulling direction  $h^*$  (i.e. direction in which the human is expected to pull the object) is detected during more than  $t_{th}$  seconds (lines 17-21). As the value of  $t_{th}$  is small (0.25 secs), the system is still responsive enough to release the object in a timely manner (see Section 6.5.2). The reliable object handover algorithm ensures the safety of the object by releasing only when the pulling force direction is consistent with the pre-set perturbation force direction for a period of time. As detailed in Section 6.2, the algorithm also ensures the safety of the object by maintaining the initial wrench (line 23) when perturbation forces are applied on the object.

## 6.5 Experimental Results

The algorithm presented was evaluated on a real Shadow Robot Hand attached to a Schunk arm modified to compensate for the additional weight. Figure 6.6 shows the experimental set-up. Although the arm stays fixed in all of the experiments, the set-up provides a natural handover configuration by placing the forearm parallel to the ground. The initial configuration of the fingers was manually set in the centre of the hand workspace (i.e. positions that the end-effectors can reach) and was used for all the experiments as a starting point for the initial grasping. This position allowed large finger motions without lost of contact when perturbation forces are applied to the object. From this initial approximate position, the fingers were manually adjusted to generate a stable grasp of the object. Therefore, every experiment

**Algorithm 1** Reliable object-handover algorithm

---

```

1: procedure OBJECT-HANDOVER( ${}^E\mathbf{f}_j$ ) ▷ BioTAC forces
2:    ${}^B\mathbf{f}_j \leftarrow T({}^E\mathbf{f}_j)$  ▷ Transform  ${}^E\mathbf{f}_j$  to base
3:   Update  ${}^E\mathcal{W}$  with  ${}^E\mathbf{f}_j$  ▷ Pull detect sliding window
4:   Update  ${}^B\mathcal{W}^1$  and  ${}^B\mathcal{W}^2$  with  $\sum_j {}^B\mathbf{f}_j$  ▷ Event detect sliding window
5:    $\Delta {}^B\mathbf{f}_L \leftarrow \text{Mahalanobis} [E [{}^B\mathcal{W}^2], E [{}^B\mathcal{W}^1]]$  ▷ Load force change
6:   Set empty  $\vartheta$ 
7:   for  $j = \{1, 2, 3\}$  do ▷ For each finger
8:      $\Delta {}^E\theta_j \leftarrow \arctan \left[ \frac{\bar{\mathbf{f}}_y^j}{\bar{\mathbf{f}}_x^j} \right] - \arctan \left[ \frac{\mathbf{f}_y^j}{\mathbf{f}_x^j} \right]$ 
9:      $\Delta {}^E\phi_j \leftarrow \arctan \left[ \frac{\bar{\mathbf{f}}_z^j}{\sqrt{(\bar{\mathbf{f}}_y^j)^2 + (\bar{\mathbf{f}}_x^j)^2}} \right] - \arctan \left[ \frac{\mathbf{f}_z^j}{\sqrt{(\mathbf{f}_y^j)^2 + (\mathbf{f}_x^j)^2}} \right]$ 
10:     $\vartheta \leftarrow \vartheta \cup [\Delta {}^E\theta_j, \Delta {}^E\phi_j]$  ▷ Angular changes feature vector
11:   end for
12:   Compute  $\hat{\mathbf{x}}_{k|k-1}$  and  $\mathbf{P}_{k|k-1}$  ▷ Kalman Filter Predict Step
13:   Compute  $\hat{\mathbf{x}}_k$  and  $\mathbf{P}_{k|k}$  ▷ Kalman Filter Update Step
14:   if  $\Delta {}^B\mathbf{f}_L > f_{th}$  then ▷ Event detection
15:     Update  $p(h_k^j | \hat{\mathbf{x}}_k) \forall j \in [1...n]$  using Bayes Rule
16:      $\underset{\mathbf{x}}{\text{dir}}_k = \max p(h_k^j | \hat{\mathbf{x}}_k)$ 
17:     if  $\text{dir}_k \neq \text{dir}_{k-1}$  then
18:        $t^* = t$ 
19:     else if  $t - t^* > t_{th}$  and  $\text{dir}_k = h^*$  then
20:       ReleaseObject & End
21:     end if
22:   end if
23:    $\mathbf{\Gamma}_j \leftarrow KJ(\mathbf{q}_j)^T ({}^B\bar{\mathbf{f}}_j - {}^B\mathbf{f}_j)$  ▷ Send efforts to joints
24: end procedure

```

---

had slightly different configurations of grasp forces applied on the object. Before performing the experiments the electrodes of the BioTAC were calibrated to avoid drifts of the readings due to changes in the sensor gel after a series of runs. In order to avoid damaging the robotic hand when applying a perturbation force, the grasp effort controller (see Section 6.2) was used during all the experiments. The effort controller adapts the contact forces online to the initial configuration and, thus, ensures the safety of the hand and the object. However, the effort controller interaction also makes the problem of detecting the object perturbation force direction more complex

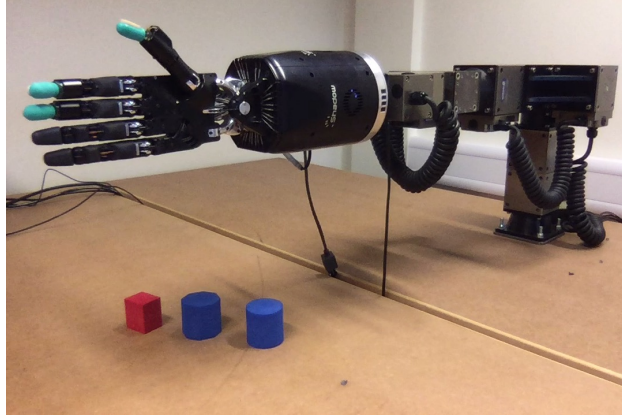


Figure 6.6: Experimental set-up.

as the controller tries to restore the forces and reduce the perturbation.

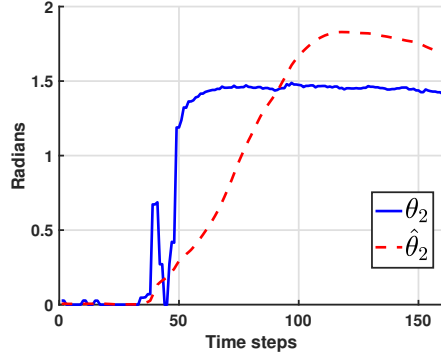
Readings for four perturbation force directions were collected, i.e. from the receiver's horizontal grasp perspective forward, backward, up and down, which were used to obtain models of their corresponding likelihood function. All training data were obtained from the initial grasp configuration using a foam cube of size  $4.5 \times 3.9 \times 3.25$  cm. The training data are stored when the variation of the load forces exceeds a threshold ( $f_{th} = 0.5$ ) as the algorithm only computes the posterior probability of the perturbation force directions when this condition is met (see Section 6.3.2). This implies that the number of training trials required varied for each perturbation class. The number of trials used to model each object perturbation force was typically between 5 and 10.

As mentioned in Section 6.3, the detection of pulling force directions relies on the variations of the contact force estimate of each finger. It was experimentally found that a sequence of perturbations is often applied to the object by the human receiver of the object instead of one unique pulling force direction. For instance, small force perturbations might be applied while the

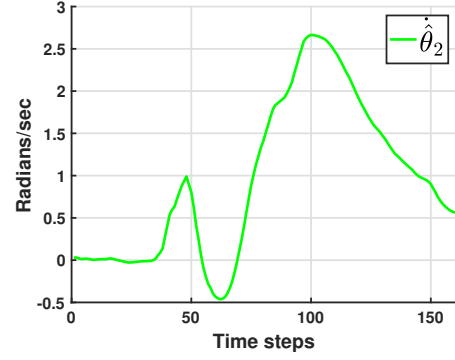
receiver grasps the object, followed by a pull action. Data of small perturbations in random directions that are generated when an object is grasped simultaneously by the robot and the human were collected and used to model the shake/jerk generated during the “joint grasp” state. Unlike training data from intentional perturbation forces, the data used for obtaining the models of the “joint grasp” state were sampled at fixed rate. This method resulted in collecting more data for each trial than in other perturbations data collection and, consequently, the number of trials used to model the “joint grasp” state was reduced to three. Although the system does not release the object when detecting a “joint grasp”, considering small perturbations in random directions was found to enhance the reliability of the handovers (see Section 6.5.2).

### 6.5.1 Force Evolution Filtering

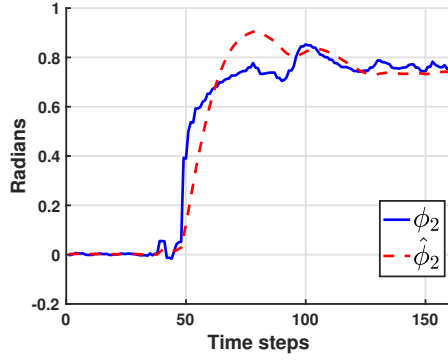
In this experiment a number of object perturbations was performed on a grasped object to understand the effect of the Kalman Filter on the force angle variations with respect to the stable grasp reference. As explained in Section 6.3.3, the true state  $\mathbf{x}_k$  is filtered to reduce the noise and to obtain additional hidden features, i.e. angle variation velocities. The experiment was evaluated on three foam objects with different geometries: a cube, a cylinder and an octagonal prism (see Figure 6.6); and the initial grasping forces were used as the reference for the controller. A perturbation force was then applied to the object and the posterior  $\hat{\mathbf{x}}$  and true  $\mathbf{x}$  force states were stored at every time step. Figure 6.7 shows the evolution of the raw and filtered angle variations, and their corresponding velocities during a downward perturbation force. Figure 6.7(a) plots the azimuth angle variations of the index finger  $\theta_2$  and Figure 6.7(c) represents the elevation angle variations  $\phi_2$



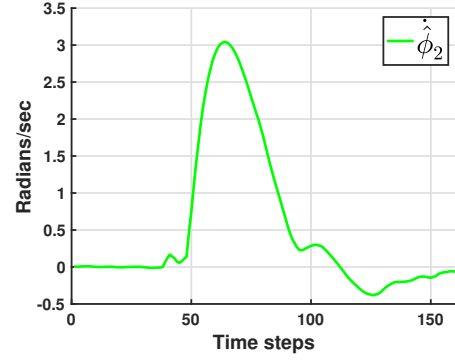
(a) Azimuth angular deviation.



(b) Azimuth angular velocity.



(c) Elevation angular deviation.



(d) Elevation angular velocity.

Figure 6.7: Index finger angle variations during downwards perturbation force.

for the same finger. The solid lines represent the angular deviations of true force state  $\mathbf{x}$  while the filtered angle trajectories,  $\hat{\theta}_2$  and  $\hat{\phi}_2$ , are represented by dashed lines. Figures 6.7(b) and 6.7(d) show the velocity hidden state of the azimuth  $\dot{\theta}_2$  and elevation  $\dot{\phi}_2$  angles respectively i.e. obtained from the posterior force state  $\hat{\mathbf{x}}$ .

It can be observed that filtering the signal reduces the noise in  $\mathbf{x}$ , smoothing the trajectory. Furthermore, it was found that smooth trajectories together with velocities simplify statistical modelling and enables each per-



turbation to be modelled as a Normal distribution (see Section 6.3.4), i.e. instead of using mixtures of Gaussians as in [8] and [9]. Hence, the presented approach eliminates the need to provide the number of Gaussians of each mixture model and allows one to deal effectively with occasional sudden changes in the object perturbation forces.

### 6.5.2 Object Perturbation Release Detection Experiments

This section presents the experimental results of the approach described in Section 6.3 for detecting object perturbation force directions. In this set of experiments a total of 48 trials were collected from four different object perturbation directions and 12 additional trials of “joint grasps” (i.e. small perturbations in random directions) using three objects: a cube, an octagonal prism and a cylinder. For all data collection, the initial grasp configurations were manually adjusted to generate an initial stable grasp approximately similar in all trials. This led to slightly different grasp configurations and applied forces over the objects. The variations in initial grasp configurations are not the only differences between trials, since the object pulling forces were not controlled and the output of the grasp effort controller varies across trials. Figure 6.8 shows the classification estimate for one second long perturbation forces in forward and downward directions. It was found that the perturbation forces very often consist of a sequence of two or more force perturbations over the object. For instance, both examples depicted in Figure 6.8 detect a “joint grasp” at the beginning of the perturbation since the pulling force is not significant enough yet. Then, although the initial estimate might be incorrect, the system always estimates the correct perturbation direction when sufficient evidence is presented, proving that the system can generalise to un-

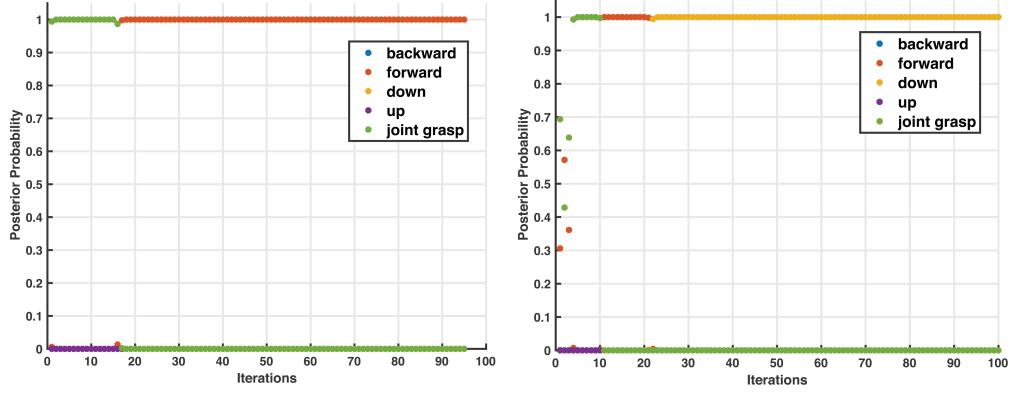


Figure 6.8: Recursive estimation of perturbation force directions.

seen objects as the cylinder or the octagon prism were not used for training. Moreover, it can be observed that the largest estimated probability remains stable after the correct perturbation force direction has arisen.

As explained in Section 6.4, the reliable object handover algorithm detects that the receiver is pulling the object in a certain direction when its estimate is higher than any other direction estimate during  $t_{th}$  seconds. Therefore, the value of  $t_{th}$  establishes a trade-off between system responsiveness and classification accuracy. The aim is to find a value of  $t_{th}$  that guarantees that the system will perform reliable handovers. This is discussed in the next sections.

### 6.5.2.1 Response vs. Accuracy

In this experiment the time required for successful identification of the correct perturbation was computed for different values of  $t_{th}$  seconds using the 48 trials from four different perturbation force directions detailed above. The time needed to detect the “joint grasp” state is not considered as the proposed algorithm only computes a new estimate of perturbation force directions if a significant change on the load force is detected. This makes it impossible

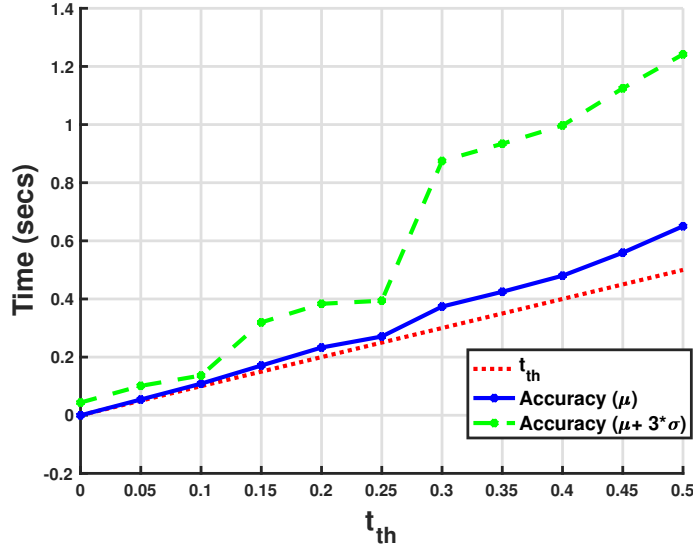


Figure 6.9: Identification speed for different values of  $t_{th}$ .

to replicate similar conditions with the whole algorithm running as the data used for modelling a “joint grasp” were collected without using the event detection system. Furthermore, considering that the handover should never be completed during the detection of a “joint grasp” state, the time needed to detect it would not be representative of the system responsiveness for object release.

Figure 6.9 shows the average time needed for successful identification  $\mu$  and standard deviation ( $\sigma$ ) for different values of  $t_{th}$  since the perturbation force is first applied on each trial used in this experiment. It is worth noting that there is a lower bound of identification time as perturbations are detected when the estimate is consistent during a period longer than  $t_{th}$  seconds. Therefore, the minimum time that detecting the correct perturbation can take is  $t_{th}$  seconds and it is represented by a red dotted line (see Figure 6.9). The solid line represents the average time ( $\mu$ ) needed for successful

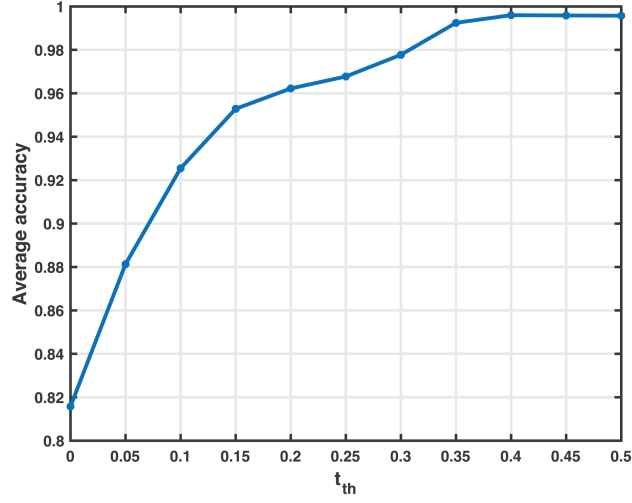


Figure 6.10: Identification accuracy for different values of  $t_{th}$ .

identification while  $\mu + 3\sigma$  is represented by the dashed line, showing the range in which successful object perturbation force detection is performed 99.7% of the times. Despite the average detection time growing progressively with increasing  $t_{th}$ , the standard deviation significantly increases for values greater than 0.25. It can be concluded that, although the value of  $t_{th}$  should remain as small as possible, a good response is achieved for values smaller or equal to 0.25, for which the detection seldom takes longer than 0.4 seconds.

Additionally, the average classification accuracy of the proposed approach for different values of  $t_{th}$  is analysed using the same 48 trials from four different object perturbation force directions used in the rest of the experiments of this section. This experiment considers the 12 additional trials of “joint grasps” discussed in Section 6.5.2 as their misclassification could result in unexpected object release. Figure 6.10 shows the average accuracy as a function of  $t_{th}$ . An 81.57% identification accuracy is obtained when only one estimate is considered (i.e.  $t_{th} = 0$ ) and increases along with larger values

Table 6.1: Confusion matrix of perturbation force directions for  $t_{th} = 0.15$ .

	Back	Front	Down	Up	Joint grasp
Back	88.54 %	0.00 %	0.00 %	0.00 %	11.46 %
Front	1.75 %	92.98 %	0.00 %	0.00 %	5.26 %
Down	0.00 %	0.00 %	97.49 %	2.51 %	0.00 %
Up	0.00 %	0.00 %	0.00 %	97.41 %	2.59 %
Joint grasp	0.00 %	0.00 %	0.00 %	0.00 %	100.0 %

of  $t_{th}$ . In order to choose an appropriate value for  $t_{th}$ , one could select the value such that greater values do not significantly increase the classification accuracy. According to that criterion, a 95.29% identification accuracy was obtained when selecting  $t_{th} = 0.15$ .

### 6.5.2.2 Tuning for Reliable Handovers

In practice a reliable handover system should never release the object when it is not supposed to. Although the results of the previous experiment show that  $t_{th} = 0.15$  provides the best trade-off between system responsiveness and classification accuracy, the aim is also to keep the rate of false positives, that might trigger the object release, as low as possible. Table 6.1 shows the confusion matrix of perturbation force directions for  $t_{th} = 0.15$  which, as mentioned above, provided the best trade-off between system responsiveness and classification accuracy. Despite the average classification accuracy being 95.29%, one can understand that trials misclassified as “joint grasp” will not make the system fail as the object will not be released on this state, i.e. yellow cells on the table. However, it is fair to say that it makes the system less responsive, as detecting the “joint grasp” state while a perturbation force is

Table 6.2: Confusion matrix of perturbation force directions for  $t_{th} = 0.25$ .

	Back	Front	Down	Up	Joint grasp
Back	85.71 %	0.00 %	0.00 %	0.00 %	14.29 %
Front	0.00 %	100.0 %	0.00 %	0.00 %	0.00 %
Down	0.00 %	0.00 %	100.0 %	0.00 %	0.00 %
Up	0.00 %	0.00 %	0.00 %	98.16 %	1.84 %
Joint grasp	0.00 %	0.00 %	0.00 %	0.00 %	100.0 %

being applied in the direction set for object release will delay the completion of the handover.

In Table 6.1 the colour of the cells of true positives is green. Red cells are false positives that may result in unexpected release. Therefore, the system will erroneously release the object 1% of the time when  $t_{th} = 0.15$ . Nevertheless, this limitation can be overcome by increasing  $t_{th}$  at the cost of reducing the system responsiveness. As discussed in Section 6.5.2.1, the system response is not significantly decreased for values of  $t_{th} \leq 0.25$  secs. The confusion matrices for different values of  $t_{th}$  were computed and it was found that the number of false positives that might result in unexpected object releases was zero for values greater than or equal to 0.25 seconds. Table 6.2 shows the confusion matrix for  $t_{th} = 0.25$  in which only the values belonging to the diagonal or the “joint grasp” are different from zero. It is worth noting that  $t_{th} = 0.25$  maintains good system responsiveness as the average time needed for successful identification is 0.27 seconds (see Figure 6.9). Therefore,  $t_{th} = 0.25$  provides an adequate balance between average accuracy (96.77%), average responsiveness (0.27 secs) and system reliability (0% unexpected object releases).

### 6.5.3 Force Adaptation and Object Handover

This section presents experimental results of the reliable handover algorithm when a sequence of perturbation forces with different directions is applied to the object. Specifically, force direction detection experiments were performed along with the effort adaptation controller for sequences of two combined perturbations. Hence, the response of the approach was tested using a variety of consecutive events including, at the end, a perturbation force that was pre-set to be the direction that completes the handover, i.e. the hand has to open the fingers releasing the object. The algorithm was evaluated for every combination of perturbing forces such as two opposite object rotations, vertical forces, pushing forward and pulling backwards.

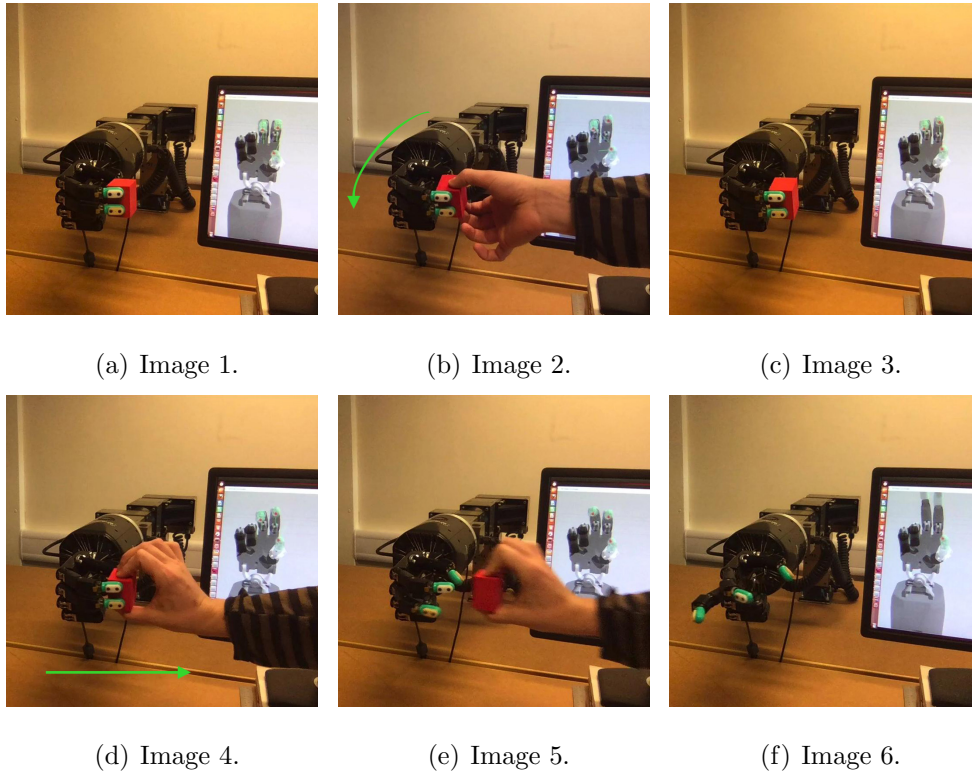


Figure 6.11: Object rotation and pulling event sequence.

Figure 6.11 shows the sequence of one of the trials, where the object is rotated in a counter-clockwise direction and then pulled backwards from the robot hand, i.e. triggering the object release. Although the fingers adapt individually and the controller has no information on the geometry of the object, the system kept contact with the object and maintained a stable grasp when the object was rotated by the human (see Figures 6.11(b) and 6.11(c)). Then, the object was released when a perturbation force backwards was detected as shown in Figures 6.11(d) to 6.11(f).

Figure 6.12 shows the evolution of the components of the forces over time in their corresponding end-effector frame for the above experiment. The first two solid vertical lines (i.e. left column) in the time sequence represent the start and end of the object rotation and were set using the event detection system (see Section 6.3.2). The third vertical line (i.e. right column) sets the beginning of the pulling force perturbation while the final vertical line signals the pulling force and consequent object handover. During a perturbation force, the contact forces deviate from the initial configuration and the fingers change their position while trying to keep the difference with the reference forces as small as possible. When the perturbation force ends the controller keeps trying to restore the reference contact forces. However, the hand could not generate the exact same forces since the configuration of the fingers changed and the robot did not have enough degrees of freedom to compensate for these variations. This happens in many trials because each finger has only three joints and therefore the grasping configurations were not manipulable. Nonetheless, the controller maintained the forces close to the initial reference and, although the configuration of the hand changed after the first perturbation, the proposed approach successfully detected the pulling event and released the object.



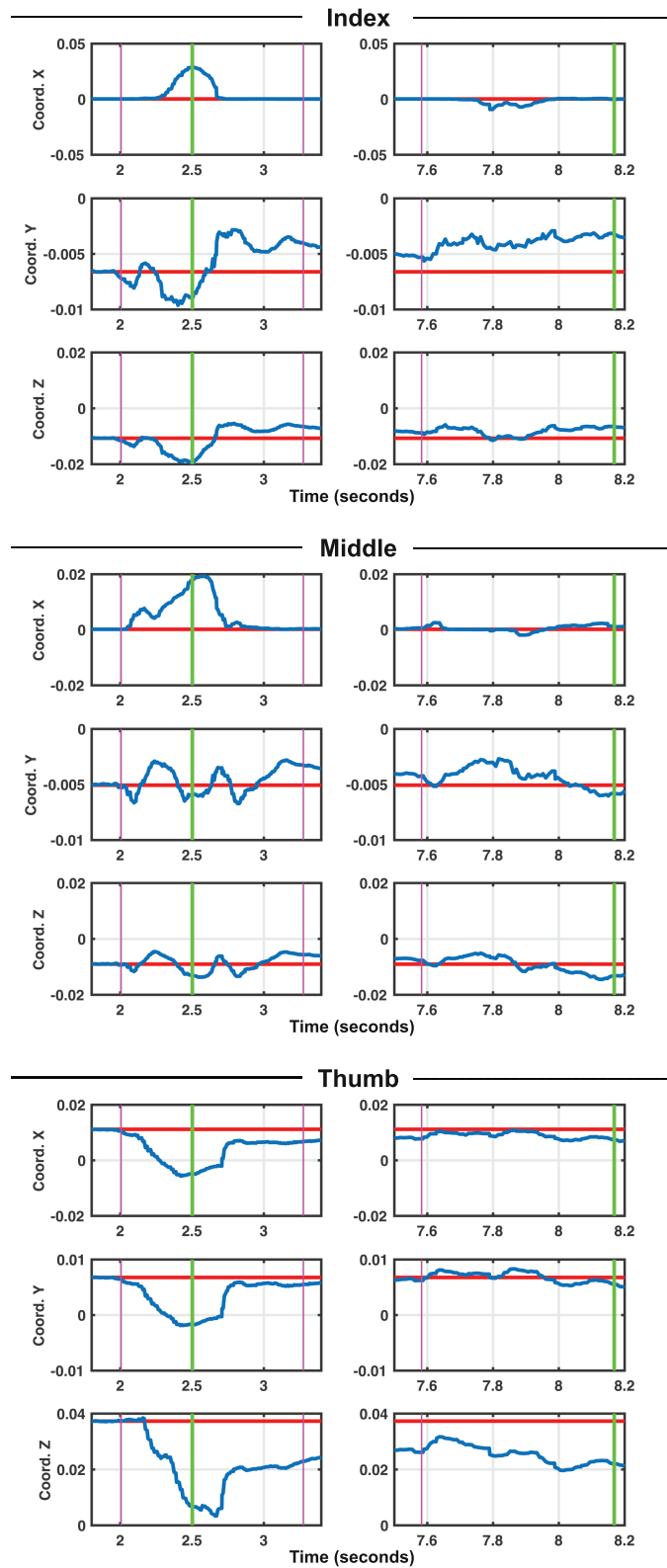
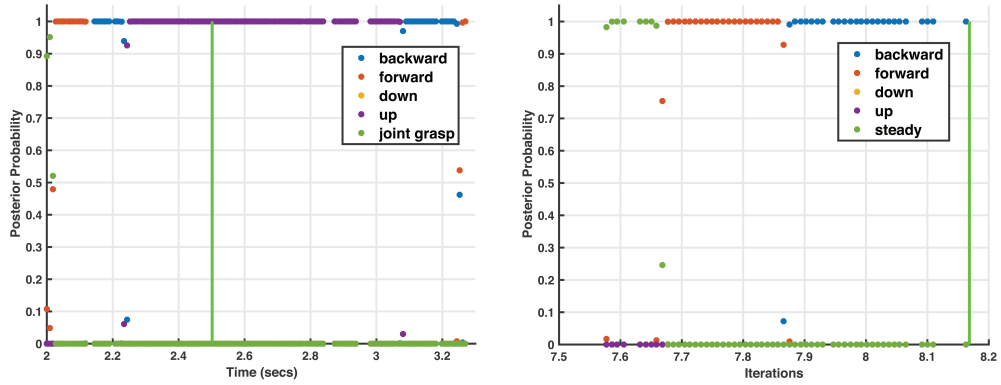


Figure 6.12: Forces response against object rotation (left column) and pulling (right column) events for each finger.



(a) Counterclockwise roll perturbation force detection. (b) Detection of backwards pull perturbation force triggers the object release.

Figure 6.13: Perturbation on object classification of a sequence of events.

The performance of the continuous detection of the perturbation force directions was tested in the current experiment. Figure 6.13 plots the results of the classification for the time periods in which the object perturbations are applied which corresponds to the time periods of Figure 6.12 from 2 to 3.3 seconds, and 5.5 to 8.2 seconds. In Figure 6.13, the horizontal axis is the time while the vertical axis is the probability of each direction. A solid vertical line shows the exact moment when the correct force perturbation is detected i.e. consistent estimation during more than 0.25 secs. It is worth noting that the algorithm only computes the posterior estimates of the perturbation events when a change in the load force is detected (see Section 6.3.2) and, therefore, the classification procedure in Figure 6.13 is not performed at a fixed rate. Although the object rotations were not included in the training sets, the system was able to correctly classify an object perturbation force being applied upwards when rotating the object counter-clockwise. Intermediate object perturbation force directions were detected for a short period of time, prior to the detection of the correct one. Moreover, Figure 6.13(a) shows a

sequence of posterior estimates when the object was released from the roll rotation. Interestingly, this sequence of estimates is an inversion of the one observed prior to the detection of a perturbation upwards. However, none of these sequences of estimates kept a consistent estimate for longer than  $t_{th} = 0.25$  seconds and, thus, they did not result in an object release. Figures 6.13(a) and 6.13(b) show that only the probability of the correct force direction is consistently estimated sufficiently long enough to be detected and, if required, triggered the object release (see Figure 6.13(b)).

## 6.6 Summary

This chapter presents an algorithm to perform reliable robot to human object handovers, which has been implemented using a Shadow Robot Hand equipped with BioTAC tactile sensors. Current, state-of-the-art approaches assume the handover is going to take place with no potential problem. However this is not always the case as external forces could be applied on the object during the handover process, resulting in the object falling or the robot hand being damaged. To solve these two issues, the proposed approach adapts the grasping with respect to perturbation forces over the object and releases the object only when the receiver pulls the object in a pre-set direction. Relying on tactile sensing, the proposed algorithm combines effort joint control, event detection, and identification of object perturbation force directions in order to perform reliable handovers.



## Chapter 7

# Conclusions and Future Work

The combination of various stimulus modalities (i.e. visual, hearing, tactile) is crucial for humans to build an accurate representation of their surroundings and act under uncertainty or incomplete information. While traditionally robots have perceived their environment through contactless sensing (e.g. proximity, visual), the applications of tactile sensing in robotics have recently received increasing research interest. Chapter 2 presented a literature review of tactile sensing and its applications in robotics and identified three shortcomings that were subsequently addressed in this thesis:

- Robotic material identification of object surfaces being approached only in terms of an episodic process;
- Robotic grasping approaches that do not implement active exploration methods being restricted to objects with known geometry;
- Robot-Human handover systems not considering unexpected perturbations during the object transfer.

Chapter 3 reviewed the techniques employed to address these limitations. Machine learning, robotics and control systems, and signal processing meth-

ods were exploited to endow robots with innovative tactile sensing capabilities. Additionally, Chapter 3 described the robotic platform used in this thesis for the implementation of the proposed grasping and object handover approaches.

Chapter 4 presented a technique for surface material identification that approached tactile sensing as a continuous process, combining the vibration signals generated when contacting a surface with thermal information. The proposed approach was proven to outperform state-of-the-art material identification approaches when the same information was provided.

Chapter 5 detailed a model of the contact forces using the BioTAC tactile sensor. The proposed model was used to design a precision grasp controller that allows a Shadow robotic hand to grasp objects of unknown geometry by adapting the fingertips towards a given set of contact forces.

Chapter 6 designed an innovative system to hand over objects between a robot and a human while preventing the robot from damage if force perturbations occur on the object. The proposed approach adapts the grasp using an effort controller to keep the contact forces constant and release the object only when the human can reliably retain the object. The system thus avoids damage to robot or object.

## 7.1 Summary of Thesis Contributions

The primary focus of this thesis was to explore tactile sensing to endow robots with the ability to collaborate with humans in object manipulation tasks. To fulfil this objective, the work presented in Chapters 4, 5, and 6 contributes towards enhancing robotic capabilities on material identification, robotic grasping, and robot-human handovers. Sections 7.1.1, 7.1.2, and 7.1.3

summarise the contributions of this thesis.

### 7.1.1 Continuous Material Identification through Tactile Sensing

Object material identification provides information relating to contact properties such as its friction coefficient or compliance, which are significant to endow robots with robust manipulation of objects. The continuous material identification approach presented in Chapter 4 performs a sequence of evaluations during the same exploratory movement, updating the estimate of all trained materials using the latest tactile information. The contribution of the proposed material identification approach is three-fold:

*Continuous material identification.* The proposed approach allows fast and very accurate material identification using vibration signals, taking into account the sequential nature of tactile sensing.

*Multimodal continuous material identification.* The vibration-only approach can be easily extended to other sensing modalities. Thus, including temperature information was found to significantly reduce the time needed to identify the material correctly.

*Benchmarking with state-of-the-art.* A comparative study with tactile material identification techniques (i.e.  $k$ NN, ANN and SVM) and a set of descriptors taken from the literature to characterise the vibration signal was performed. The experimental results showed that the proposed multimodal continuous material identification approach outperformed the other methods when the same information was used.

### 7.1.2 Robotic Grasping of Unknown Shaped Objects

Robotic grasping is a core skill to allow robots to manipulate objects as the majority of tasks requires, at some point, that an object is held in the hand. The robotic grasping system presented in Chapter 5 provides continuous adaptation of the fingers upon contact, to reach a suitable contact force configuration. The contribution of this method is three-fold:

*Contact forces model of the BioTAC.* Although the BioTAC sensor is not a force sensor, a model of the contact forces was developed to allow a Shadow robotic hand to be controlled. The contact forces at the fingertip were modelled using the impedance and pressure signals of the BioTAC tactile sensor.

*“Shadow Robot - Kinect sensor” homogeneous transformation.* A technique to compute the homogeneous transformation between a Shadow Robotic hand and the Microsoft Kinect sensor was presented. The homogeneous transformation enables computation of the coordinates of an object (i.e. detected using a Kinect sensor) relative to the robot and, therefore, sends the Cartesian position commands to the proposed precision grasp controller.

*Precision grasp controller.* A precision grasp controller was implemented in a Shadow Robotic hand equipped with BioTAC sensors in the thumb, first, and middle fingers. The proposed robotic grasping technique combined Cartesian position and contact force control using the proposed contact model to drive the fingers towards the object and adapt the grasp configuration upon contact. The proposed approach was evaluated in a tabletop scenario and experimental results proved the system was capable of grasping a variety of objects with different shapes, size, and contact properties.



### 7.1.3 Robot-Human Object Handovers through Continuous Tactile Sensing

One of the most common object manipulation tasks that humans perform when collaborating with each other is object handover. It is a difficult process that needs to consider setbacks during its execution such as human shaking or collisions with other objects that can result in the object, the robot, or both being damaged. Moreover, the release of the object was designed to be triggered only if the receiver is holding the object in a secure manner. The contribution of the approach developed in this thesis is three-fold:

*Grasping effort controller.* The proposed approach for handing over objects between a robot and a human used a grasping effort controller that adjusts the grasp to keep the contact forces constant and, thus, continuously adapt to perturbations on the object.

*Continuous classification of perturbation force directions.* When a perturbation event was detected, a continuous classification of the direction in which the receiver pulled was computed using tactile information and, thus, the reliable handover algorithm decided whether to release the object or not.

*Reliable handover algorithm.* When the proposed approach perceived a perturbation being applied on the object, it triggered the continuous classification of perturbation force directions. If the human receiver pulled in the correct direction during 250 ms the robot released the object, as it could be confident that the object was secured by the human. This technique reduces the computational cost of running the continuous classification of perturbation force directions and provides reliability against small-scale variations on the contact forces. Experimental results on a Shadow Robot hand equipped with BioTAC sensors shows that the system correctly adapted to the forces

applied on the object, maintained the grasp and released the object only if the human receiver could reliably hold the object.

## 7.2 Future Work

The contributions of this thesis entail significant improvements for robotic tactile perception and manipulation of objects. However, endowing robots with the required skills to collaborate with humans in a reliable manner involves developments in many other areas. This section outlines a number of directions in which the work can be extended in order to progress towards autonomous robots collaborating in human environments.

Although the material identification approach presented in Chapter 4 was proven to be more accurate than other state-of-the-art approaches when the same information was provided, its performance in a real robot has not been evaluated in this thesis. To obtain relevant data for classification using the recursive material estimation algorithm, the robot should be able to keep the contact forces of the finger constant during the exploration of different shaped surfaces. An interesting approach to perform this exploration could combine the precision grasp controller presented in Chapter 5 to drive the fingers upon contact commanding small magnitude contact forces, with compliant motion control of one finger to perform small explorations on the surface, as in [103], until the highest material estimate does not change any more.

This thesis also contributes to material identification by defining new descriptors using the heat transfer between the tactile sensor and the object surface material. Although the proposed descriptors enhance recognition speed and accuracy when combined with tactile micro-vibrations using the continuous material identification approach, they are not suitable for

thermal-only material identification. The definition of additional descriptors could provide the necessary information to identify materials by their thermal properties and enhance multimodal material identification. Moreover, the proposed recursive Bayesian estimation framework enables an easy integration with other sensing modalities. For instance, kinaesthetic sensing information obtained during the surface exploration could be integrated and material identification speed and accuracy could be enhanced by generating visual texture based priors.

While the proposed precision grasp controller was proven to grasp a number of objects made with different materials, more sophisticated grasping could be achieved if providing adequate contact force configuration for the material of which the object is made. Analysis of the contact properties such as friction coefficients or compliance of all materials trained in the material identification approach could be used to find the optimal contact force configuration to command the precision grasp controller. Hence, the proposed approach could be used to grasp highly delicate objects without breaking them.

The contact force model developed in this thesis can be exploited for purposes other than object grasping and handover. Future research will explore different applications of the contact force model such as, for instance, object manipulation, slip detection or object recognition. Moreover, the proposed method to find the homogeneous transformation between a Shadow Robot hand and a Microsoft Kinect sensor allows the integration of 3D vision for dexterous manipulation tasks. However, a scenario with more than one Shadow Robot hand has not been evaluated in this thesis. Future steps could include a second Shadow Robot hand (i.e. left handed) to take advantage of 3D vision and tactile sensing when performing dual arm manipulation tasks.

The proposed approach to grasp unknown shaped objects could be easily generalised to other experimental scenarios by addressing the motion of the robotic arm to which the robot is attached. Therefore, moving the Schunk arm so the palm of the Shadow hand is facing the object at a certain distance would significantly increase the workspace of the robot. In addition, the precision grasp controller could be used in teleoperated robots. A number of input devices such motion capture, data gloves, or brain-computer interfaces can be used to capture the movements of the hand and the arm of a human operator. As the proposed precision grasp controller adjusts the pressure applied to the object, the robot could be teleoperated by a human to perform precision grasps without the risk of breaking the object or the robot.

Future research will also further investigate the human-robot interaction protocol to predict the direction in which the receiver is going to pull the object during the handover. While this work provides insights into how humans receivers pull in different directions during object handover, the factors that determine the direction need to be explored in depth. Thus, visual information could be used to analyse the conditions in which the handover is going to take place (e.g receivers height, pose or approaching hand) and predict the pulling direction that should trigger the object release. Moreover, using the precision grasp controller presented in Chapter 5 together with a vision system for detection of grasped objects could be used to develop a system that allow robots to act as a receiver in human-robot and robot-robot handover scenario. The collection of data used to train the recursive identification of perturbation force directions system could be speeded up significantly by controlling the motion and forces applied on the object when the receiver pulls from it.

Comprehensive analysis of the robot arm kinematics could be used to

extend the proposed robot-human handover approach to a full arm/hand system. The integration of kinaesthetic information in the proposed effort controller could allow the robot to adapt to perturbations applied on the robot and the object. Finally, collaboration with researchers on cognitive science and physiology could extend the understanding of how humans perform handovers and the social implications during human-robot interaction. These findings could contribute to the development of more advanced systems that empower robotics to collaborate with humans in daily tasks.

## 7.3 Conclusion

The aim of this thesis was to use tactile sensing to improve robot-human collaboration capabilities in object manipulation tasks. The research described in this dissertation addresses the objective by presenting contributions in the areas of material identification, unknown shaped object grasping and Robot-Human object handover. Approaching tactile sensing as a continuous process was a cross-sectional concept of this thesis, which was explored to enhance robotic perception and adaptation upon contact. This thesis is a step towards the development of autonomous robots that collaborate with humans in every day scenarios.



## Bibliography

- [1] K. Noda, K. Hoshino, K. Matsumoto, and I. Shimoyama. A shear stress sensor for tactile sensing with the piezoresistive cantilever standing in elastic material. *Sensors and Actuators A: physical*, 127(2):295–301, 2006.
- [2] J. Dargahi. A piezoelectric tactile sensor with three sensing elements for robotic, endoscopic and prosthetic applications. *Sensors and Actuators A: Physical*, 80(1):23–30, 2000.
- [3] T. Zhang, H. Liu, L. Jiang, S. Fan, and J. Yang. Development of a flexible 3-d tactile sensor system for anthropomorphic artificial hand. *IEEE Sensors Journal*, 13(2):510–518, 2013.
- [4] J. Fishel, G. Lin, and G.E. Loeb. SynTouch LLC BioTAC product manual, v. 16. Technical report, 2013.
- [5] H. Elias. *EtherCAT Dual CAN Hand User Manual*. Shadow Robot Company, v0.9 edition.
- [6] A. Gómez Eguíluz, I. Rañó, S. Coleman, and T.M. McGinnity. Continuous material identification through tactile sensing. In *IEEE World Congress on Computational Intelligence (IEEE WCCI)*, 2016. proceeding in International Joint Conference on Neural Networks (IJCNN).

- [7] A. Gómez Eguíluz, I. Rañó, S. Coleman, and T.M. McGinnity. A multi-modal approach to continuous material identification through tactile sensing. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.
- [8] A. Gómez Eguíluz, I. Rañó, S. Coleman, and T.M. McGinnity. Reliable object handover through tactile force sensing and effort control in the shadow robot hand. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.
- [9] A. Gómez Eguíluz, I. Rañó, S. Coleman, and T.M. McGinnity. Towards robot-human reliable hand-over: Continuous detection of object perturbation force direction. In *26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, August 2017.
- [10] A. Gómez Eguíluz, I. Rañó, S. Coleman, and T.M. McGinnity. Multi-modal material identification through recursive tactile sensing. *Robotics and Autonomous Systems*, 2018.
- [11] A. Gómez Eguíluz, I. Rañó, S. Coleman, and T.M. McGinnity. Reliable robotic handovers through tactile sensing. *Under review in Autonomous Robots*. Submission id: AURO-D-17-00174.
- [12] D.L. Schacter. *Psychology second edition*, 41 Madison Avenue, New York, NY 10010, 2011.
- [13] R.S. Johansson and Å.B. Vallbo. Tactile sensibility in the human hand: relative and absolute densities of four types of mechanoreceptive units in glabrous skin. *The Journal of physiology*, 286(1):283–300, 1979.



- 
- [14] E.R. Kandel, J.H. Schwartz, T.M. Jessell, S.A. Siegelbaum, and A.J. Hudspeth. *Principles of neural science*, volume 4. McGraw-hill New York, 2000.
  - [15] H.B. Helbig and M.O. Ernst. Optimal integration of shape information from vision and touch. *Experimental Brain Research*, 179(4):595–606, 2007.
  - [16] M.O. Ernst and M.S. Banks. Humans integrate visual and haptic information in a statistically optimal fashion. *Nature*, 415(6870):429–433, 2002.
  - [17] A. Amedi, K. von Kriegstein, N.M. van Atteveldt, M.S. Beauchamp, and M.J. Naumer. Functional imaging of human crossmodal identification and object recognition. *Experimental Brain Research*, 166(3-4):559–571, 2005.
  - [18] T. Kassuba, C. Klinge, C. Hölig, B. Röder, and H.R. Siebner. Vision holds a greater share in visuo-haptic object recognition than touch. *Neuroimage*, 65:59–68, 2013.
  - [19] E. Baumgartner, C.B. Wiebel, and K.R. Gegenfurtner. Visual and haptic representations of material properties. *Multisensory research*, 26(5):429–455, 2013.
  - [20] M. Gentilucci, I. Toni, E. Daprati, and M. Gangitano. Tactile input of the hand and the control of reaching to grasp movements. *Experimental brain research*, 114(1):130–137, 1997.
  - [21] H. Yousef, M. Boukallel, and K. Althoefer. Tactile sensing for dexterous in-hand manipulation in roboticsa review. *Sensors and Actuators A: physical*, 167(2):171–187, 2011.

- [22] A.M. Almassri, W.Z. Wan Hasan, S.A. Ahmad, D.N. Talib A.J. Ishak, A.M. Ghazali, and C. Wada. Pressure sensor: state of the art, design, and application for robotic hand. *Journal of Sensors*, 2015.
- [23] J.M. Romano and K.J. Kuchenbecker. Methods for robotic tool-mediated haptic surface recognition. In *IEEE Haptics Symposium (HAPTICS)*, pages 49–56, February 2014.
- [24] E. Kerr, T.M. McGinnity, and S. Coleman. Material classification based on thermal and surface texture properties evaluated against human performance. In *13th International Conference on control automation robotics & vision (ICARCV)*, pages 10–12, 2014.
- [25] V. Chu, I. McMahon, L. Riano, C.G. McDonald, Q. He, J. Martinez-Tejada, M. Arrigo, T. Darroll, and K.J. Kuchenbecker. Using robotic exploratory procedures to learn the meaning of haptic adjectives. In *Proceedings - IEEE International Conference on In Robotics and Automation (ICRA)*, pages 3048–3055, 2013.
- [26] Z. Kappassov, J.A. Corrales, and V. Perdereau. Tactile sensing in dexterous robot handsreview. *Robotics and Autonomous Systems*, 74:195–220, 2015.
- [27] S. Stassi, V. Cauda, G. Canavese, and C.F. Pirri. Flexible tactile sensing based on piezoresistive composites: a review. *Sensors*, 14(3):5296–5332, 2014.
- [28] J. Fraden. *Handbook of modern sensors*, volume 3. New York: Springer, 2010.
- [29] I.D. Mayergoyz. *Mathematical models of hysteresis and their applications*. Academic Press, 2003.

- 
- [30] M.R. Cutkosky, R. D. Howe, and W.R. Provancher. Force and tactile sensors. In *Springer Handbook of Robotics*, chapter 19, pages 455–476. Springer Berlin Heidelberg, 2008.
  - [31] L. Seminara, M. Capurro, P. Cirillo, G. Cannata, and M. Valle. Electromechanical characterization of piezoelectric pvdf polymer films for tactile sensors in robotics applications. *Sensors and Actuators A: Physical*, 169(1):49–58, 2011.
  - [32] R.S. Dahiya and M. Valle. *Tactile sensing for robotic applications*. INTECH Open Access Publisher, 2008.
  - [33] M. Razavy. *Quantum theory of tunneling*. World Scientific, 2013.
  - [34] R. Walker. Developments in dextrous hands for advanced robotic applications. In *In Proc. the Sixth Biannual World Automation Congress*, pages 123–128, Seville, Spain, 2004.
  - [35] P. Kampmann and F. Kirchner. Integration of fiber-optic sensor arrays into a multi-modal tactile sensor processing system for robotic end-effectors. *Sensors*, 14(4):6854–6876, 2014.
  - [36] H. Yussof, M. Ohka, H. Suzuki, and N. Morisawa. Tactile sensing-based control system for dexterous robot manipulation. In *Advances in Computational Algorithms and Data Analysis*, chapter 15, pages 199–213. Springer Netherlands, 2009.
  - [37] S. Sareh, A. Jiang, A. Faragasso, Y. Noh, T. Nanayakkara, P. Dasgupta, L.D. Seneviratne, H.A. Wurdemann, and K. Althoefer. Bio-inspired tactile sensor sleeve for surgical soft manipulators. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 1454–1459. IEEE, 2014.

- 
- [38] R.S. Dahiya and M. Valle. *Robotic tactile sensing: technologies and system*. Springer Science & Business Media, 2012.
  - [39] M.I. Tiwana, S.J. Redmond, and N.H. Lovell. A review of tactile sensing technologies with applications in biomedical engineering. *Sensors and Actuators A: Physical*, 179:17–31, 2012.
  - [40] J.M. Romano, K. Hsiao, G. Niemeyer, S. Chitta, and K.J. Kuchenbecker. Human-inspired robotic grasp control with tactile sensing. *IEEE Transactions on Robotics*, 27(6):1067–1079, 2011.
  - [41] L.T. Jiang and J.R. Smith. Seashell effect pretouch sensing for robotic grasping. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2851–2858, May 2012.
  - [42] J. Fishel, V.J. Santos, and G.E. Loeb. A robust micro-vibration sensor for biomimetic fingertips. In *IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 659–663, October 2008.
  - [43] Y. Tenzer, L.P. Jentoft, and R.D. Howe. Inexpensive and easily customized tactile array sensors using mems barometers chips. *IEEE Robotics and Automation Magazine*, 21(3):89–95, 2014.
  - [44] R.S. Johansson and J.R. Flanagan. Coding and use of tactile signals from the fingertips in object manipulation tasks. *Nature Reviews Neuroscience*, 10(5):345–359, 2009.
  - [45] N. Wettels, J.A. Fishel, and G.E. Loeb. Multimodal tactile sensor. In *The Human Hand as an Inspiration for Robot Hand Development*, pages 405–429. Springer, 2014.

- 
- [46] J.S. Son, M.R. Cutkosky, and R.D. Howe. Comparison of contact sensor localization abilities during manipulation. *Robotics and autonomous systems*, 17(4):217–233, 1996.
- [47] K.J. Kuchenbecker, W.R. Provancher, G. Niemeyer, and M.R. Cutkosky. Haptic display of contact location. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS), 12th International Symposium on*, pages 40–47. IEEE, 2004.
- [48] Y. Yamada, T. Morizono, Y. Umetani, and H. Takahashi. Highly soft viscoelastic robot skin with a contact object-location-sensing capability. *IEEE Transactions on Industrial electronics*, 52(4):960–968, 2005.
- [49] W.R. Provancher, M.R. Cutkosky, K.J. Kuchenbecker, and G. Niemeyer. Contact location display for haptic perception of curvature and object motion. *The International Journal of Robotics Research*, 24(9):691–702, 2005.
- [50] N. Wettels and G.E. Loeb. Haptic feature extraction from a biomimetic tactile sensor: Force, contact location and curvature. In *Robotics and Biomimetics (ROBIO), IEEE International Conference on*, pages 2471–2478. IEEE, 2011.
- [51] N.F. Lepora, U. Martinez-Hernandez, H. Barron-Gonzalez, M. Evans, G. Metta, and T.J. Prescott. Embodied hyperacuity from bayesian perception: Shape and position discrimination with an icub fingertip sensor. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 4638–4643. IEEE, 2012.

- 
- [52] N.F. Lepora, U. Martinez-Hernandez, M. Evans, L. Natale, G. Metta, and T.J. Prescott. Tactile superresolution and biomimetic hyperacuity. *IEEE Transactions on Robotics*, 31(3):605–618, 2015.
- [53] H. Liu, K.C. Nguyen, V. Perdereau, J. Bimbo, J. Back, M. Godden, L.D. Seneviratne, and K. Althoefer. Finger contact sensing and the application in dexterous hand manipulation. *Autonomous Robots*, 39(1):25–41, 2015.
- [54] A. Molchanov, O. Kroemer, Z. Su, and G.S. Sukhatme. Contact localization on grasped objects using tactile sensing. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 216–222. IEEE, 2016.
- [55] K.M. Lynch, H. Maekawa, and K. Tanie. Manipulation and active sensing by pushing using tactile feedback. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 416–421, 1992.
- [56] M.R. Cutkosky and J.M. Hyde. Manipulation control with dynamic tactile sensing. In *6th International Symposium on Robotics Research*, volume 117, 1993.
- [57] H. Maekawa, K. Tanie, and K. Komoriya. Manipulation of an unknown object by multifingered hand with rolling contact using tactile feedback. *Transactions of the Society of Instrument and Control Engineers*, 31(9):1462–1470, 1995.
- [58] H. Maekawa, K. Tanie, K. Komoriya, M. Kaneko, C. Horiguchi, and T. Sugawara. Development of a finger-shaped tactile sensor and its

- evaluation by active touch. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 1327–1334. IEEE, 1992.
- [59] H. Zhang and N.N. Chen. Control of contact via tactile sensing. *IEEE Transactions on Robotics and Automation*, 16(5):482–495, 2000.
- [60] Q. Li, C. Schürmann, R. Haschke, and H. Ritter. A control framework for tactile servoing. In *Robotics: Science and Systems(RSS)*, 2013.
- [61] R. Li, R. Platt, W. Yuan, A. ten Pas, N. Roscup, M.A. Srinivasan, and E. Adelson. Localization and manipulation of small parts using gelsight tactile sensing. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 3988–3993. IEEE, 2014.
- [62] N.F. Lepora, K. Aquilina, and L. Cramphorn. Exploratory tactile servoing with active touch. *IEEE Robotics and Automation Letters*, 2(2):1156–1163, 2017.
- [63] Q. Li, R. Haschke, and H. Ritter. Learning a tool’s homogeneous transformation by tactile-based interaction. In *Humanoid Robots (Humanoids), 16th IEEE-RAS International Conference on*, pages 416–421. IEEE, 2016.
- [64] Z. Kappassov, J.A. Corrales-Ramon, and V. Perdereau. Zmp features for touch driven robot control via tactile servo. In *International Symposium on Experimental Robotics*, pages 234–243. Springer, 2016.
- [65] Q. Li, R. Haschke, and H. Ritter. A visuo-tactile control framework for manipulation and exploration of unknown objects. In *Humanoid Robots (Humanoids), 15th IEEE-RAS International Conference on*, pages 610–615. IEEE, 2015.

- 
- [66] C.A. Jara, J. Pomares, F.A. Candelas, and F. Torres. Control framework for dexterous manipulation using dynamic visual servoing and tactile sensors feedback. *Sensors*, 14(1):1787–1804, 2014.
  - [67] Y. Chebotar, O. Kroemer, and J. Peters. Learning robot tactile sensing for object manipulation. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 3368–3375. IEEE, 2014.
  - [68] A. Petrovskaya, O. Khatib, S. Thrun, and A.Y. Ng. Bayesian estimation for autonomous object manipulation based on tactile sensors. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 707–714. IEEE, 2006.
  - [69] A. Cavallo, G. De Maria, C. Natale, and S. Pirozzi. Slipping detection and avoidance based on kalman filter. *Mechatronics*, 24(5):489–499, 2014.
  - [70] F. Veiga, H. Van Hoof, J. Peters, and T. Hermans. Stabilizing novel objects by learning to predict tactile slip. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 5065–5072. IEEE, 2015.
  - [71] B. Heyneman and M.R. Cutkosky. Slip classification for dynamic tactile array sensors. *The International Journal of Robotics Research*, 35(4):404–421, 2016.
  - [72] Z. Su, K. Hausman, Y. Chebotar, A. Molchanov, G.E. Loeb, G.S. Sukhatme, and S. Schaal. Force estimation and slip detection/classification for grip control using a biomimetic tactile sensor. In *Humanoid Robots (Humanoids), 15th IEEE-RAS International Conference on*, pages 297–303. IEEE, 2015.



- 
- [73] H. Yussof, M. Ohka, J. Takata, Y. Nasu, and M. Yamano. Low force control scheme for object hardness distinction in robot manipulation based on tactile sensing. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 3443–3448. IEEE, 2008.
- [74] H. Liu, Y. Wu, F. Sun, and D. Guo. Recent progress on tactile object recognition. *International Journal of Advanced Robotic Systems*, 14(4):1729881417717056, 2017.
- [75] M. Kaboli, D. Feng, K. Yao, P. Lanillos, and G. Cheng. A tactile-based framework for active object learning and discrimination using multi-modal robotic skin. *IEEE Robotics and Automation Letters*, 2017.
- [76] A. Schneider, J. Sturm, C. Stachniss, M. Reiser, H. Burkhardt, and W. Burgard. Object identification with tactile sensors using bag-of-features. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 243–248. IEEE, 2009.
- [77] L. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International journal of computer vision*, 73(2):213–238, 2007.
- [78] H. Liu, Y. Yu, F. Sun, and J. Gu. Visual–tactile fusion for object recognition. *IEEE Transactions on Automation Science and Engineering*, 14(2):996–1008, 2017.
- [79] J. Fishel and G.E. Loeb. Bayesian exploration for intelligent identification of textures. *Frontiers in neurorobotics*, 6, 2012.

- 
- [80] D. Xu, G.E. Loeb, and J.A. Fishel. Tactile identification of objects using bayesian exploration. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 3056–3061, May 2013.
- [81] M.K. Johnson, F. Cole, A. Raj, and E.H. Adelson. Microgeometry capture using an elastomeric sensor. In *ACM Transactions on Graphics (TOG)*, volume 30(4), page 46. ACM, 2011.
- [82] S. Chitta, J. Sturm, M. Piccoli, and W. Burgard. Tactile sensing for mobile manipulation. *IEEE Transactions on Robotics*, 27(3):558–568, 2011.
- [83] A.M. Okamura and M.R. Cutkosky. Feature detection for haptic exploration with robotic fingers. *The International Journal of Robotics Research*, 20(12):925–938, 2001.
- [84] T.J. Prescott, M.E. Diamond, and A.M. Wing. Active touch sensing. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 366(1581):2989–2995, 2011.
- [85] J. Tegin and J. Wikander. Tactile sensing in intelligent robotic manipulation—a review. *Industrial Robot: An International Journal*, 32(1):64–70, 2005.
- [86] Z. Su, J.A. Fishel, T. Yamamoto, and G.E. Loeb. Use of tactile feedback to control exploratory movements to characterize object compliance. *Frontiers in Neurorobotics*, 6:1–9, July 2012.
- [87] U. Martinez-Hernandez, T.J. Dodd, M.H. Evans, T.J. Prescott, and N.F. Lepora. Active sensorimotor control for tactile exploration. *Robotics and Autonomous Systems*, 87:15–27, 2017.

- [88] M. Regoli, N. Jamali, G. Metta, and L. Natale. Controlled tactile exploration and haptic object recognition. *arXiv preprint arXiv:1706.08697*, 2017.
- [89] J. Edwards, J. Lawry, and C. Melhuish. Extracting textural features from tactile sensors. *Bioinspiration & Biomimetics*, 3(3):035002, 2008.
- [90] F. De Boissieu, C. Godin, B. Guilhamat, D. David, C. Serviere, and D. Baudois. Tactile texture recognition with a 3-axial force mems integrated artificial finger. In *Robotics: Science and Systems*, pages 49–56, June 2009.
- [91] J. Sinapov, V. Sukhoy, R. Sahai, and A. Stoytchev. Vibrotactile recognition and categorization of surface textures by a humanoid robot. *IEEE Transactions on Robotics*, 3(27):488–497, 2011.
- [92] N. Jamali and C. Sammut. Majority voting: material classification by tactile sensing using surface texture. *IEEE Transactions on Robotics*, 32(27):508–521, 2011.
- [93] S. Decherchi, P. Gastaldo, R.S. Dahiya, M. Valle, and R. Zunino. Tactile-data classification of contact materials using computational intelligence. *IEEE Transactions on Robotics*, 3(27):635–639, 2011.
- [94] C.M. Oddo, M. Controzzi, L. Beccai, C. Cipriani, and M.C. Carrozza. Roughness encoding for discrimination of surfaces in artificial active-touch. *IEEE Transactions on Robotics*, 3(27):522–533, 2011.
- [95] P. Dallaire, P. Giguère, D. Émond, and B. Chaib-draa. Autonomous tactile perception: A combined improved sensing and bayesian non-parametric approach. *Robotics and autonomous systems*, 62(4):422–435, 2014.

- 
- [96] D.S. Chathuranga, V.A. Ho, and S. Hirai. Investigation of a biomimetic fingertip's ability to discriminate fabrics based on surface textures. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics: Mechatronics for Human Wellbeing (AIM)*, pages 1667–1674, 2013.
- [97] R. Li and E.H. Adelson. Sensing and recognizing surface textures using a gelsight sensor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1241–1247, 2013.
- [98] E. Hellinger. Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die reine und angewandte Mathematik*, 136:210–271, 1909.
- [99] D.S Chathuranga, Z. Wang, Y. Noh, T. Nanayakkara, and S. Hirai. Robust real time material classification algorithm using soft three axis tactile sensor: Evaluation of the algorithm. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, 2015.
- [100] S.S. Baishya and B. Bäuml. Robust material classification with a tactile skin using deep learning. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 8–15. IEEE, 2016.
- [101] T. Bhattacharjee, J. Wade, and C.C. Kemp. Material recognition from heat transfer given varying initial conditions and short-duration contact. In *Proceedings of Robotics: Science and Systems(RSS)*, 2015.
- [102] J. Wade, T. Bhattacharjee, R.D. Williams, and C.C. Kemp. A force and thermal sensing skin for robots in human environments. *Robotics and Autonomous Systems*, 96:1–14, 2017.

- [103] M. Kaboli, R. Walker, and G. Cheng. In-hand object recognition via texture properties with robotic hands, artificial skin, and novel tactile descriptors. In *Humanoid Robots (Humanoids), 15th IEEE-RAS International Conference on*, pages 1155–1160. IEEE, 2015.
- [104] E. Kerr, T.M. McGinnity, and S. Coleman. Material classification based on thermal propertiesa robot and human evaluation. In *Robotics and Biomimetics (ROBIO), IEEE International Conference on*, pages 1048–1053. IEEE, 2013.
- [105] J. Hoelscher, J. Peters, and T. Hermans. Evaluation of tactile feature extraction for interactive object recognition. In *Humanoid Robots (Humanoids), 15th IEEE-RAS International Conference on*, pages 310–317. IEEE, 2015.
- [106] T. Yoshikawa. Multifingered robot hands: Control for grasping and manipulation. *Annual Reviews in Control*, 34(2):199–208, 2010.
- [107] A. Bicchi and V. Kumar. Robotic grasping and contact: A review. In *Robotics and Automation (ICRA), IEEE International Conference on*, volume 1, pages 348–353. IEEE, 2000.
- [108] A. Bicchi. Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity. *IEEE Transactions on robotics and automation*, 16(6):652–662, 2000.
- [109] M. Ciocarlie, K. Hsiao, E.G. Jones, S. Chitta, R.B. Rusu, and I.A. Şucan. Towards reliable grasping and manipulation in household environments. In *Experimental Robotics*, pages 241–252. Springer, 2014.
- [110] M.A. Roa and R. Suárez. Grasp quality measures: review and performance. *Autonomous robots*, 38(1):65–88, 2015.

- 
- [111] K. Hsiao, S. Chitta, M. Ciocarlie, and E.G. Jones. Contact-reactive grasping of objects with partial shape information. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 1228–1235. IEEE, 2010.
  - [112] J.A. Corrales, F. Torres, and V. Perdereau. Finger readjustment algorithm for object manipulation based on tactile information. *International Journal of Advanced Robotic Systems*, 10(1):9, 2013.
  - [113] H. Dang and P.K. Allen. Stable grasping under pose uncertainty using tactile feedback. *Autonomous Robots*, 36(4):309–330, 2014.
  - [114] M. Li, Y. Bekiroglu, D. Kragic, and A. Billard. Learning of grasp adaptation through experience and tactile sensing. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 3339–3346. IEEE, 2014.
  - [115] Z. Chen, T. Wimböck, M.A. Roa, B. Pleintinger, M. Neves, C. Ott, C. Borst, and N.Y. Lii. An adaptive compliant multi-finger approach-to-grasp strategy for objects with position uncertainties. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 4911–4918. IEEE, 2015.
  - [116] M. Li, K. Hang, D. Kragic, and A. Billard. Dexterous grasping under shape uncertainty. *Robotics and Autonomous Systems*, 75:352–364, 2016.
  - [117] R. Platt Jr, A.H. Fagg, and R.A. Grupen. Null-space grasp control: Theory and experiments. *IEEE Transactions on Robotics*, 26(2):282–295, 2010.

- 
- [118] Z. Li, P. Hsu, and S. Sastry. Grasping and coordinated manipulation by a multifingered robot hand. *The International Journal of Robotics Research*, 8(4):33–50, 1989.
- [119] K. Tahara, K. Maruta, A. Kawamura, and M. Yamamoto. Externally sensorless dynamic regrasping and manipulation by a triple-fingered robotic hand with torsional fingertip joints. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 3252–3257. IEEE, 2012.
- [120] M. Li, H. Yin, K. Tahara, and A. Billard. Learning object-level impedance control for robust grasping and dexterous manipulation. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 6784–6791. IEEE, 2014.
- [121] R.D. Howe, N. Popp, P. Akella, I. Kao, and M.R. Cutkosky. Grasping, manipulation, and control with tactile sensing. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 1258–1263. IEEE, 1990.
- [122] T. Takahashi, T. Tsuboi, T. Kishida, Y. Kawanami, S. Shimizu, M. Iribe, T. Fukushima, and M. Fujita. Adaptive grasping by multi fingered hand with tactile sensor based on robust force and position control. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 264–271. IEEE, 2008.
- [123] J. Felip and A. Morales. Robust sensor-based grasp primitive for a three-finger robot hand. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 1811–1816. IEEE, 2009.

- 
- [124] J.S. Son, R. Howe, J. Wang, and G.D. Hager. Preliminary results on grasping with vision and touch. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, volume 3, pages 1068–1075. IEEE, 1996.
- [125] N. Sommer. *Multi-contact tactile exploration and interaction with unknown objects*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2017.
- [126] J. López-Coronado, J.L. Pedreño-Molina, A. Guerrero-González, and P. Gorce. A neural model for visual-tactile-motor integration in robotic reaching and grasping tasks. *Robotica*, 20(1):23–31, 2002.
- [127] L. Natale and E. Torres-Jara. A sensitive approach to grasping. In *Proceedings of the sixth international workshop on epigenetic robotics*, pages 87–94, 2006.
- [128] Y-B. Jia. Grasping curved objects through rolling. In *Robotics and Automation (ICRA), IEEE International Conference on*, volume 1, pages 377–382. IEEE, 2000.
- [129] D. Wang, B.T. Watson, and A.H. Fagg. A switching control approach to haptic exploration for quality grasps. In *Proceedings of the Workshop on Robot Manipulation: Sensing and Adapting to the Real World*, 2007.
- [130] A. Bierbaum, M. Rambow, T. Asfour, and R. Dillmann. A potential field approach to dexterous tactile exploration of unknown objects. In *Humanoid Robots (Humanoids), 8th IEEE-RAS International Conference on*, pages 360–366. IEEE, 2008.
- [131] A. Bierbaum, M. Rambow, T. Asfour, and R. Dillmann. Grasp affordances from multi-fingered tactile exploration using dynamic potential



- fields. In *Humanoid Robots (Humanoids), 9th IEEE-RAS International Conference on*, pages 168–174. IEEE, 2009.
- [132] R. Platt. Learning grasp strategies composed of contact relative motions. In *Humanoid Robots (Humanoids), 7th IEEE-RAS International Conference on*, pages 49–56. IEEE, 2007.
- [133] A. Edsinger and C.C. Kemp. Human-robot interaction for cooperative manipulation: Handling objects to one another. In *16th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2007.
- [134] M. Huber, M. Rickert, A. Knoll, T. Brandt, and S. Glasauer. Human-robot interaction in handing-over tasks. In *17th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 107–112, August 2008.
- [135] E. Martinson, A. Huaman Quispe, and K. Oguchi. Towards understanding user preferences in robot-human handovers: How do we decide? In *Robot and Human Interactive Communication (RO-MAN), 26th IEEE International Symposium on*. IEEE, 2017.
- [136] M. Cakmak, S.S. Srinivasa, M.K. Lee, J. Forlizzi, and S. Kiesler. Human preferences for robot-human hand-over configurations. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 1986–1993. IEEE, 2011.
- [137] H.B. Suay and E.A. Sisbot. A position generation algorithm utilizing a biomechanical model for robot-human object handover. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 3776–3781. IEEE, 2015.

- 
- [138] E. Martinson, A. Blasdel, and E.A. Sisbot. Personalizing object handover with an electronic health record. In *Robot and Human Interactive Communication (RO-MAN), 25th IEEE International Symposium on*, pages 1144–1149. IEEE, 2016.
- [139] K. Nagata, Y. Oosaki, M. Makikura, and H. Tsukune. Delivery by hand between human and robot based on fingertip force-torque information. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 750–757, 1998.
- [140] J. Aleotti, V. Micelli, and S. Caselli. An affordance sensitive system for robot to human object handover. *International Journal of Social Robotics*, 6(4):653–666, 2014.
- [141] J. Bohren, R.B. Rusu, E.G. Jones, E. Marder-Eppstein, C. Pantofaru, M. Wise, L. Mösenlechner, W. Meeusen, and S. Holzer. Towards autonomous robotic butlers: Lessons learned with the PR2. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 5568–5575, 2011.
- [142] I. Kim and H. Inooka. Hand-over of an object between human and robot. In *IEEE International Workshop on robot and Human Communication*, pages 199–203, 1992.
- [143] W. P. Chan, C. A. Parker, H. F. Van der Loos, and E. A. Croft. Grip forces and load forces in handovers: implications for designing human-robot handover controllers. In *In Proceedings of the seventh annual ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, volume 6, pages 9–16, March 2012.

- 
- [144] W.P. Chan, C.A.C. Parker, H.F.M. Van der Loos, and E.A. Croft. A human-inspired object handover controller. *International Journal on Robotics Research (IJRR)*, 32(8):971–983, 2013.
- [145] J.R. Medina Hernández, F. Duvallet, M. Karnam, and A. Billard. A human-inspired controller for fluid human-robot handovers. In *Humanoid Robots (Humanoids), 16th IEEE-RAS International Conference on*, pages 324–331, 2016.
- [146] S. Parastegari, E. Noohi, B. Abbasi, and M. Žefran. A fail-safe object handover controller. In *IEEE International Conference on In Robotics and Automation (ICRA)*, pages 2003–2008, May 2016.
- [147] A.M. Okamura, N. Smaby, and M.R. Cutkosky. An overview of dexterous manipulation. In *Robotics and Automation (ICRA), IEEE International Conference on*, volume 1, pages 255–262. IEEE, 2000.
- [148] G. ElKoura and K. Singh. Handrix: animating the human hand. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 110–119. Eurographics Association, 2003.
- [149] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern classification*. John Wiley & Sons, 2012.
- [150] J.A. Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, 4(510):126, 1998.
- [151] J.M. Bernardo, M.J. Bayarri, J.O. Berger, A.P. Dawid, D. Heckerman, A.F.M. Smith, and M. West. The variational bayesian em algorithm for incomplete data: with application to scoring graphical model structures. *Bayesian statistics*, 7:453–464, 2003.

- 
- [152] C. Biernacki, G. Celeux, and G. Govaert. Choosing starting values for the em algorithm for getting the highest likelihood in multivariate gaussian mixture models. *Computational Statistics & Data Analysis*, 41(3):561–575, 2003.
- [153] S. Lloyd. Least squares quantization in PCM. *IEEE transactions on information theory*, 28(2):129–137, 1982.
- [154] J.M. Pena, J.A. Lozano, and P. Larranaga. An empirical comparison of four initialization methods for the k-means algorithm. *Pattern recognition letters*, 20(10):1027–1040, 1999.
- [155] K.P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [156] W. Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997.
- [157] C.M. Bishop. *Pattern recognition and machine learning*. springer, 2006.
- [158] M.T. Hagan and M.B. Menhaj. Training feedforward networks with the marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6):989–993, 1994.
- [159] K.I. Kim, K. Jung, S.H. Park, and H.J. Kim. Support vector machines for texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1542–1550, 2002.
- [160] H.C. Taneja. *Advanced Engineering Mathematics, Vol. 2*. I. K. International Pvt Ltd, New Delhi, India, 2008.

- 
- [161] J.W. Cooley and J.W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.
- [162] S.W. Smith. *The scientist and engineer’s guide to digital signal processing*. California Technical Pub. San Diego, 1997.
- [163] A.V. Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999.
- [164] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [165] R. Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
- [166] B. Siciliano and O. Khatib. *Springer handbook of robotics*. Springer, 2016.
- [167] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.
- [168] Y.C. Ho and R. Lee. A bayesian approach to problems in stochastic estimation and control. *IEEE Transactions on Automatic Control*, 9(4):333–339, 1964.
- [169] B. León, A. Morales, and J. Sancho-Bru. *From robot to human grasping simulation*. Springer, 2014.
- [170] J.R. Napier. The prehensile movements of the human hand. *Bone & Joint Journal*, 38(4):902–913, 1956.

- 
- [171] J.M.F. Landsmeer. Power grip and precision handling. *Annals of the rheumatic diseases*, 21(2):164, 1962.
  - [172] T. Feix, J. Romero, H.B. Schmiedmayer, A.M. Dollar, and D. Kragic. The grasp taxonomy of human grasp types. *IEEE Transactions on Human-Machine Systems*, 46(1):66–77, 2016.
  - [173] Z. Su, J. Fishel, T. Yamamoto, and G.E. Loeb. Use of tactile feedback to control exploratory movements to characterize object compliance. *Frontiers in Neurorobotics*, 6:7, 2012.
  - [174] J. Guo and N. Jiangnan. Kinematics and Statics Analysis of Dexterous Hand. In *Advances in Engineering Research, International Conference on Mechatronics, Control and Materials (ICMCM)*, Changsha, China, October 29-30 2016.
  - [175] Peter Corke. *Robotics, Vision and Control: Fundamental Algorithms In MATLAB® Second, Completely Revised*, volume 118. Springer, 2017.
  - [176] Z. Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012.
  - [177] R.B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.