UNIVERSITY OF BIRMINGHAM

Estimation of objects' inertial parameters, and their usage in robot grasping and manipulation

By

Nikolaos MAVRAKIS

A thesis submitted to the University of Birmingham for the degree of

Doctor of Philosophy

School of Metallurgy and Materials
College of Engineering and Physical Sciences
University of Birmingham

September 27, 2019

UNIVERSITY OF
BIRMINGHAM

University of Birmingham Research Archive

e-theses repository

This unpublished thesis/dissertation is copyright of the author and/or third parties. The intellectual property rights of the author or third parties in respect of this work are as defined by The Copyright Designs and Patents Act 1988 or as modified by any successor legislation.

Any use made of information contained in this thesis/dissertation must be in accordance with that legislation and must be properly acknowledged. Further distribution or reproduction in any format is prohibited without the permission of the copyright holder.

"The true delight is in the finding out rather than in the knowing."

Isaac Asimov

Abstract

The subject of this thesis is the estimation of an object's inertial parameters by a robotic arm, and the exploitation of those parameters in the design of efficient manipulation criteria. The inertial parameters of objects describe the resistance of the object to an applied force, and dictate its motion. Research has shown that humans intuitively exploit them for their everyday manipulations. As humans are very capable of performing efficient manipulations, it is natural that robots should use the inertial parameters as well. Additionally, as the inertial parameters are not straightforward to calculate, there is the need for development of methods that can estimate them online. This thesis focuses on two directions, developing novel methods so that robots can accurately estimate the inertial parameters of an object, as well as developing manipulation criteria that can make robot task completion more efficient. The relevant literature is gathered, categorised and analytically described, and the innovation gaps are identified. The thesis offers novel research solutions on the problem of estimation of the inertial parameters with minimal robot interaction. The paradigm is shifted from the existing literature, and a data-driven estimation algorithm is introduced, that achieves accurate results with both simulated and real data. Additionally, the presented research is offering novel manipulation criteria that are affected by the object's inertial parameters. The results suggest that knowledge of the inertial parameters can make the robot tasks more power-efficient and safe to their surroundings. The core methodology is shown to be versatile to the robotic platform. Though most experiments are performed on a terrestrial robot, a numerical example is also shown for a space robot. The results of the thesis suggest that the developed methods can be used in various environments, with the most suitable being extreme environments where accuracy, efficiency and autonomy is required.

Acknowledgements

I would like to primarily thank my supervisors, Prof. Rustam Stolkin and Dr Amir Masoud Ghalamzan Esfahani, for their continuous support and guidance in the development of this thesis, as well as the journey through robotics research.

Many thanks to all the colleagues in Birmingham and everywhere, that helped me through this exciting phase of research.

I would like to thank my parents and brother, and my friends for supporting and believing in me, in every aspect of my academic start. Finally, special thanks to Evi for supporting me in my life and giving me strength through all these years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Robotic manipulation across various environments

One of the first robots to perform manipulation tasks was the Unimate manipulator by Unimation. It was a 6-axis programmable arm, with a metallic gripper as end-effector. It was installed firstly in an assembly line of a General Motors plant. It could be pre-programmed to execute a number of pick-and-place and welding tasks. The Unimate robot did not have capabilities for autonomous and intelligent operations, and the purpose for its first installation was to automate the stacking of metallic casts.

Ever since this first introduction of such a system to an assembly line, the intelligence and autonomy capabilities of robots have increased dramatically. Instead of merely being programmed to execute pre-calculated motions, robots can perceive their environment and surroundings though various sensing modalities, make decisions, plan for actions, and execute them efficiently. Industrial production has skyrocketed with the incorporation of more intelligent robotic manipulators in assembly lines. While industrial plants still remain the most prominent environments for robotic operations, autonomous systems are gradually more and more employed for all sorts of applications. And while the main purpose of a robot was to handle known objects or parts coming on an assembly line and executing pre-designed tasks and trajectories with them, the deployment of robots in uncontrolled environments requires intelligent manipulation of objects with various geometries.

Health care robots are increasingly incorporated in hospitals and care homes. The increasing amount of ageing populations and the rising life expectancy in many countries worldwide are a crucial factor for introduction of robots to hospitals, care homes, and other healthcare facilities. Some examples are telemanipulators and mobile rehabilitation robots that assist with everyday patient care. To do so, healthcare robots have to manipulate trays with food, medication, and other supplies, as well as handle care equipment for the patients they treat.

The fast pace of modern life, and the increased amounts of stress and loneliness humans tend to feel during their lives have left large amounts of population in need of care in their homes. Robots are bound to be more and more prevalent in household environments, for assisting with everyday household tasks, as well as companions and carers hoe elderly and disabled people. Such a robot is expected to manipulate all types of objects existing in a household environment, such as dishes for stacking a dishwasher, clothes for ironing, and food and supplies for fetching to a human.

With the rise of the world population, and the dietary choices of many people shifting towards vegetarian options, robots are starting to be deployed in crop fields and agricultural sites in an attempt to increase production and minimise costs. Typical tasks of robots in agriculture are inspection of crop state and data gathering, fertiliser spraying, and the picking up of fruit and other produce. Such robots need to perform both dexterous and delicate manipulations to handle produce, as well as more brute manipulations to handle equipment.

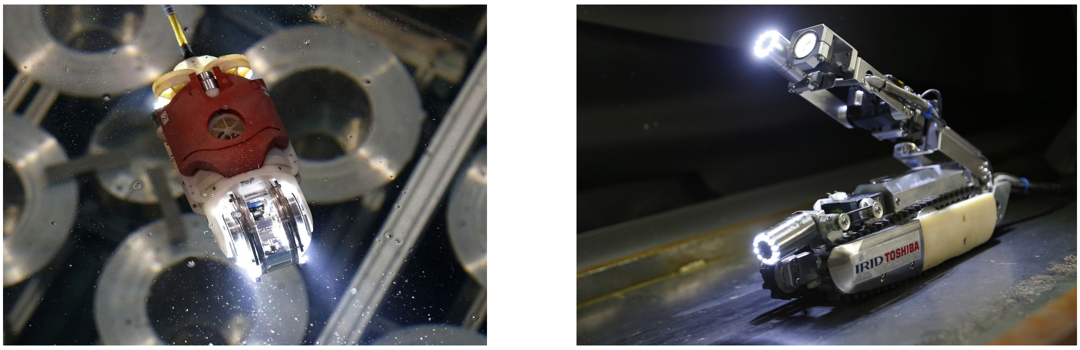
Search-and-rescue scenarios more and more require the presence of robots for related tasks. Natural disasters can happen in any place on Earth, and they leave injured people and wreckage behind them. Offering help to injured people in such conditions can be sped up significantly by employing robots for area observation and identification, terrain traversing, and debris manipulation. Similar cases include accident sites, and war zones, where robots can aid in offering assistance with debris clearance, area surveillance and bomb diffusal.

Finally, extreme environments are the prime candidate for future deployment of robots (Fig. 1.1). Underwater and offshore exploration and asset inspection require also the use of robotic systems due to their inaccessibility from humans. Robots can offer a way of conducting teleoperated inspection and maintenance, and manipulate underwater cables and valve handles. Additionally, the majority of the oceans on Earth, as well as a large number of lakes and rivers have not been explored, aside from sonar measurements. Employing robots on oceanic and lake environments can reveal a lot about their geological structure and oceanographic data, as well as retrieving objects of interest, such as salvage or archaeological objects. Robots are also becoming prevalent in nuclear plants (Fig. 1.2). Since the beginning of the Atomic age, many countries have been stockpiling the residual nuclear waste in landfills and warehouses. The waste containers are sometimes very old, and need to be cut, opened and their contents sorted and segregated, so as to ensure optimal storage conditions and free as much storage space as possible. Since the waste materials are of various types, forms and contamination levels, it is very dangerous for humans to execute such operations. As a result, robots need to be employed for that matter, and novel navigation, perception, manipulation and autonomy algorithms need to be developed. Finally, robots are being studied for and utilised in space applications, such as planetary exploration, on-orbit servicing, modular telescope assembly, active debris removal, and assistance of astronauts on board the International Space Station. Space robots have the additional challenge of manipulating large and heavy objects in zero-gravity environments, so their manipulation algorithms need to be adapted to work in such conditions.

It is clear that in this era of increased presence of robotic systems in various environments, robots will have to intelligently manipulate objects and adapt to uncertainties and new conditions. As a result, the study of robotic grasping and manipulation is one of the most rich and prominent robotics fields, with lots of challenges to overcome.

1.2 Challenges in object grasping and manipulation

The challenges that a deployed robot may meet are unique and dependent on the environment. Nonetheless, there exist specific common requirements that a robot



(a) (b)

Figure 1.1: Robots are used more and more in extreme environments for inspection of areas where humans can not enter. A typical example are robots in nuclear plants. (a) Sunfish-inspired robot for submerged reactor core inspection. (b) Scorpion robot developed for inspection of the plant's radioactive interior.

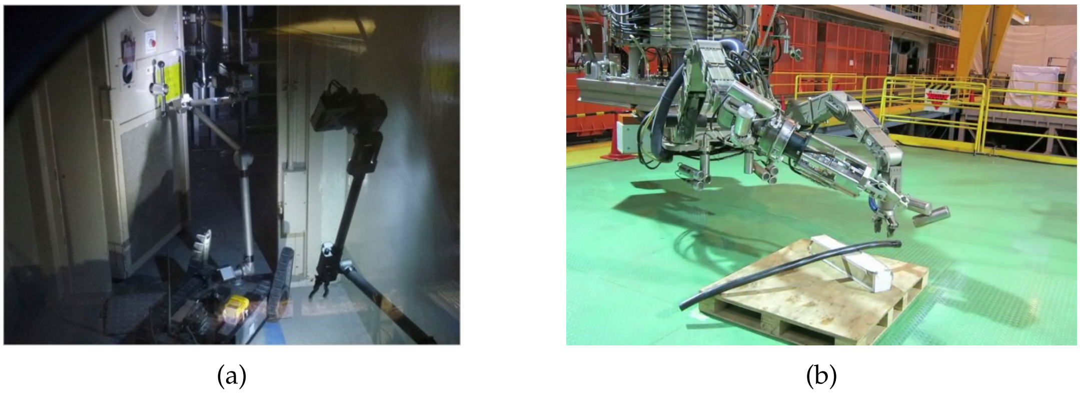
system must meet in order to ensure efficient and continuous operations.

1.2.1 Autonomy requirements

In many environments, robots need to perform a lot of operations for prolonged times with limited teleoperation, or no teleoperation at all. For example, robots on the Martian surface suffer from large latency of the radio-transmitted commands, as it takes several minutes for the communication signals to cross the Earth-Mars distance. Another example includes robots in deep mines or nuclear plants, where thick plant walls or underground caves may prohibit sending commands to the robot. It is evident that in such conditions, the robot needs to have the ability to make autonomous decisions to fulfill the assigned tasks. As a result, the robot needs to have adequate capability of perceiving the surroundings with various sensors, intelligent decision making ability, and planning and execution capabilities for actions within the perceived environment. The information about the surroundings need to be as rich as possible, for better planning and decision making.

1.2.2 Reliable teleoperation

In some cases, a communication channel needs to be established between the robot and a human user. The robot is then teleoperated and there is limited need for autonomous operations. Challenges for teleoperated robots include real-time sensor processing and control capabilities, unobstructed and minimum-latency communication between the controlling user and the robot platform, as well as rich and continuous sensor information for the user to evaluate. A typical example would be the robots in healthcare, working in spaces with patients that suffer from highly contagious diseases.



(a)

(b)

Figure 1.2: Among the challenges of extreme environments, accurate object manipulation is one of the most important. It enables the robot to access obstructed areas, retrieve samples and objects of interest, and help people. (a) A mobile manipulator designed for access in disaster situations, shown opening a door. (b) A robot developed for manipulation of heavier objects inside the Fukushima plant.

1.2.3 Shared autonomy

A paradigm that receives increasing attention is that of shared control, tele-autonomy or human-supervised autonomy. A robot is operating autonomously under close supervision by a human. The human sets a few higher level tasks (e.g. grasp the object, or move to a target position) to the robot, and the robot takes care of the underlying low-level perception, decision making, action and control to solve the task. The human can gain full control of the robot instantly at any time. A prominent example of shared autonomy operation are self-driving cars. After given a destination, the robot drives on its own, but with a human operator ready to regain full control if needed. Since shared autonomy operations include a level of decision-making by the robot, information acquisition about the environment is very important, as it increases efficiency and reduces execution time, thus minimizing the mental workload of the human user.

1.2.4 Power efficiency

As most robots need to continuously perform operations, power consumption and efficiency are important design factors. Robots with no recharging capability have limited lifespan, but even when the robot is capable of recharging, the power systems can only withstand a fixed number of charge-discharge cycles before they stop working. As a result, the actuation and motions of the robot need to be designed and executed with minimum power consumption. An autonomous robot can use information from the environment to make a minimum-power decision, e.g. by measuring the ground slope along the field of view, it can select the minimum-distance way to overcome a terrain bump, thus reducing the wheel operation time.

1.2.5 Resilient design

In many cases, the structural design and the components of the robot need to be designed to cope with harsh conditions and wear. A nuclear robot may need to be radiation and temperature resistant (depending on the environment of deployment and the dose rate of radiological materials that are present). Similarly, a robot in agriculture needs to be resistant to dust, rain, sunlight, heat and insects. The challenge of building resilient robot depends on the situation in which it will be used.

1.3 Object perception in robotics

One of the main properties of robots is the ability to perceive and handle objects. These objects can be useful in numerous ways e.g. a target object to be examined and manipulated, a tool to be used, or an obstacle to be avoided. The purpose of object perception is to extract unique properties of the object, that separate it from its surroundings, and use these properties in grasping and manipulation reasoning, planning, execution, and quality evaluation. Robots have numerous ways of perceiving their surroundings, using various types of sensors. This section provides some typical object information the robot can gather through the perception process, as well as methods of obtaining this information.

1.3.1 Visual information

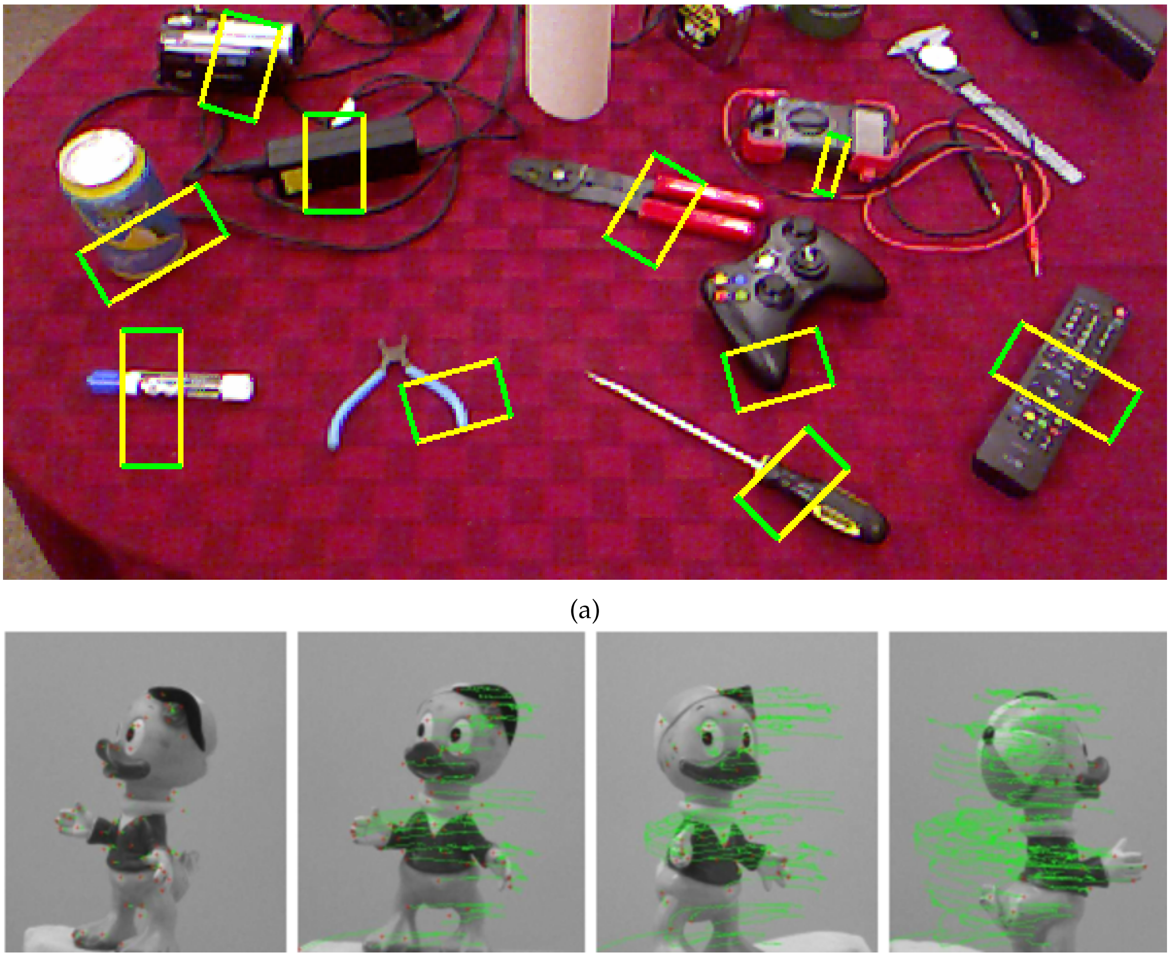
Perhaps the most common method of object perception in robotics is by using visual sensors. Examples of such sensors are RGB mono and stereo cameras, structured light sensors and event-driven cameras. The robot can visually perceive objects and surroundings by collecting raw data such as RGB or grayscale images, depth maps, infrared maps and point clouds. Information collected from these raw data leads to understanding about the object geometry and local shape, texture, and motion. This information can be used for numerous purposes such as feature motion tracking, grasp and motion planning, visual affordance learning etc. Examples are shown in Fig. 1.3.

1.3.2 Tactile information

Robots can perceive and learn a lot about objects by touching them. The way they extract information through touching is by using tactile sensors. A tactile sensor is sophisticated piece of hardware that can translate a physical contact with a touched object to a force measurement. Examples include resisting, magnetic and piezoelectric tactile sensors. A classic way of tactile sensing is using tactile arrays on robot fingers, that can generate a grid of force signals when an object is touched. More novel approaches include tactile skin sensors that can be worn on any part of the robot. By using tactile sensing, a robot can understand object geometries and generate partial 3D object models, infer the object's texture by inspecting the object surface, as well as predict stable grasps. Examples of tactile sensors are shown in Fig. 1.4.

1.4 Inertial parameters of objects

The main argument of this thesis is that, additionally to obtaining information on geometric and texture properties of an object, the robot can use visual and contact



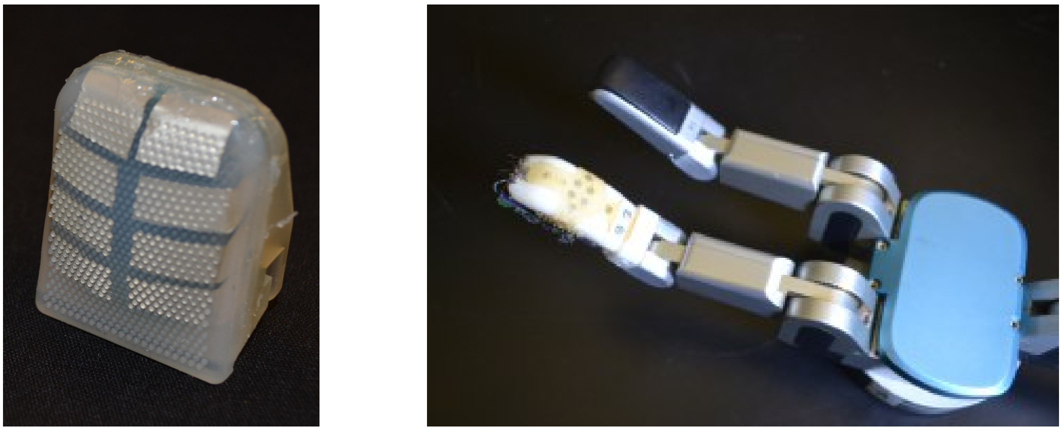
(a)

(b)

Figure 1.3: Examples of object visual perception in robotics. (a) Visual perception using RGB images. The robot is able to detect potential handle-like grasping points on objects. Image by Lenz et al., 2015. (b) SIFT features detected on an object, used for motion tracking. The features are extracted by filter processing of the acquired images, and can be detected from different views of the same object. Image by Delponte et al., 2006.

stimuli to estimate and use the inertial parameters of a rigid object for more efficient grasping and manipulation. With the term *Inertial Parameters* of the object, we refer to the object's mass m, moment of inertia tensor I and centre of mass (CoM) $\boldsymbol{p_{cm}}$. The mass contains information of the object's heaviness, and so the robot can understand how easy it is to lift or move it. The inertia tensor contains information of the object's mass distribution, and can be used to find what parts of an object are heavier than others. The CoM of the object is the 3-dimensional point around which the mass of the body is equally distributed. It offers an approximation of the object's mass distribution as a single point, which is useful for kinemodynamic modelling and visualisation.

Let O_w be a 3D fixed coordinate frame used as reference. Assuming our object is modelled as a rigid body with continuous volume, let V be the object's total volume, dV an elementary volume in a point with distance $\boldsymbol{r} = (x, y, z)$ from O_w, and $\rho(\boldsymbol{r})$ the object's density at that point. If the density is uniform along the body volume, then



(a) (b)

Figure 1.4: Examples of tactile sensors. (a) A tactile array sensor on a robot finger. The tactile array elements can be capacitive, piezoelectric, resisting etc., and when the robot touches an object they transform the measured property in a sensed force signal. Image from http://bdml.stanford.edu. (b) A flexible skin tactile sensing element. Flexible skin is a more advanced tactile sensing element that can be stretched to cover any part of the robot surface. They are capable of measure normal and sliding forces, as well as object vibration. Image from http://www.washington.edu/

$\rho(\boldsymbol{r})$ is a constant. The inertial properties of the whole body are then given by:

$$m = \int_V \rho(\boldsymbol{r})dV \tag{1.1}$$

$$\begin{aligned} I &= \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{pmatrix} \\ &= \begin{pmatrix} \int_V \rho(\boldsymbol{r})(y^2+z^2)dV & \int_V \rho(\boldsymbol{r})(-xy)dV & \int_V \rho(\boldsymbol{r})(-xz)dV \\ \int_V \rho(\boldsymbol{r})(-yx)dV & \int_V \rho(\boldsymbol{r})(x^2+z^2)dV & \int_V \rho(\boldsymbol{r})(-yz)dV \\ \int_V \rho(\boldsymbol{r})(-zx)dV & \int_V \rho(\boldsymbol{r})(-zy)dV & \int_V \rho(\boldsymbol{r})(y^2+x^2)dV \end{pmatrix} \end{aligned} \tag{1.2}$$

$$\boldsymbol{p}_{cm} = \frac{1}{m}\int_V \rho(\boldsymbol{r})\boldsymbol{r}dV \tag{1.3}$$

The mass m of the object is a positive real number. The inertia tensor I is symmetric and positive definite. The diagonal elements are named *moments of inertia* and the off diagonal elements *products of inertia*. The inertia tensor is always expressed w.r.t. a coordinate frame. To switch between different frames, the *parallel axis theorem* and *rotation theorem* are used. For a 3D body and any point in space, it is always possible to find a coordinate frame for which the product of inertia elements are zero, and the inertia tensor diagonal. The axes of this coordinate system are called *principal axes* of the object, and they can be found by solving the eigenvalue problem of the inertia tensor. The CoM may or may not lie inside the body.

In this thesis, we use the mass m, diagonal inertia tensor w.r.t. $\boldsymbol{p}_{cm}$, and CoM $\boldsymbol{p}_{cm}$. When the tensor needs to be expressed in a different frame, the parallel axis and rotation theorems are used.

1.5 Problem statement

Taking the previous analysis into consideration, the research question of this thesis is formed as follows:

How can a robot in an unknown environment estimate the inertial parameters (mass, CoM, inertia tensor) of an object, and how can these parameters be used to improve task-relevant grasping and manipulation?

As a consequence, the main research question can be broken down into two sub-problems, namely estimation of the inertial parameters and their usage in task-relevant grasping and manipulation. The thesis is inspired by the need for robots to manipulate heavy objects or objects with peculiar mass distributions, and as such, many of the problem configurations and evaluation criteria are inspired by such conditions.

1.5.1 Estimation of inertial parameters

The first problem that this thesis examines is the estimation of the inertial parameters of an object. Objects examined are rigid bodies, without any deformable or moving parts. For simplicity, the objects are assumed to have unknown but uniform density. Such an assumption is logical in some environments such as nuclear or industrial plants, many of the used components tend to be made of uniform materials (e.g. wooden pallets, metal tools and components, debris pieces, bricks etc.). The results of the presented experiments could be extended to non-uniform distribution objects, however this extension may not be straightforward. Since this thesis provides a paradigm shift in the estimation procedure, assumptions about simplicity were preferred. Finally, this thesis examines minimum interaction estimation methods. As will be presented analytically in the next Chapter, minimum interactions estimations are methods that extract as much information from the object as possible, and at the same time offer increased safety for unknown environments.

1.5.2 Usage in robot tasks

The second problem that is examined is the usage of the object's inertial parameters in robot task execution. While the term "robot tasks" is very broad, robots usually need to handle objects in order to examine them, place them in containers, or generally move around while grasping them. Such operations are conducted continuously, and as a result, efficiency is a key property. Intuitively, when a robot manipulates heavy objects or objects with peculiar mass distribution it applies more effort, and thus torque and power requirements increase. As a result, in this thesis the focus is on how the estimated dynamics of the object are combined with the dynamics of robots in typical motion tasks, such as pick-and-place, and how to make manipulation of objects more efficient by creating metrics for grasp selection and trajectory generation.

1.6 Contributions of the thesis and published papers

The relevant literature in the field of objects' inertial properties, and exploitation of these properties in robotics, is scarce and not interconnected. This thesis provides insight on these subjects, and connects them to provide fundamental and solid research on the topic.

1.6.1 Thesis novelties

The contributions of the thesis can be summarised as follows:

- Establishes the role of the inertial parameters of objects, as a valuable resource that robots should attempt to estimate in order to make robot operations more efficient and safe.

- Shifts the estimation procedure paradigm from analytical, model based approaches, to methods that make estimations based on large amounts of data and machine learning algorithms. Analytical approaches for inertial estimation usually require a lot of assumptions, knowledge of uncertainties, and control over the environment. By employing large amount of collected data, and approximating real motion models with learning algorithms, the methods in this thesis are able to generalise for different objects and estimate parameters in more uncertain environments.

- Provides novel methods for the estimation of an object's inertial parameters in robotics. The methods are using modern machine learning approaches, and are capable of correcting for environmental or execution uncertainties and noise. Their performance is evaluated and the results show accurate estimation. They rely on raw data that can be collected with highly used commercial sensors, making them versatile and usable in different robot systems.

- Identifies benchmarking challenges in the estimation procedure and in robot pushing operations. Since the thesis introduces new concepts in the inertial estimation based on large amounts of data and learning methods, the need for novel datasets and benchmarking protocols between different learning methods and datasets arises and is highlighted.

- Further advances a novel, recently established research field of robot manipulation, task-relevant robot grasping, while showing how the inertial parameters can be used for such purpose.

- Provides simple but fundamental methodologies for new manipulation criteria. These criteria do require knowledge of the robot and object dynamics, increasing the motivation for the presented estimation methods. A robot's dynamics are usually known up to a specific complexity, and the object dynamics can be found with the methods described in the thesis. Thus, the thesis introduces manipulation criteria that can be incorporated to many different robot setups and environments.

- Demonstrates the validity of the approaches in various robot systems. Since they employ fundamental object and robot properties, the theoretical results from the presented research can be used in home, industrial and other environments. Furthermore, the methods presented in the thesis are not dependent on specialised hardware or software, enabling their use in a more system-agnostic way.

1.6.2 Publications resulting from this thesis

The research conducted in this thesis has led to a number of scientific paper submissions in conferences and journals. The papers published as part of the thesis are the following:

1. **N. Mavrakis**, A. M. Ghalamzan. E., and R. Stolkin, *"Estimating An Object's Inertial Parameters By Robotic Pushing: A Data-Driven Approach"*, in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2020.

2. **N. Mavrakis** and R. Stolkin, *"Estimation and exploitation of objects' inertial parameters in robotic grasping and manipulation: A survey."*, Robotics and Autonomous Systems 124, p. 103374, 2020.

3. **N. Mavrakis**, A. M. Ghalamzan. E., and R. Stolkin, *"Minimum Object Internal Force Trajectory Optimization For On-Orbit Dual-Arm Space Robots"*, appeared in International Symposium on Artificial Intelligence, Robotics and Automation in Space, 2018.

4. A. M. Ghalamzan E., **N. Mavrakis** and R. Stolkin, *"Grasp that optimises objectives along post-grasp trajectories"*, in IEEE/RSI International Conference on Robotics and Mechatronics, pp. 51-56, 2017.

5. **N. Mavrakis**, A. M. Ghalamzan. E., and R. Stolkin, *"Safe Robotic Grasping : Minimum Impact-Force Grasp Selection"*, in IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4034-4041, 2017.

6. **N. Mavrakis**, A. M. Ghalamzan. E., R. Stolkin, L. Baronti, M. Kopicki, and M. Castellani, *"Analysis of the inertia and dynamics of grasped objects, for choosing optimal grasps to enable torque-efficient post-grasp manipulations"*, in IEEE-RAS International Conference on Humanoid Robots, pp. 171-178, 2016. (Finalist for Best Interactive Paper Award).

1.7 Thesis structure

The remainder of the thesis is structured as follows:

- Chapter 2 provides a literature review for inertial parameters estimation and their usage in robotics. The motivation for the research is explained from psychophysics papers that describe the uses of objects' inertial parameters in human manipulation. The existing robotics work on inertial parameter estimation is provided and categorised, with emphasis on both classical approaches and state-of-the-art methods. A brief overview of inertial parameter usage is also provided.

- Chapter 3 introduces initial approaches to estimating the inertial parameters of objects in simulation, by using machine learning algorithms. For data collection and testing, a dataset of simulated objects along with physically-realistic inertial parameters is created and presented. The Chapter also touches the field of big-data estimation, by employing a more thorough and sophisticated data collection method, larger amounts of data, different training procedures, and testing on a dataset of real objects.

- Chapter 4 examines the use of the inertial parameters, along with the robot dynamics, to make manipulation tasks more energy-efficient, by minimising the required joint torques to execute manipulations of grasped objects. The work in this chapter represents a novel contribution in the emerging research area of task-relevant grasp selection, i.e. selecting a robotic grasp on an object that makes post-grasp manipulation more efficient. Simulations and experiments with real robots are presented.

- Chapter 5 continues the work in this field, this time by selecting the best grasp for making the post-grasp manipulation safer in case of the robot colliding with the environment. A number of simulations and evaluations with a real robot are presented.

- Chapter 6 combines the post-grasp metrics of the previous chapters with other metrics in the literature, in order to demonstrate the possibility of conflict between the best grasps for different metrics, and the need for a proper task-relevant optimisation method. Simulation experiments are presented.

- Chapter 7 offers a different test case for the inertial parameters of an object, that of an in-orbit space robot which must handle a captured object. The dynamics of the space robot are combined with those of the captured object to ensure manipulation that induces minimum forces on the object. Since this work offers an extension of ideas presented in earlier chapters to a novel application task, only a numerical evaluation is executed.

- Chapter 8 presents a discussion on the thesis, closing remarks, and suggests possible ways of extending the presented work in inventive ways.

Chapter 2

Literature Review

2.1 Introduction

In literature from mechanical engineering, there are numerous existing methodologies on how to estimate the inertial parameters of objects, and they are presented by Schedlinski et al., 2001. Usually, these methods require special hardware and execution (e.g. vibrating tables to measure the oscillation modes), and are ideal to use in more controlled environments, such as some industrial plants. As a result, they can be difficult to transfer in robot environments, as such hardware may not be available. While some basic principles are shared between the two fields, estimation in robotics is usually conducted by approximating the physical laws and relations that include the inertial properties (e.g. accelerating motion) and inference.

The main goal of this Chapter, is to present and categorise methods for estimating the inertial parameters of objects in robotics fields, and showcase their usage in robot grasping and manipulation. The focus lies only on rigid objects, and without any moving or configurable parts. In addition, the chapter does not examine methods of how the robots estimate *their own* inertial parameters, i.e. the dynamics of their mechanical links, but instead the interest is in estimation methods of *other objects*. The presented studies are organised in categories based on the estimation methods and the environments that they are suitable for. The Chapter describes how the estimation is traditionally conducted in analytical, model-based ways, and how state-of-the-art methods are incorporating novel data-driven algorithms. It should also be noted that the focus lies on terrestrial applications, and methods related to space or underwater environments were omitted for brevity.

The chapter is structured as follows: a brief report of how humans estimate and use the inertial parameters is given, followed by an extensive categorisation of the estimation methods. Then, a wide range of usages of the inertial parameters are presented for different tasks.

2.2 Human perception and usage of inertial parameters

Motivation for the use of inertial parameters in robotics comes from the corresponding human perception of object's inertial parameters and heaviness. While there is a large variety of psychophysics works that study this domain, some prominent ones are mentioned to provide some context behind the robotics works.

Humans tend to feel that larger objects are heavier than smaller objects of equal mass. This phenomenon is called the *size-weight illusion*, and has been studied for over a century (Murray et al., 1999, Pick Jr et al., 1967, Amazeen et al., 1996, Flanagan et al., 2000). The illusion is multi-modal (ibid.), and it has been confirmed to appear even when a human has some prior knowledge of the object's size, be it visual or tactile cues (Ellis et al., 1993). While it is difficult to exactly pinpoint the origin of the illusion, the mass and inertia tensor of the object have been shown to influence it (Amazeen et al., 1996, Plaisier et al., 2012).

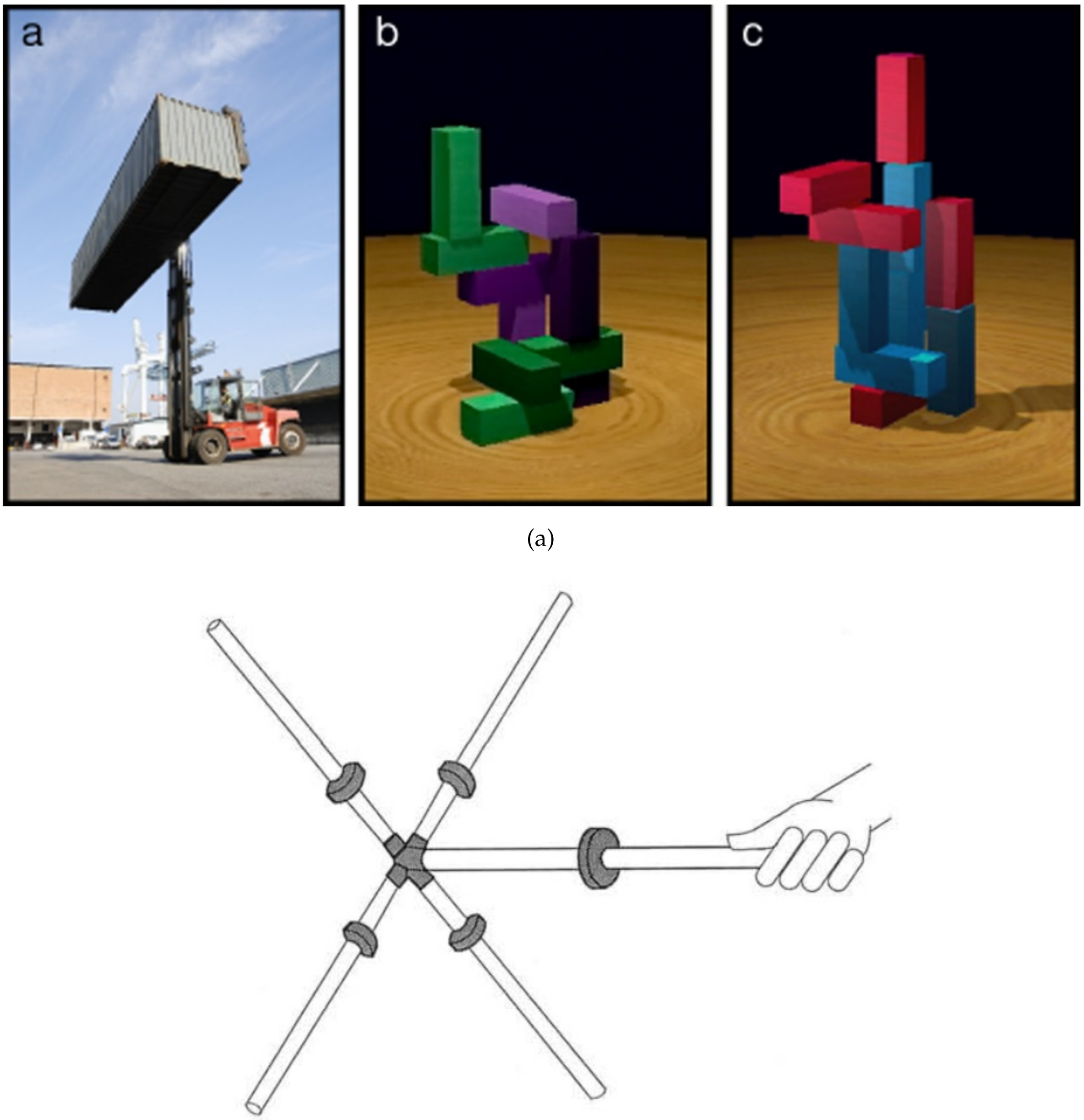
Mass in nature is realised in two ways: gravitational (static) mass and inertial (dynamic) mass. Gravitational mass is felt when an object lies on a person's hand, and inertial mass is felt when the person alters the motion state of the body. Humans perceive mass in one of the two ways, or a combination (ibid.). The two masses are equivalent in physical sense. Nevertheless, Tiest et al., 2010 showed that inertial mass perception of humans is highly affected by the motion type (acceleration or deceleration) and magnitude, and that the dynamic mass perception result can be two times lower than the static. These results are in accordance to previous experiments done in weightless environments (Ross et al., 1982). In a recent approach (Schmidtler et al., 2018), the inertial mass perception of humans that push a trolley was studied, and a linear mixed model was generated from a large number of reference stimuli. This linear mixed model can be transferred to power amplifying systems (such as robotic devices) to assist in human-robot collaboration tasks. Even without interaction, people are able to accurately infer relations between masses of objects in a scene through a mental simulation of the objects' interactions (Hamrick et al., 2016). Bingham et al., 1993 demonstrated that the estimation of a planar object's CoM from a pinching grasp, varies with the object's shape, size, symmetry and orientation.

Finally, there are numerous studies that show how humans feel properties of held objects by using their mass distribution (i.e. elements of the inertia tensor) through a stimulation mechanism of their muscular and tendon system, called *dynamic touch*. Through dynamic touch, people are able to estimate properties such as object orientation (Pagano et al., 1992), position of a grasp relative to the object (Pagano et al., 1994), and object length (Kingma et al., 2002).

Since humans are good at grasping and manipulating objects, it is natural that the study of the perception and role of the inertial parameters in human manipulation, has led to applications in the robotics field. The next sections describe an extensive list of works that showcase how robots can estimate and use the inertial properties of objects.

2.3 Estimation of inertial parameters

In this section, the estimation methods are categorised based on the amount and method of interaction with the object. In total, three categories are presented, *Purely Visual*, *Exploratory*, and *Fixed-Object*. In this thesis, the focus is on Exploratory methods for the parameter estimation process, and specifically robotic pushing of objects. The Fixed-Object category is also highly related to the thesis methodology, because during the parameter exploitation process the robot securely grasps the object and uses manipulation models that are frequently described in this category. Methods that fall within the Purely Visual category are not examined in this thesis, however they are also described both for completeness and to give to provide the overall picture in parameter estimation methods.



(a)

(b)

Figure 2.1: Numerous studies have shown how humans reason about the inertial properties of objects and how they involuntarily use them for perception. (a) Using complex scenes like the pictured, Hamrick et al., 2016 demonstrated that humans are capable of inferring masses of objects by playing a mental simulation of the expected scene outcome, if given sufficient prior information. The inferences have been shown to be quite accurate. (b) A *tensor object*, is a set of cylinder handles with adjustable ring weights. As the rings' positions are adjusted, the object's inertia tensor changes. Studies have used such objects to demonstrate that a persons perception of an object's properties such as length, orientation in space, and grasping point on it, are a function of the object's inertia tensor (Amazeen et al., 1996; Pagano et al., 1994; Pagano et al., 1992).

Table 2.1: Overview of the inertial parameter estimation methods

Method Type	Description	Pros/Cons	Estimated Parameters
Purely Visual	Use of only visual information (object geometry, RGB images, depth images, point clouds, video segments etc.) and possibly existing relationships between visual and inertial properties (density, size-mass formulas in organic objects etc.)	+ Require little hardware + Easier to implement + Require easily obtainable raw data - Require a lot of prior knowledge about the object - Training can be time consuming and require large datasets	Mass, 3D inertial parameters (under assumptions)
Exploratory	Require basic interaction with the object. The applied forces and object motion are measured and the parameters are estimated from physical laws, or learning models.	+ Accurate estimation + Ideal for most autonomous robotics scenarios - Estimation based mostly on analytical models that require controlled environment	2D inertial parameters, 3D inertial parameters by object tilting
Fixed-Object	The object is fixed on the robot's end-effector. As it moves along a trajectory, the end-effector wrenches and joint motions are measured and the parameters are estimated from the dynamic equations in the least squares way.	+ Very accurate estimation + Portability through various robots - Require grasping or fixing of the object, which may not be always possible	3D inertial parameters

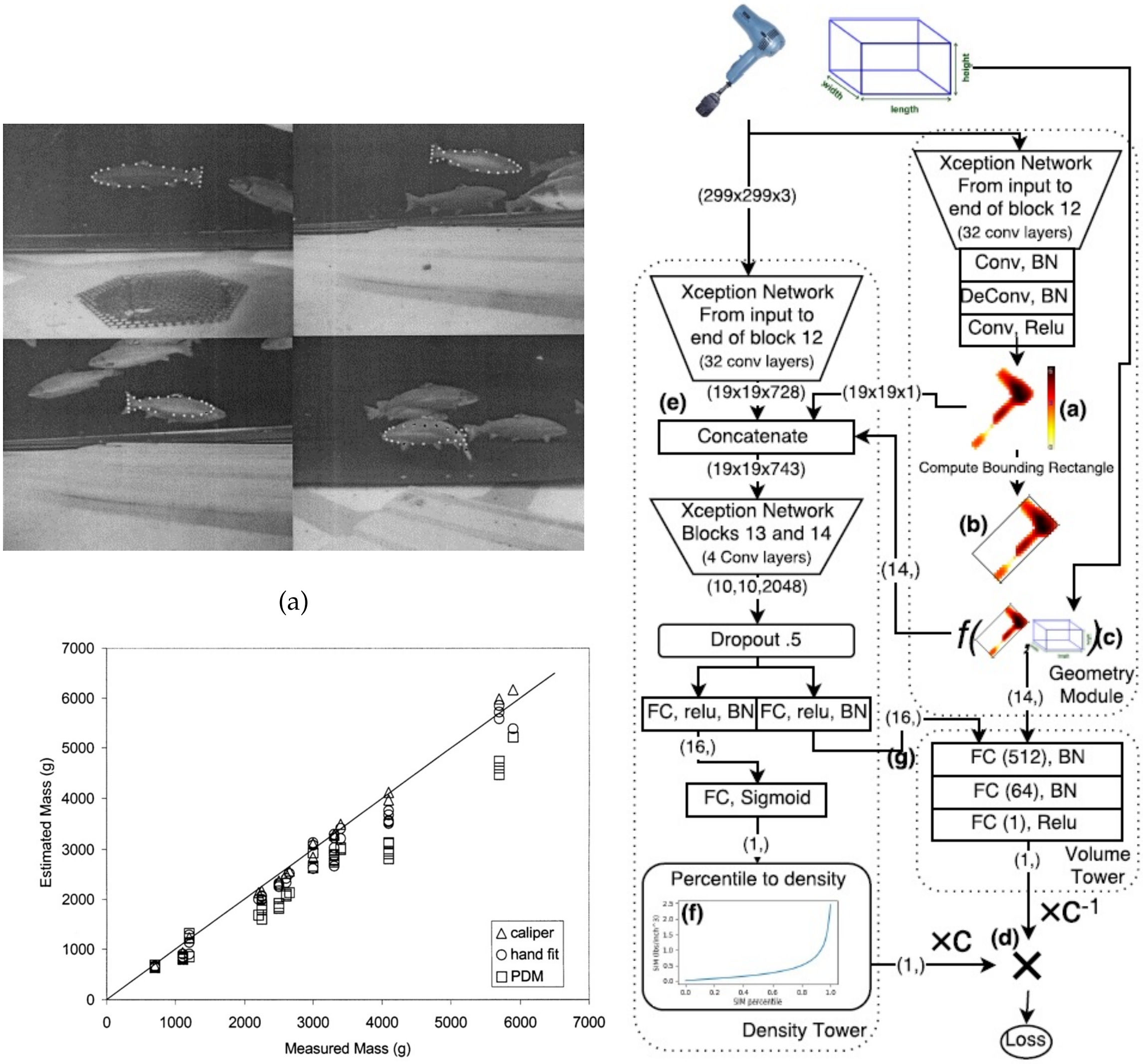
2.3.1 Visual methods

The inertial parameters of an object are a function of their volume, volumetric distribution, and density. While the volume of the object is relatively easy to measure from visual cues, the density distribution throughout the volume is a property that is usually not known, or very difficult to determinate. Furthermore, it can be variable along the volume of the object. As a result, estimating the inertial parameters purely from visual elements seems like an ill-posed problem.

To solve this problem, the first attempts to estimate the inertial parameters had to assume known and uniform object density. Many objects satisfy this assumption, like most natural resources (e.g. wooden trunks, rocks), industrial components (e.g. wooden pallets, metal components, debris), and some household objects (e.g. dishes, chairs). Such an assumption simplifies the problem, and reduces it to extraction of volumetric properties. Chien et al., 1986 calculate the inertia tensor of voxel representation of objects, assuming known and equal mass and size for each block, thus uniform density over the object. The inertia matrix and principal axes are calculated, and the octree/quadtree representation is projected on the axes to be used for recognition of 3D object models. One of the most famous approaches is by Mirtich, 1996, where the author separated a rigid body into polyhedra of uniform density, calculated the mass, CoM, and inertia tensor for each polyhedron, and combined them to get the inertial parameters of the total object. The calculation of parameters was conducted by projecting the 3D volume integrals necessary for the identification of the inertial parameters as 2D surface integrals, and then as 1D line integrals using the divergence, projection and Green's theorem respectively. The author achieved fast computation times, and this method has been widely used in computer graphics.

In the first studies, density was used not as a property to be calculated but merely as a relationship that connects visual and inertial properties. As computing capabilities and image processing techniques improved, the relation between visual and inertial properties could be built from real data and statistical modelling. This is typical when the object is of organic nature, an animal, or an industrial component. Lines et al., 2001 used a stereo camera pair to detect the geometric outline and truss measurements of swimming fish. These measurements have been proven to relate linearly with the fish mass, thus transforming the problem to shape detection. They were able to build a regression model that captures these linear relations, and tested the method with new fish populations with accurate results. Relative results were drawn by Omid et al., 2010, where image processing techniques were used to calculate the volume of citrus fruits and a regression model was fit in their respective masses, and also by Vivek Venkatesh et al., 2015, this time using axis-symmetry of fruits for better image processing. Similar techniques have been employed for estimating the weight of livestock animals such as pigs (Yang et al., 2007) and chicken (Amraei et al., 2017) for monitoring, and an analytical review on weight estimation of livestock animals has been given by Tscharke et al., 2013. In another approach, Bailey et al., 2004 estimated the weight of cup produce objects by extracting their volume in high-speed images, sampling and weighing some items in the produce, and using the measured weights to build a model to connect volume and weight.

The last decade has seen a sharp increase in computing capabilities, as well as sharp growth of big-data learning techniques. In the field of visual estimation of inertial parameters, there has been a shift in the research interest towards using big-data techniques for estimation. A notable case is presentedby Standley et al., 2017. The authors used a large dataset of images taken from objects sold in Amazon.com, as well as the corresponding masses. They also generated a test set of household objects.



(b)

(c)

Figure 2.2: Mass estimation using purely visual features. When a physical connection between visual input (such as size, volume and shape) and mass exists between the object, the problem reduces to visual detection of features. (a) Extraction of visual features in video sequences, that correspond to the shape of the fish (Lines et al., 2001). (b) When the extracted features are matched with measured masses, regression methods can be used to estimate the mass of new fish. A comparison of the estimated and the actual mass of the fish demonstrates the accuracy of such methods. (c) A state-of-the-art learning network for estimating an object's mass from a 2D image (Standley et al., 2017). Two different network modules calculate the two elements needed for mass calculation: volume and density. They get as input an RGB image of the object, a thickness map and a bounding box. The network is able to calculate the object's mass almost as accurately as human perception. Novel learning approaches like this can be used to solve the ill-posed problem of estimating the inertial parameters from visual cues.

They proposed a network architecture where the density of the object is calculated using an RGB image and a thickness mask, and the volume of the object is calculated from the object's bounding box and its' occupancy percentage. The two values were combined, and the mass estimate was provided. They compared the network performance with classical learning algorithms, as well as mass predictions from people in a human experiment. The result was minimum estimation error among other learning algorithms, and close performance to human intuition.

Another recent direction is using big-data for learning the underlying mechanisms of object interactions from visual data, and inferring the mass or relative heaviness of the objects in the scene. Such methods are typically trained by video sequences in simulation, and deployed on real sequences. Learning and modelling the interactions between objects is a very new and rapidly expanding field, and thus documenting all the related work is out of the scope of the thesis. Instead, a number of prominent works is given, mostly related to learning interaction dynamics and include objects' inertial parameters in the procedure. The idea of describing object interactions as learning models originated by Diuk et al., 2008, where an *Object Oriented Markov Decision Process* was presented as a representation of object states and object interaction relations. Scholz et al., 2014 extend this notion, by introducing *Physics-Based Reinforcement Learning*, where the state dynamics and transitions are described to closely represent state-space Newton Euler dynamics. The uncertainties in object inertial and other parameters are modelled as belief over prior distributions. One of the most prominent studies is the one presented by Wu et al., 2015. In this paper, the authors presented Galileo, a learning model that is able to perceive physical properties of objects from video segments. It consists of a generative model that employs a physics engine to simulate object collisions, with an object-based hypothesis space. The generative model labels real video data by calculating the likelihood between the object velocity measurement in the real video and measurements from the physics-based videos. The labelled data are used to train a deep learning network, that can then estimate relative masses between the two objects, as well as other interaction outcomes. This work was further extended (Wu et al., 2016), where the authors employed physical laws to learn more object properties from unlabelled videos.

Chang et al., 2016 the authors introduced the concept of *Neural Physics Engine*. It also uses object-based representations of a scene, namely state vectors with each object's mass, friction and motion variables. A system comprised of an encoder, an interaction neighbourhood mask and a decoder are then able to predict the velocity of an object from previous states. The system was tested in inferring the object's mass based on a prior, and outperformed other well-known prediction methods.

2.3.2 Exploratory methods

As discussed above, a relation between visual and inertial properties (either density or other) is essential for purely visual identification of the inertial parameters. In autonomous robotics scenarios this information is usually not available, as an object of the environment may have non-uniform density, or be composed of objects with different density distributions. In this case, the robot needs to interact with the object in order to extract measurements useful for the estimation process. This category includes the works that require an amount of basic interaction with the object for the estimation, such as poking, pushing or tilting.

The inertial parameters of objects dictate their physical motion under the application of a force, through analytical physical laws. As a result, a classic approach for identifying the inertial parameters is by applying some simple way of contact on the object (pushing, poking, tilting, etc.), measuring the object's motion (velocity, applied force etc.) and calculating the inertial parameters from physical law equations. The first studies in the domain relied heavily on estimating the results through analytical laws of motion, and so required strong assumptions about the robot environment as well as complete control over the interaction procedure. For example, when a robot pushes an object on a surface and measures the motion, it needs to know the friction coefficient between the object and surface, the friction coefficient between the pushing finger and the object, as well as prevent the finger from sliding on the object's surface. A great deal of studies have been conducted for similar estimations, some with strict assumptions and some with more relaxed.

Yoshikawa et al., 1991 determined the centre of friction of an object lying on a surface, by pushing with a mobile manipulator. They assumed the object's supporting surface to be a 2D square grid, and estimated the centre of friction and friction distribution over the grid using applied force and torque measurements. The pushing mechanics that were analysed by Mason, 1986, Goyal et al., 1991a, and Goyal et al., 1991b, suggest that when the surface friction is uniform and isotropic, and the pressure distribution of the object's weight on the surface symmetric, then the centre of friction coincides with the 2D projection of the object's CoM on the surface. As a result, the work by Yoshikawa et al., 1991 can be considered as an estimator of CoM under special circumstances. Krotkov, 1995 presented a series of methods of estimating material properties of objects with robot interaction. Among others, one of the presented methods suggested applying a quick strike on the object, measuring the applied force and its duration, and observing its velocity through a high speed camera. Then, through the impulse equation, the mass of the object can be determined. As the paper mostly described robot perception of material type through striking and listening to the resulting sound, no results were provided for the mass estimation method. Nevertheless, such an approach is plausible, especially nowadays with the existence of high rate force sensors and motion trackers. Fukuda et al., 1999 were able to determine the mass and CoM of large (graspless) objects under limited knowledge of their shape. The object was high enough to be able to tilt. When the object was tilted, a *Gravity Equi-Effect plane* was defined as a plane between the CoM and the contact axis between object and surface. They proposed that since different planes intersect at the CoM, if three gravity planes are found, then the CoM can be estimated. By tilting the object, computing the planes from the object's geometry, and using the distance between finger and gravity plane, they were able to calculate the mass and 3D CoM of the object. They extended this method for the case of round-edged objects (Yu et al., 2004), with similar results.

Tanaka et al., 2003 and Tanaka et al., 2004 estimated the mass of symmetric objects on a table. They applied a force on the visual centroid of the object, that started with low values and increased slowly. By detecting the force value that made the object move and comparing it with the normal force (that is a function of the static friction coefficient and the object's mass), they were able to calculate the object's mass. This work is one of the pioneering in inertial property estimation from pushing, and it required a lot of assumptions (symmetric object, known friction coefficient, non-sliding contact). Nevertheless, the estimation results were very accurate. In a similar configuration, Yu et al., 2005 used a 2-fingered robot manipulator with force sensors to estimate a rectangular object's mass, 2D CoM and rotational inertia, as well as the motion friction coefficient. They applied a set of pushes on different sides of the object, and they measured the finger forces, as well as the motion of the fingers and object. The measurements were used to estimate the inertial parameters of the objects from planar Newton laws in a least squares approach.

The paper by Methil et al., 2006 estimates the mass and rotational inertia of a pushed wheelchair. The authors use a simplified model of the pushchair's motion, and a special 2-fingered steering mechanism with 8 degrees of freedom. While pushing, the steering of the wheelchair was managed by an adaptive controller, that was able to provide stable measurements of the wheelchair's mass and rotational inertia after almost a minute of pushing.

Similarly to the Purely Visual category, the advances of image processing and statistical modelling, along with increased computing capabilities enabled the relaxation of the assumptions for complete control over the environment and interaction. The interaction could be studied as an action that carries noise and uncertainty. Artashes et al., 2013 estimated the CoM and mass distribution of an object, by applying a quick strike on pre-calculated candidate points. The object's point cloud and geometry, as well as the tumbling motion profile were used for the estimation of CoM, rotational inertia and mass distribution.

Franchi et al., 2014 proposed a decentralised approach, that uses a number of mobile robots pushing the object. Each robot pushes on a specific point on the object, and by using the applied forces and geometry of the pushing points that is communicated from each robot through a consensus algorithm, the authors are able to calculate the rotational velocity of the object, and thus the rotational inertia, 2D CoM, and mass. This approach is extended further (Franchi et al., 2015), where the authors measure only the noisy velocity signals of each robot's end-effector. Fazeli et al., 2017a conducted an identifiability analysis for the inertial parameters of objects with known geometry, when the objects are under sticking or sliding contact with a rigid surface. By expressing the motion equation of an object with contact and friction constraints as a complementarity problem, they demonstrated that with the object motion trackable, when the contact forces are unknown the identifiable inertial parameter is the mass-to-rotational-inertia ratio, and under known contact forces both mass and inertia can be identified. They confirmed this hypothesis in a series of experiment where they let objects with known geometry slide in a 2D plane and hit a surface. The formulation of this work was extended thoroughly by Fazeli et al., 2017b, with more objects.

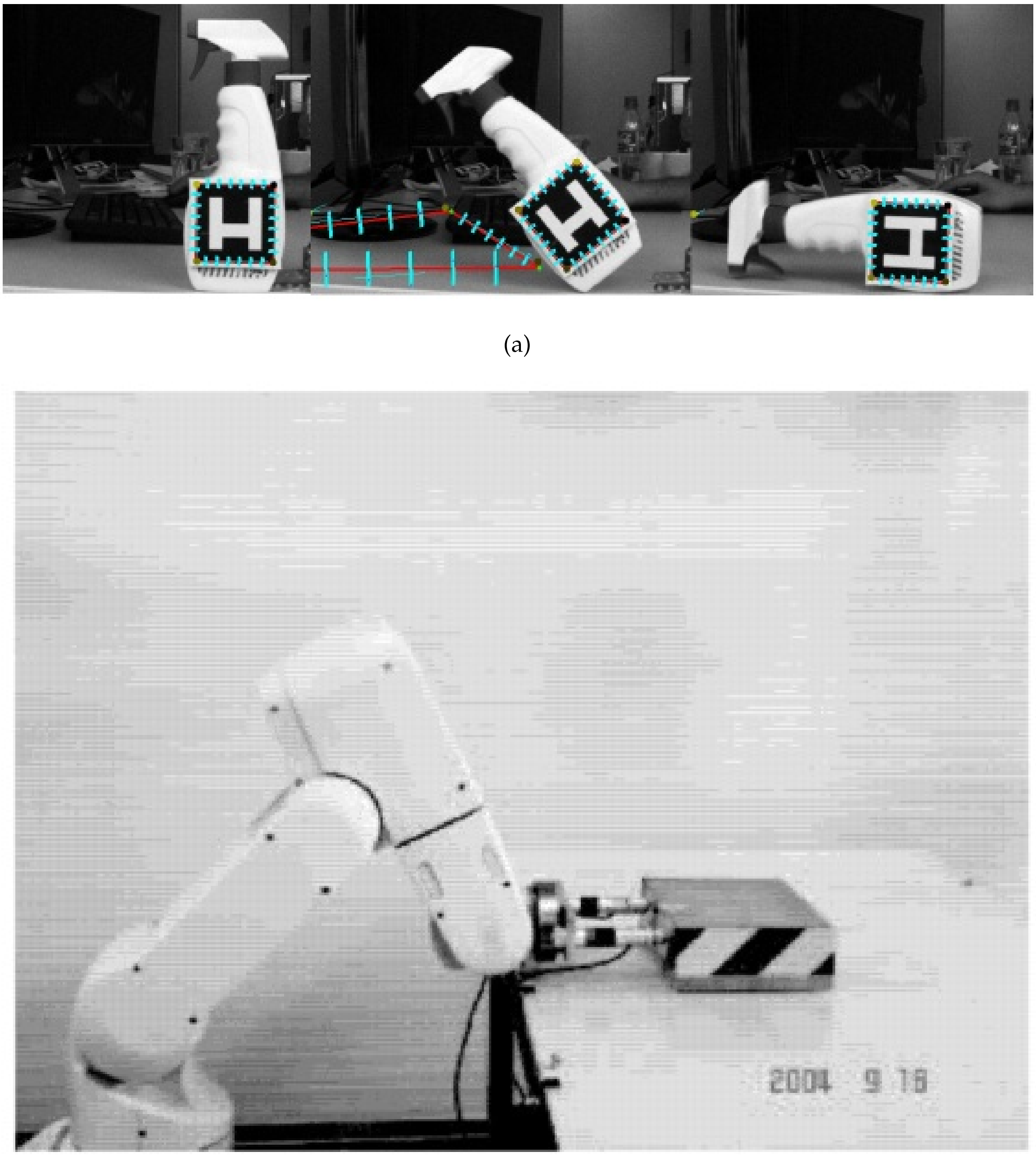
Murooka et al., 2017 estimated the inertial parameters of a large heavy object by applying a set of handling operations recursively (pushing, lifting, tilting), measuring the contact wrench and position as well as the object's pose, and calculating the likelihood in a grid-based Bayesian estimation scheme. An example from tele-operation is shown by Ni et al., 2018, where the authors estimated the dynamic parameters of an object (mass and friction coefficients), to reconstruct a VR simulation useful for teleoperation. They did so by identifying three phases of the push depending on the object's motion, the static, critical and sliding phase. By modelling the contact forces as a mass-damper-spring, using different models for the frictional force in each phase, and measuring the object's motion, they were able to estimate the object's mass and build an accurate point-cloud-based VR environment that describes the scene.

Again, the cutting-edge techniques profit from the advancing of big-data and interaction learning techniques of the last decade. In these techniques, the physical laws are not used directly in the calculation, and instead they are translated as learning models that result from a large number of robot-object interactions. The Physics-Based Reinforcement Learning of Scholz et al., 2014 was further augmented by Scholz et al., 2015. The authors used a mobile robot that pushes objects and measures the applied wrench and object motion. By maximising the log likelihood of the motion over given prior distributions and introducing a penalty term for object's ending state, the authors were able to estimate the mass and other physical parameters of the object. Other examples are the works by Zhu et al., 2016 and Zhu et al., 2017. In these two works, the authors generated objects with different dynamic parameters (mass and friction coefficient) in the Bullet physics engine, as well as motion profiles when a force is applied on them. When they applied a force on a real object, they used Bayesian optimisation and Entropy Search to identify the simulated object whose motion closely matched the real object's motion. They then used the motion of the resulting simulated model to predict the motion of the real object, with high accuracy. While the authors primarily focused on the prediction part and did not provide results for the mass estimation part, their work is one of the first to employ big-data methods on the inertial parameter estimation problem. Finally, Novin et al., 2018 used a Bayesian Regression Model to learn the inertial parameters of a hospital walker, by tracking the motion of a real robot pushing the walker in 39 trials. They used the learned model for manipulation planning, prediction and control of the walker motion, achieving low errors.

2.3.3 Fixed-object methods

The third estimation category includes the methods where there is a fixed connection between the robot system and an unknown load. This connection manifests in many forms, i.e. a robot that rigidly grasps the object, or has it otherwise attached at the end-effector. Studies in this category are inspired from classical dynamic model identification techniques in robotics, where a robotic arm executes excitation trajectories and gets torque measurements from the joints and force measurements from wrist sensors. The model can then be identified by rearranging the dynamic equations and solving them in a least squares way. Examples of such work include the works by Khosla et al., 1985, Swevers et al., 2007, Gautier, 1997, and a survey paper in Wu et al., 2010. Only a couple of these studies are mentioned, as the main focus is the determination and usage of the inertial parameters of *other objects* and not the robot itself, but the techniques are similar in both cases.

Unlike the other categories, in this category all the 3D inertial parameters of the object are usually calculated, due to the capability of object motion in 3D space. Traditionally, these methods were preferred and optimised for robots operating in industrial environments. One of the first works in this category was the study by Olsen et al., 1985, where the authors proposed the equations for estimating the dynamics of a robot with rotary joints as well as the dynamics of the load, by using measurements of joint angles, velocities, accelerations, torques, and force and torque applied on the load. They proposed test motions of only one joint at a time, and proposed the equations for estimation, without conducting an experiment. Atkeson et al., 1985



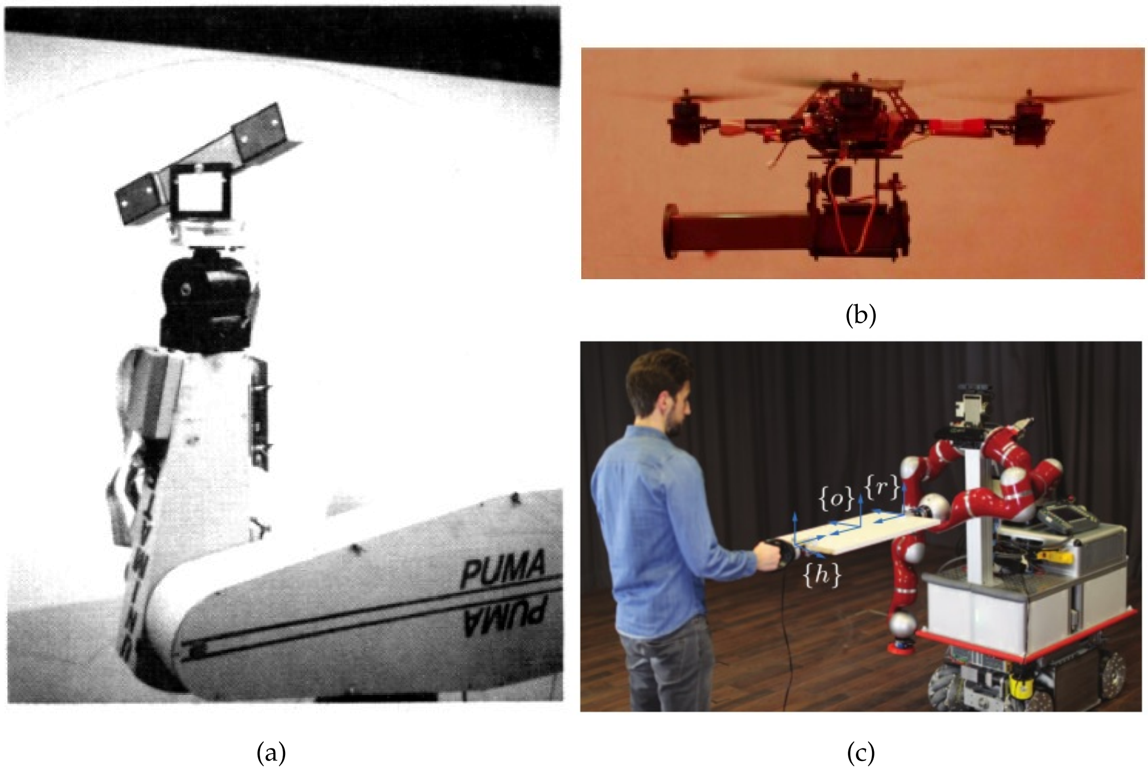
(a)

(b)

Figure 2.3: Examples of exploratory methods. By applying a simple action on the robot and observing its motion, the inertial parameters can be calculated. (a) The mass, CoM position and mass distribution can be calculated by striking an object, measuring the applied force, and tracking its rotational motion profile. Image by Artashes et al., 2013. (b) Similarly, the object can be pushed by a robot and the inertial parameters can be estimated by planar motion laws. Image by Yu et al., 2005.

provided a method for estimating the parameters by using only measurements of force and torque on a wrist, as well as linear and angular position, velocity and acceleration of the sensing frame. They conducted two experiments with two different manipulators, and concluded that the estimation is more accurate when the measurement signals are less noisy. Dutkiewicz et al., 1993 identified the inertial and frictional parameters of both the robot and the load by employing integral dynamic model, which is a function of only the joint and load positions and velocities, and not accelerations. The identification was done by measuring force and torque on the wrist, as well as joint positions, velocities and torques, and applying a least squares estimator between measured and calculated dynamics. A similar approach was taken by Swevers et al., 2002, where the authors determined the 3D dynamics of a load using a maximum likelihood method, after getting noisy joint torque measurements and noise-free joint motion measurements. In addition, they were able to identify uncertainties in the robot dynamics, such as motor and transmission losses. They tested the model in an industrial manipulator with accurate results. Kubus et al., 2008 used force and torque measurements from a sensor on the wrist of an industrial robot, as well as measurements of angular velocity, and linear and angular accelerations, to estimate the inertial parameters of the load. They used excitation trajectories and applied a Total Least Squares method for the estimation, which enabled them to make on-line predictions on the parameters. The method was able to provide an accurate estimation in as little as 1.5 seconds. Chinthaka et al., 2017 developed a system of estimating an object's mass and rotational inertia, while held by a robot and being moved in a pendulum-type oscillation way. Dong et al., 2018 developed a method similar to the ones by Atkeson et al., 1985 and Kubus et al., 2008, with the difference that they used the torque difference between motion based on the calculated dynamics without the load and actual motion. To separate the effect of the mass, CoM and inertia tensor on the motion, the authors used 3 different excitation trajectories. Finally, Farsoni et al., 2018 used an Extended Kalman Filter to calculate the motion accelerations of an industrial robot with a load, and a Recursive Total Least Squares method combined with wrist force and torque measurements for the identification of the load parameters.

Recently, here has been a trend of taking robots out of the factory cages and operating them in close proximity of humans, as well as in outdoor environments. In addition, unmanned aerial vehicle (UAVs) manipulation is an emerging research field. These trends have shifted the focus of fixed-object inertial estimation from a single industrial arm, to UAVs, multiple robots, or even human-robot interactions. The core of the estimation process remains the same i.e. measurements of motion and force signals and model-based estimation of the parameters. Mellinger et al., 2011 designed an unmanned aerial vehicle (UAV) with two customised grippers, and provided the Euler equations of motion for the UAV holding an object. They estimated the mass, inertia and CoM during both undisturbed hover and motion with disturbances. Similarly, Lee et al., 2016 described the combined dynamic equations of a hexacopter with a 2-DOF arm. They demonstrated how the inertial parameters of an unknown load are incorporated in the total system dynamics, and how the the inertial parameters of the object can be found from the known dynamics of the robot, and the motion of the hexacopter. They built an adaptive controller that can control the arm's end-effector on a desired trajectory, while estimating and compensating for the unknown load. A different approach was taken by Corah et al., 2017, where the authors created an algorithm for the estimation of an object's mass and 2D CoM, by deploying a number of UAVs to lift the object. After iteratively deploying UAVs on the object and taking



(b)

(a)

(c)

Figure 2.4: Examples of load dynamics identification. The load is grasped or otherwise attached to the robot and it is moved in the workspace. Motion, force, torque and and robot joint measurements are taken, and the parameters are estimated by using the equations of object motion and the robot dynamics, usually in a least squares way. (a) For industrial applications, the load is fixed on the manipulator. The robot executes specific excitation motions that provide more informative measurements. By measuring the joint angles, positions, accelerations and torques, as well as wrist forces and torques, the load parameters are estimated from the total dynamic model. Image by Atkeson et al., 1985. (b) Similar approaches are followed in aerial manipulation, where the transferred load is firmly grasped and moved by an unmanned aerial vehicle and measurements are taken. Image by Mellinger et al., 2011. (c) In the case of human-robot collaborative transportation, the robot mimics the human motion and at the same time estimates the parameters by measuring wrist forces and torques and using the object's equations of motions. Image by Cehajic et al., 2017.

taking force measurements, an estimate was given by maximising the divergence between the force measurements and the parameters that produce the measurement. Marino et al., 2017 used three mobile robots applying coordinated wrenches on an object, to estimate the mass, CoM, and inertia tensor of the object. They applied coordinated motions that resulted in pure translations or pure rotations on the object, and calculated the parameters with the grasping and motion equations of the composite system. The result was used to manipulate the object with minimum squeezing force. Estimation with coordinated transferring was also the subject of Cehajic et al., 2017. The authors estimated the inertial parameters of an object being handled by a robot and a human, by expressing the inertial parameters as a function of the robot and object motion. They then projected the robot motion used for identification in the

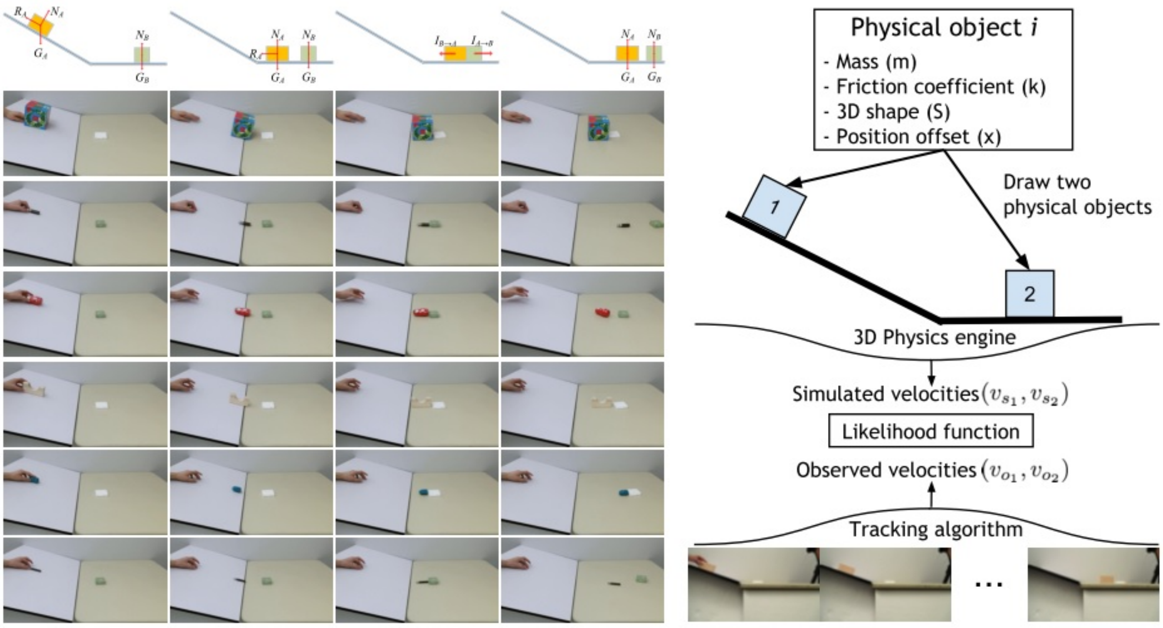


Figure 2.5: Recent estimation methods use large amounts of data to model dynamic interactions between objects. They then learn the mapping between video frames and motion, and use this mapping to infer the object's mass and friction coefficient. The Galileo system shown in the figure is able to infer the relative mass of two objects from a video of their collision and a comparison with a collision from simulation. Image by Wu et al., 2015

null space of the grasp configuration around the object. This way, they were able to control the robot to match the desired human motion, and calculated the inertial parameters without disturbing the human task.

Finally, machine learning has not been used extensively for this category, mostly due to the increased accuracy and efficiency of the model-based methods. In one of the first approaches to include robot learning in the procedure of load estimation, Petkos et al., 2007 simulated a 3-DOF manipulator to follow sinusoidal trajectories. They measured the joint positions, velocities, and accelerations, and used Locally Weighted Projection Regression to learn a dynamic model. By repeating this process with a number of simulated manipulators with reference loads attached, they were able to estimate the dynamics of new loads. Angel et al., 2016 conducted a simulation to map changes in a held payload's mass, to variances in the joint torques. After simulating for different masses and motions, a set of clustering algorithms were compared to separate the payload variations into classes. These classes can then be used to distinguish between different masses that are fixed on the robot arm.

2.4 Exploitation of inertial parameters

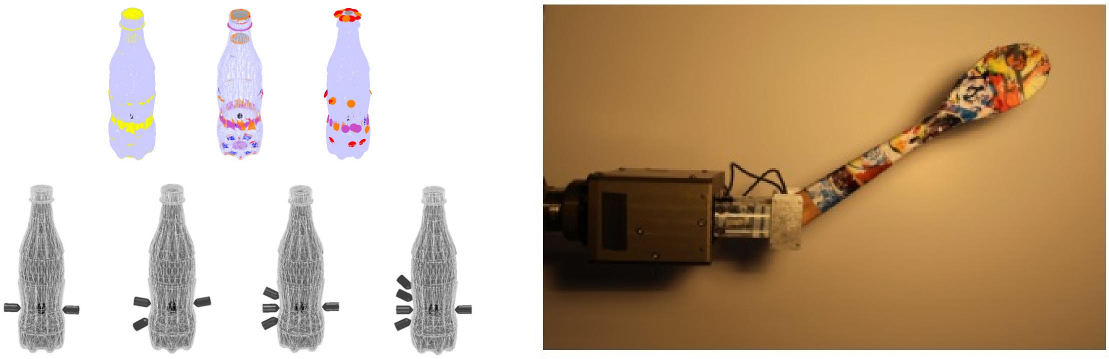
In the last sections we described how state-of-the-art learning methods attempt to directly learn dynamic models from object motions generated by physics engines. This eliminates the need to estimate the object's inertial parameters, as the models that dictate the dynamic motion can be learned and represented in an intelligent way (e.g. a neural network or a statistical representation), and applied to many different objects without retraining. However, there are many cases where the value of the inertial parameters need to be determined numerically.

As discussed in previous sections, humans both intuitively and actively use the inertial parameters of objects to perceive their other properties, generate fixed and stable grasps on the object surface and facilitate manipulation tasks. For example, when a person encounters a heavy box, they realise that they must use both arms to lift it and transfer it. By nature, the person will place their hands on the object in an antipodal way, to provide support and stability, as well as put less strain on their arm muscles. When they transfer the object, they will keep their posture upright to minimise back strain. In this case, the person has conducted grasp and manipulation planning influenced by the object's inertial parameters. As humans are very effective in manipulating objects, it is natural that this ability to exploit the objects' inertial parameters should be transferred to robots as well.

This section presents a large number of papers that demonstrate how the inertial parameters of objects are incorporated in algorithms to make robot grasping and manipulation more efficient. It is not easy to categorise these papers, as disproportionate amounts of work exist for different manipulation tasks. Many works from the fields of grasp planning, manipulation planning, and controller design are mentioned. The goal of this section is to show how the inertial parameters can be used in different grasping and manipulation tasks, and so justifying the need to estimate them in the first place.

The object's inertial properties can be used to augment the dynamic model of a robot. Indeed, when a robot rigidly holds an object, the dynamic model changes according to the dynamics of the object and the grasping point on the object. One of the first works to provide analytical formulations for this property was by Khatib, 1987. The author projected the dynamics of the object and the dynamics of the robot on a specific point in space, called *operational point*, and showed that the total dynamics can result from a simple addition of dynamic matrices. The formulation was also provided for the case of multiple manipulators handling an object. The results were extended by Khatib, 1995, where the dynamics of robots and objects were used to formulate other manipulation criteria such as reflected load on a manipulator, effective mass and inertia in manipulation movements and dynamic consistency. In addition, the augmented object model was used by Chang et al., 2000 for branching configurations in which two robots with two arms each manipulated an object.

Most grasping algorithms in the literature exploit the geometry of objects to plan for and execute grasps. Nevertheless, the inertial parameters are essential in the analytic formulation of grasping, as both the grasp force mapping and the closed-form equations of motion require the dynamics of the object (Murray et al., 1994). As a result, a lot of studies have used the inertial parameters of objects to solve problems in robot grasp synthesis, planning and quality evaluation. Montana, 1992 provided a grasp stability analysis that incorporates both spatial and contact stability. By including the inertial parameters of the object, the author developed a matrix, the eigenvalues of which are a metric for the grasp stability. In a similar manner, Bruyninckx et al., 1998 developed a stability grasp metric by dividing the grasp stiffness matrix and the object's inertial matrix. Again, the eigenvalues of the division were related to the stability of the grasp. Borst et al., 2004 developed a grasp quality metric by using the



(a) (b)

Figure 2.6: The inertial parameters of objects are used as a property that makes robot grasping and manipulation algorithms more efficient. (a) Some grasp synthesis algorithms use the object's inertial parameters to generate stable and minimum-disturbance grasps. Image by Lippiello et al., 2013 (b) The inertial parameters are also incorporated in the dynamic model of the robot for designing controllers. In this image, the slippage of the object is a function of the inertial parameters, and an accurate controller is designed. Image by Viña et al., 2015.

Grasp Wrench Space, namely the set of wrenches a grasp can counterbalance, and the *Object Wrench Space*, namely the set of wrenches an object will produce while moved for a task. For a given grasp, the scaling between the two spaces acts as a grasp quality measure. To construct the Object Wrench Space, the mass and CoM of the object are required. Lopez-Damian et al., 2005 generated antipodal 3-contact grasps on an object, by using the CoM, principal axis of inertia, and local object geometry. They argued that this type of antipodal grasp, where a grasping axis is aligned with one of the object's principal axis of inertia is intuitively more robust to gravity and accelerating forces. Similarly, Lippiello et al., 2013 generated grasps on object models, by identifying *minimum inertia* regions on the object. These regions were defined as those where the contact point friction cones contained the object's CoM. By also exploiting the object's surface properties and the finger area the authors produced force-closed grasps. These results were also extended by Lippiello, 2015, where the author selected the minimum inertia regions that were anthropomorphic, i.e. the finger contact normals were opposing or triangular-shaped. Buchholz et al., 2014 created an efficient algorithm for the bin-picking problem for estimating the object's pose after it has been grasped, and the pose was determined by the on-line estimation of the objects principal axis of inertia and CoM. Masuta et al., 2015, Masuta et al., 2016b and Masuta et al., 2016a described the *sensation of grasping* of a robot, as a fuzzy inference method of the object's approximate size based on the object's principal axes of inertia. The sensation of grasping was used for an integrated robotic perception system that clears a table. Kanoulas et al., 2018 used a robot that grasps an elongated object from a graspable position close to the object's visual centroid by slightly lifting the object and measuring the torque signal on the robot's wrist, they were able to iteratively adapt the grasp towards the inertial CoM of the object.

Finally, other examples of inertial parameters exploitation include works by Kubus et al., 2007, where the authors used the inertial parameters in a bin-picking problem, to recognise objects and their poses for selecting good grasps, and Murooka et al., 2014, where the authors used the inertial parameters of heavy objects to define a strategy decision system for different manipulation types (push, lift etc.).

The inertial parameters of objects are included in in-hand dexterous manipulation and control studies. Example applications include object re-grasping, re-orientation and control of finger slippage. Cole et al., 1988 and Cole et al., 1992 provided the kinemodynamic modelling and controller design of a robot grasp on an object with rolling and sliding contacts respectively. For modelling and control they assumed knowledge of the object's inertial parameters. Viña et al., 2015 and Viña et al., 2016 further use gravitational torque and visual tracking with known object's inertia, to control the in-hand planar motion of an object while grasped by a pinch grasp. Chavan-Dafle et al., 2015 and Kolbert et al., 2016 examined the problem of using the environment contacts to manipulate an already grasped object, defined the dynamics of different prehensile pushing primitives by using the object's inertial parameters, and conducted experiments using simple objects. Shi et al., 2017 described a finger motion planning scheme for in-hand manipulation of objects. They also used the inertial parameters of the object incorporated in the planning equations.

Finally, the inertial parameters of objects are necessary to develop algorithms for motion and control of dual or more manipulator robot systems. In parallel with the robot finger grasping case, the object's inertial parameters are incorporated in the dynamics of the robots to generate closed-loop models. The closed-loop models are used for solving manipulation problems such as control, trajectory generation, and others (Hayati, 1986), (Walker et al., 1991), (Schneider et al., 1992), (Yoshikawa et al., 1993), (Ren et al., 2016). Referring to all literature that uses multiple arms and object models is out of the scope of the thesis, and an extensive survey has been conducted by Smith et al., 2012.

2.5 Discussion

The separation of estimation methods in different categories was conducted based on the nature and amount of robot-object interaction. Each type of robot-object interaction requires different data types for measurements. As a result, each estimation category is best suitable for specific environments.

Purely visual methods work best assuming some prior object knowledge. That makes them ideal in environments such as industrial plants, where components of known materials need to be inspected. By using only the object's geometry acquired by cameras or depth sensors, the system can extract the inertial parameters of the component and forward them to the next in line production steps. From there, the components can be handled (packed, transported etc.) more efficiently. Another possible use would be in robotic exploration, e.g. planetary, where the material composition of surrounding obstacles in a planetary body may be determined from spectral measurements, and the inertial parameters from geometry. The purely visual methods are, under strong assumptions, able to calculate all the inertial parameters of the object. They also require only visual data (images,depth maps etc) and limited equipment to operate. Nevertheless, they lose accuracy in case the object's density distribution is unknown.

Exploratory methods are suitable for autonomous robotics, where a robot does not have any information about the environment and surrounding objects. The estimation methods offer minimum levels of interaction and this makes them ideal to use in dangerous environments such as nuclear plants and disaster sites. A shortcoming of most methods in this category is the limitation of estimating only the 2D inertia parameters of an object. The equations of motion show that the object needs to be moved along all 3 coordinate axis to estimate its inertia tensor. By tilting along one axis or pushing on a planar surface, it it impossible to extract all the inertial parameters without prior knowledge. Nevertheless, mass and rotational inertia can be estimated and used as prior knowledge for uniform-density objects.

Fixed-object methods are often used in industrial plants to calculate the inertial parameters of heavier payloads. They require firm grasping or otherwise fixing of the object, which makes them ideal for controlled environments. They are quite accurate and able to estimate all the inertial parameters. Usually they require force sensors on the robot joints or the wrist.

As shown, the usage of the inertial parameters has been fundamental over robotics research in analytical robot grasping and manipulation. The algorithms presented need numerical values of the inertial parameters in order to work. This means that, despite the tremendous progress of the presented motion prediction and interaction learning models that learn dynamical models without explicitly estimating them, there is still a need for methods that can calculate the exact values of the inertial parameters.

Chapter 3

Data-Driven Simulated Estimation by Robot Pushing

3.1 Introduction

As stated in the previous chapter, estimating the values of an object's inertial parameters is very important because it leads to design of more effective manipulation and grasping algorithms. The methods of estimation are also highly dependent on the environment, equipment, and prior object knowledge. Since in this thesis the objective is to develop algorithms that operate in unknown and potentially hazardous environments, the focus was placed into exploratory methods that naturally assume little or no knowledge over the handled object, and interact with it carefully and effectively. In particular, the preferred method of estimation is applying a small push on the object and observing its rotational motion. The applied forces and motion of the object are measured, and the estimated parameters result from a model that correlates the applied action with the resulting motion.

According to the analysis of the previous chapter, analytical methods of estimation provide very accurate results, but tend to rely heavily on strict assumptions about the object and environment, as well as control over the uncertainties in robot motion and object interaction. Such methods employ models based on physical laws and relationships between measured sizes to perform the estimation. The main issue with these models is that they tend to not incorporate uncertainties in the pushing procedure. Examples of these uncertainties include sliding of the robotic pusher on the pushed object's surface, sliding and sticking of the objects bottom part on the underlying surface, uncertainties over the object-surface and pusher-object friction coefficients, and measurement noise in the force sensing and motion tracking sensors. For this reasons, they have limited robustness along various environments and robotic setups, as well as limited portability on estimating the properties of objects of different geometries. In other sub fields of robotic grasping such as grasp synthesis, the inability of analytical based methods to cope with all these uncertainties has been tackled with the introduction of statistical and machine learning methods, i.e. data driven methods. These algorithms gather large amounts of data in the form of action-observation pairs. They then train a learning model on part of the data and test it on the rest of the data. Their performance over the dataset can be checked according to a pre-defined metric. The generalisation over new data can also be examined. Data-driven methods tend to result in slightly higher errors, but higher generalisation capability. They deal with uncertainties by largely introducing them in the model during the learning procedure.

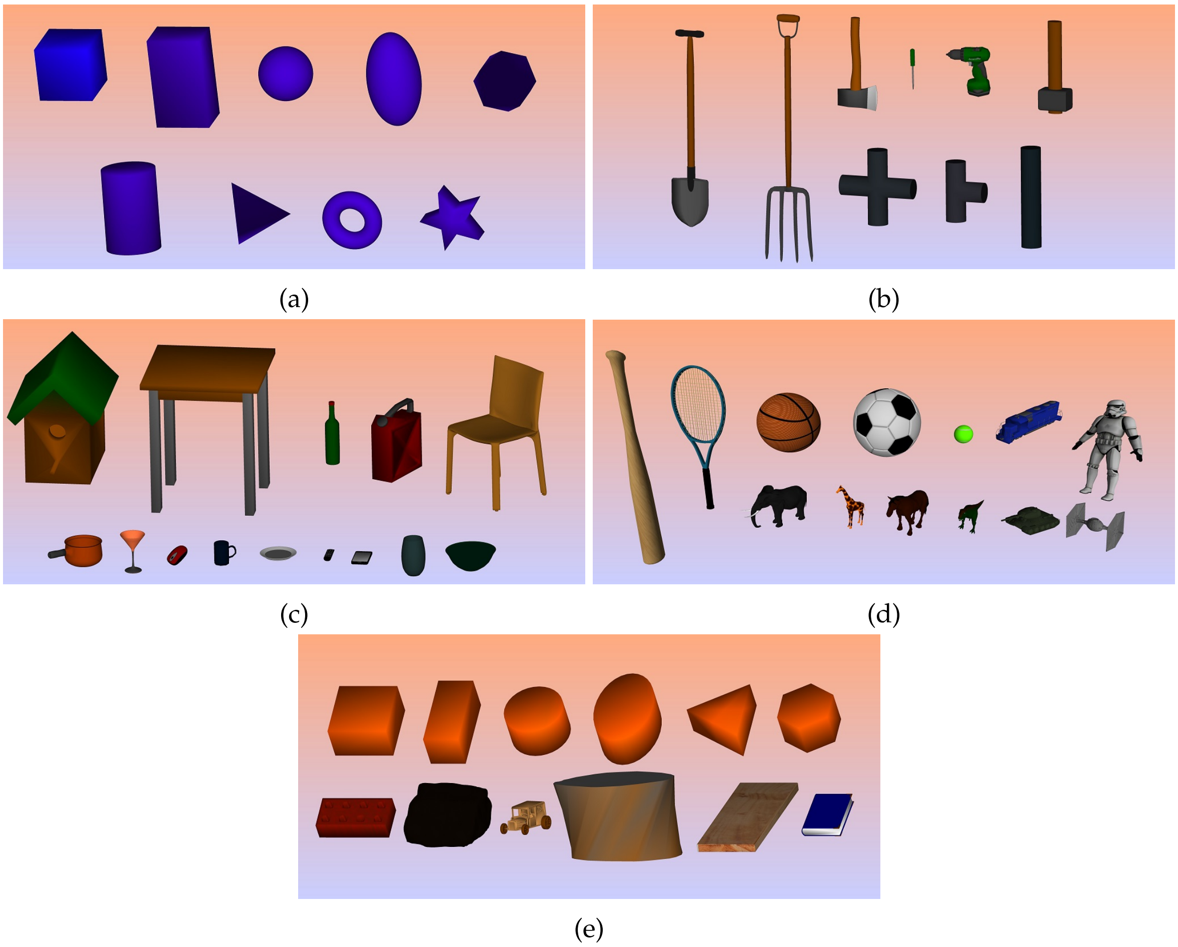
The robot algorithms developed in this thesis are operating in unknown environments with objects of various geometries, and so there is a need for sufficient overall performance and generalisation to novel objects. As a result, the focus of this thesis is in developing data-driven algorithms, that estimate the inertial parameters of objects by robot pushing. As stated, data-driven exploratory estimation methods have only been recently studied, and a number of challenges need to be addressed compared to more classic, analytical approaches. This further showcases the novelties of the thesis, and enables it to fill newfound gaps in the literature.

The main contribution of the chapter is to describe all the stages of a data-driven pushing method, from beginning to end. The stages consist of the testing object datasets, the data collection methods, the features that are extracted, the learning algorithms and the evaluating measures. The goal is to provide an interpretation of the selected approaches, as well as describe the lessons learned during their application and related experiments.

3.2 Simulated object dataset

In robotics research, machine learning algorithms require large amounts of data to train the robot and assess its performance. In order to extract large amounts of data for the algorithms, as well as provide testbeds for evaluation, many recent studies have created datasets of different objects. These datasets are increasingly common in robot grasping and manipulation research, and they serve as comparison bases across different studies. Such works for object recognition and manipulation include the ones by Calli et al., 2015; Kasper et al., 2012; Wohlkinger et al., 2012 and Pokorny et al., 2017. A dataset that is related to robot pushing is the one in Yu et al., 2016. It includes a small number of objects with basic shapes and masses together with a large number of planar pushing and interaction profiles applied to those objects.

In order to be able to develop algorithms for estimation and usage of objects' inertial properties, there is a need for an object dataset with various ranges of masses and mass distributions. However, it is very difficult to create such a dataset using real objects, because their inertial parameters may be hard to calculate of find experimentally. Instead, a good starting point would be to use simulated objects, for which the parameters can be easily calculated. In this Chapter, a dataset of simulated objects is provided. It includes 57 different realistic objects in 5 categories together with their 3D meshes, mass, inertia tensor and centre of mass. The 3D models were uploaded in Meshlab, where the inertial parameters were calculated. Meshlab calculates the parameters by applying the algorithm described in Mirtich, 1996, assuming uniform density equal to 1, that is the mass of the object is considered equal to its volume. To assign desired inertial parameters to an object, a desired mass must first be defined. The inertia tensor elements are then multiplied by the division of the new mass to the object volume. The CoM remains equal to the volumetric centroid, as the density remains uniform. The object models can be scaled and combined to generate new objects with new inertial parameters. The dataset is suitable for running simulations of pushing, manipulation, and grasping experiments. In general any interaction between the robot and the objects could be simulated. A thorough literature review in



(a)

(b)

(c)

(d)

(e)

Figure 3.1: The inertial object dataset has five object categories: (a) The *Primitives* category includes primitive objects that can be used separately or combined in many experiment types; (b) The *Tools* category includes heavier objects that suited for tool manipulation experiments; (c) The *Home* category includes common objects that are easy to find, experiment and compare with real objects; (d) The *Toys* category includes sports equipment and animal figures with complex shapes, suitable for grasp generation experiments; (e) The *Pushing* category includes objects that are ideal for pushing experiments due to their flat and elongated shapes. This dataset includes 57 objects and contains 3D mesh models and inertial information for each object. The object dimensions and masses were selected so that the dataset objects are physically similar to real world objects. The mass, CoM and inertia tensor have been calculated with Meshlab. The variety of object categories allows generation of objects to be used in learning algorithms, both for training and evaluation.

robotics-related object datasets showed that this is the first object dataset suitable for inertia-based manipulation, grasping and pushing benchmarking.

The presented inertial dataset is tested in Gazebo 2.2 and ROS Indigo. It can be used with any other physics simulator supporting URDF[1], such as V-REP and MuJoCo. The dataset includes 3D models of 57 objects together with their mass, CoM,

[1] The Universal Robotic Description Format (URDF) is an XML file format describing the kinematics and dynamics of all the elements of an object, mechanical structure, or chain of objects.

and inertia tensor expressed in a frame attached to the CoM. The parameters of each object (size, mass, friction and the inertia tensor) are adjustable, and changing them is useful for combining different object models and creating new objects of various complexity. The inertial properties of every component in a combined object can be determined numerically, thus forming an object with non-homogeneous mass distribution. Thorough description of the dataset form and object categories is given below.

3.2.1 Technical characteristics

Each object is represented as a URDF file, that defines the geometry (3D mesh), collision shape, inertia tensor, mass, CoM and friction coefficient. The inertial properties were calculated in Meshlab, which makes use of the algorithm proposed by Mirtich, 1996 to estimate the inertial properties of a mesh using uniform and unit density $d = 1$. The volume and mass values are identical for this calculation. However, the inertial properties of a non unit density object can be easily computed by scaling the obtained parameters from Meshlab by the desired density, allowing the creation of many models and shapes.

3.2.2 Object categories

The object categories are described in this section and example uses of each category are given (Fig. 3.1). It should be noted that these uses are indicative, and objects from each category can be used in other types of experiments as well.

- *Primitives:* Includes models of primitive shapes, such as sphere, cube, cylinder etc. These objects can be used in big-data experiments similar to e.g. Wu et al., 2015, as they are easy to use, scale and combine. Their inertial properties are also more straightforward to scale.

- *Tools:* Includes tools that allow a robot to learn manipulations of such an object in a realistic experiment. They have medium to large mass values. Their mass distribution and handles makes them ideal for manipulation and tool usage simulations.

- *Home:* Includes everyday objects found in homes such as mugs, bottles and books. They have mostly lower mass values, and can be used in grasping, recognition and manipulation experiments. Since they are common and easy to find in the real world, the simulations conducted using these models can be easily verified in real-robot experiments.

- *Toys:* Includes toy figure objects and basic sports equipment, such as toy cars, animal figures and balls. They also have mostly medium masses. They are designed to have peculiar geometries, so that they can be used in grasp generation and grasp evaluation experiments, e.g similar to Lopez-Damian et al., 2005.

- *Pushing:* Includes models that are suitable for pushing experiment on a table because of their flat and elongated shapes. The frictional properties can be easily changed in the URDF file, so that different materials can be simulated for more physically-realistic pushing. The models are designed to be placed with their larger surfaces down (or the wheels in the case of the car), and any face can be used for pushing.

3.3 Data-driven estimation of mass and inertia

A first attempt to approach the pushing estimation problem is described in this section. The simulated dataset is used for that matter, demonstrating its usefulness and possibility for generalisation. Overall, the section describes a simulated method for a controlled robot push, that results in estimations of mass and inertia (I_{zz}) of an object. A set of input-output pairs are generated that represent push and corresponding motion outcomes. Then, a non-parametric regression method is used to estimate the mass and rotational inertia of objects. The regression model is fed with a feature vector $\boldsymbol{v}$ from measurement signals, namely force, torque and object displacement and define a corresponding output vector $\boldsymbol{d} = [m, I_{cm}]$. The regression model is trained from the acquired data to correlate pairs of observed features and the inertial properties $\{\boldsymbol{v}, \boldsymbol{d}\}$.

3.3.1 Problem formulation

For modelling the push environment, three coordinate frames are considered: (i) $O_{cp} \in SE(2)$ attached to the end effector of the robot pushing an object, (ii) O_{cm} attached to the center of mass of the object (CoM) and (iii) $O_w \in SE(2)$ as a reference frame where $SE(2) = \mathbb{R}^2 \times SO(2)$. Fig. 3.2 shows a robot pushing a rectangular object (purple) on a table (white) with its end effector (red). The object mass and rotational inertia w.r.t. O_{cm} are m and I_{cm}, respectively. The friction coefficient between the surface of the object and the table is μ. For simplicity, the static and sliding friction coefficients are assumed equal. In addition, the friction coefficient between the point contact between the end effector and the object surface is noted with μ_c, where x_{cp} is normal to the contact surface. The robot applies a force $\boldsymbol{F} \in R^2$ at the contact point O_{cp} having a scalar projection onto x_{cp} and its normal for a total duration of T. At each time instance $t \in [0, T]$, the equations of motion of the object moving on the table are written as follows:

$$\boldsymbol{F} - \mu m g \,\text{sgn}(\boldsymbol{u}) = m\boldsymbol{\alpha} \tag{3.1}$$

$$\boldsymbol{F} \times \boldsymbol{r} - \mu m g \,\text{sgn}(\omega) = I_{cm}\gamma_{w,z} \tag{3.2}$$

where $\boldsymbol{u}$ and $\boldsymbol{\alpha}$ are linear velocity and acceleration vector expressed in O_{w}, ω and $\gamma_{w,z}$ are angular velocity and acceleration expressed in O_{cm}, $\boldsymbol{r}$ is a vector connecting the contact point to CoM and g is the gravitational constant. The contact is fixed if $\boldsymbol{F}$ lies inside the contact friction cone with angle $\theta_{fr} = tan^{-1}(\mu_c)$.

If $\text{F}_y \leq \mu_c \text{F}_x$, the robot end effector does not slide along the object surface and $\boldsymbol{r}$ will not vary during the push. Accordingly, Eq. (3.1) and (3.2) can be directly used to estimate the object mass and inertia. Nonetheless, in a real planar pushing experiment, the robot end effector may slide on the surface of the object due to imperfect contact surface, low contact friction and an uncertainty in the force direction. That

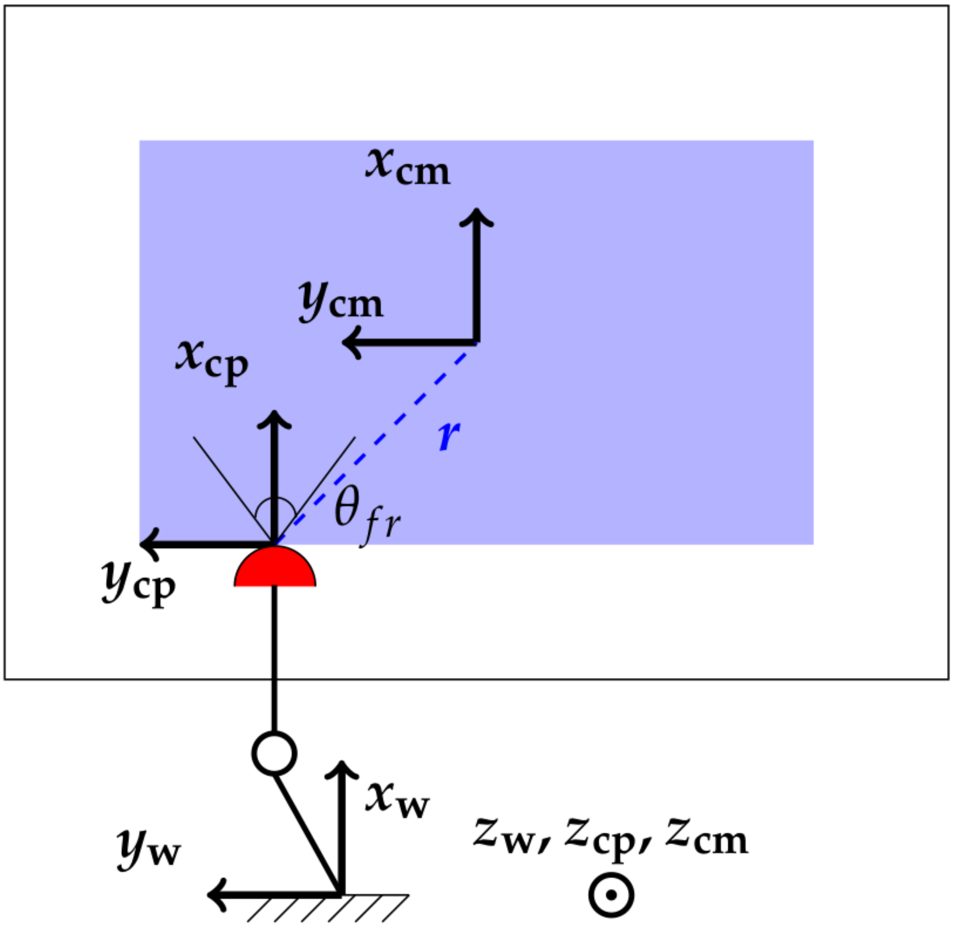
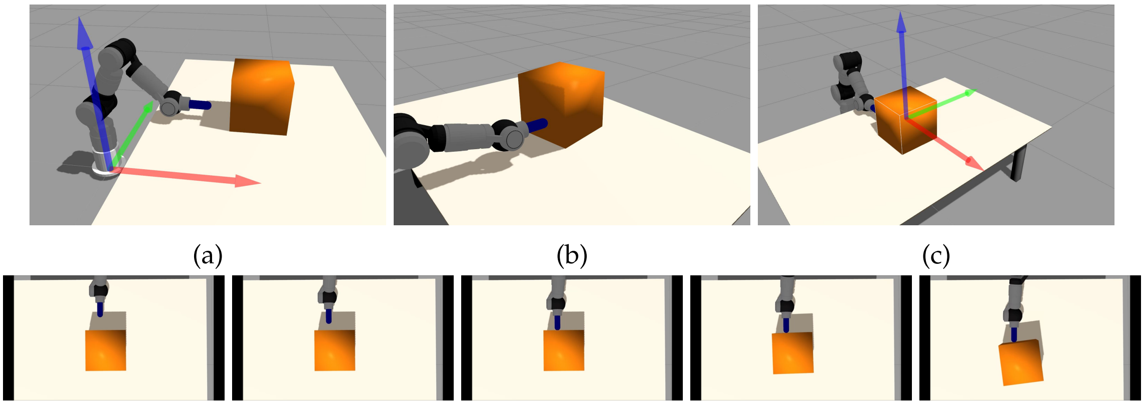


Figure 3.2: Top view of geometric configuration for a pushing experiment: a robot pushes a rectangular object (purple) on a table (white) with its end effector (red). The coordinate frames denoted with O_w, O_{cm} and O_{cp} are the world frame and coordinate frames attached to the center of mass of the object and to the contact point of the robot end effector and object surface, respectively. x_{cp} is normal to the contact surface. The vector connecting the contact point and the center of mass is denoted by $\boldsymbol{r}$. The friction cone is shown with thin lines on the contact point, and has an angle of θ_{fr}. All the z coordinates are normal to the page.

is very common in the case of open-loop pushing, i.e. when the pushing direction and force are not closed-loop controlled. Although Eq. (3.1) and (3.2) are still valid, they cannot be easily implemented as the change in $\boldsymbol{r}$ and applied force direction may not be immediately and precisely measurable. As a result, it is very difficult to build a robust deterministic model of the relationship between force and linear acceleration as well as torque and rotational acceleration. Machine learning regression approaches (Borchani et al., 2015) have been proposed to deal with such issues by building an approximate model of a linear or nonlinear behavior from a set of measured input-output data (training dataset).



(a) (b) (c)

(d)

Figure 3.3: Robot setup for cube pushing in Gazebo simulation: (a),(b) and (c) show the side, back and front view of a Schunk LWA4D robot setup in the Gazebo simulator, with a pushing tip (shown in blue) and force/torque sensor. A set of 30 cubes with different masses, dimensions, friction coefficients and rotational inertias is created. For each cube, the robot executes 10 open-loop pushes along the front edge of the cube with a predefined velocity, and measures the applied force and torque as well as the translational and rotational displacement of the cube. This leads to 300 measurement sets, that are used to train a random forest that estimates the mass and inertia of a new object by pushing it only once. Figures in (d) are instances of a push.

3.3.2 Data collection

The pushing object set was created in the Gazebo simulator and consists of 30 simulated cubes with different edge lengths ($15 \leq \mathrm{a} \leq 30$ cm), masses ($0.1 \leq \mathrm{m} \leq 1.5$ kg), friction coefficients ($0.15 \leq \mu \leq 0.6$) and rotational inertias ($5.983 \times 10^{-4} \leq I_{zz} \leq 171 \times 10^{-4}$ kgm^2). The parameter ranges are selected so that the resulting objects can be physically realistic. For example, the range of friction coefficient is selected by simulating various types of material contacts, such as wood-on-wood, or wood-on-plastic. After selecting the mass values, edge lengths, and friction coefficients from the specified ranges above, 30 random mass-edge-friction combinations were created to yield 30 different cubes. The inertia of each cube was then calculated from the mass and edge length, assuming uniform but unknown density. The parameters of each cube are shown in Table 3.1.

The world reference frame (O_{w}) and the frame attached to the cube CoM (O_{cm}) are shown in Fig. 3.3a and Fig. 3.3c. A simulated Schunk LWA4D robot manipulator with an attached stick is used to push every cube generated from the dataset, as shown in Fig. 3.3b. The robot applies 10 open-loop pushes to every cube, i.e. there is no closed-loop control of applied force and object position, velocity and movement direction. The 10 pushes are applied at points uniformly distributed across the front edge of the cube in $[-\frac{\mathrm{a}}{2}, \frac{\mathrm{a}}{2}]$ (as shown in Fig. 3.3(b)). The robot pushes the cube relatively low on the pushing face side, in order to prevent tilting the cube and degenerate its smooth motion.

The arm is equipped with a force/torque sensor at the end effector. The simulation environment is set up so that the robot always starts moving from an initial

Table 3.1: Simulated cube parameters

Cube No	Mass (kg)	Inertia ($\frac{kg}{m^2}$)	Edge length (m)	Friction coeff
1	0.147	0.00103	0.205	0.46
2	0.475	0.0053	0.236	0.15
3	0.81	0.0115	0.292	0.57
4	0.857	0.0103	0.268	0.43
5	0.24	0.0025	0.252	0.22
6	0.478	0.0039	0.221	0.31
7	0.52	0.0034	0.197	0.24
8	0.66	0.01	0.3	0.44
9	0.62	0.0083	0.284	0.37
10	0.1	0.0005	0.189	0.6
11	0.336	0.0025	0.213	0.26
12	0.76	0.0038	0.173	0.25
13	0.43	0.0048	0.26	0.4
14	0.71	0.0071	0.244	0.39
15	0.384	0.0048	0.276	0.32
16	1	0.0037	0.15	0.58
17	0.952	0.0043	0.165	0.42
18	0.289	0.0015	0.181	0.33
19	0.194	0.0017	0.2289	0.21
20	0.9	0.0037	0.157	0.2
21	1.278	0.0046	0.15	0.28
22	1.444	0.0171	0.267	0.54
23	1.056	0.0141	0.283	0.52
24	1.222	0.0057	0.167	0.53
25	1.333	0.0089	0.2	0.3
26	1.167	0.0091	0.2170	0.35
27	1.389	0.0126	0.233	0.37
28	1.5	0.084	0.183	0.49
29	1	0.0104	0.25	0.5
30	1.111	0.0167	0.3	0.56

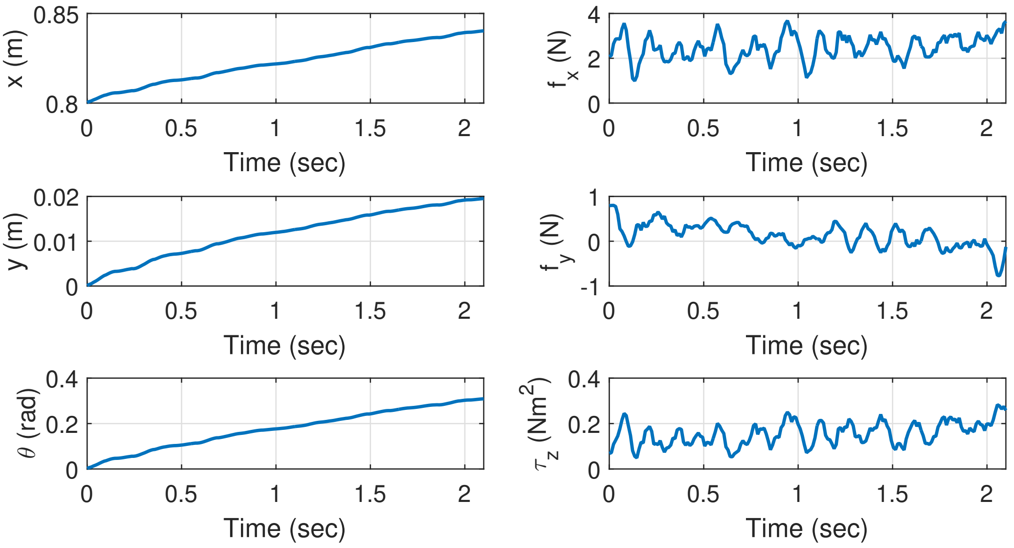


Figure 3.4: Example measurement signals for a single push. The shown sizes are measured for each push. The mean and variance of each signal are extracted. By representing each signal to its statistical values, a 12-dimensional feature vector is built, which is used as input for the regression model.

configuration shown in Fig. 3.3d, and different cubes are positioned on the table as shown in Fig. 3.3c. The robot's end effector moves in the direction of the x-axis of O_w (the red axis in Fig. 3.3a) for 50 cm where it touches the instance cube after almost 20 cm. It starts moving from zero velocity with acceleration of 0.2 m/s^2 taking less than 16 cm to reach the predefined maximum speed of 0.25 m/s. Hence, the end effector pushes all the instances with the speed of 0.25 m/s. The object position and orientation, as well as the force and torque measured by the force/torque sensor are collected at 80 Hz resulting in 6 measurement signals.

It should be noted that two rounds of data were collected in total, using the ODE and Bullet physics engines. The goal was to check whether the final result was affected by the selection of physics engine. Initial data collection and preliminary training showed regression performance was similar in both cases (less than 0.1% error difference was observed in mass and inertia). This happens because the pushing motions were relatively slow, making inertial and contact forces easier to process by the engine, leading to measurements of nearly similar quality. The Bullet engine performed faster and the motions were smoother in the simulator, and for this reason the Bullet dataset results are presented henceforth.

3.3.3 Multi-output regression random forests

To perform the multi-output regression a Multi-Output Regression Random Forest (MORRF) was selected. This section provides a brief description on how MORRFs operate and conduct predictions. A more detailed description of Random Forests is given in Breiman, 2001, and their multi-output counterparts in Borchani et al., 2015.

Decision trees for regression

Random Forests are based on Classification and Regression Decision Trees (CARTs) Duda et al., 2012. A CART is a non-parametric prediction model that has a structures tree-like form. It works by partitioning the feature space in hypercubes (in the case where the feature vector has dimension larger than 3). Each node of the tree contains a condition for one of the feature space variables, and the tree is split according to the outcome of the condition. A Regression Tree (RT) splits the node by searching all variables of the feature vector to find the split that minimises the *impurity*, or the mean-squared error of the target in the data set resulting from the split. In the case of single-output regression, the MSE at a node N is defined as:

$$MSE = \frac{1}{n_t}\sum_{i=1}^{n_t}(y_i - \bar{y})^2 \tag{3.3}$$

where n_t is the number of remaining data points at the node, y_i the target variable of each data point, and $\bar{y}$ the mean of the target variables. In the case of multi-output regression, the split rule is similar, but is is calculated for the multi-variate target:

$$MSE = \frac{1}{n_t}\sum_{l=1}^{d}(\sum_{i=1}^{n}(y_i^{(l)} - \bar{y})^2 \tag{3.4}$$

where d is the dimension of the target vector. Starting from the first node, the sets are split iteratively until a pre-defined termination criterion applies, and a leaf node is reached. The prediction value of the target is then the average of the leaf space. The number of splits at each node is variable, although it is typically 2 and the tree is then a *binary* tree. The training algorithm of a binary multi-output regression tree is given below:

- Let S_{tr} the training set, with $S_{tr} = (X_1, Y_1)...(X_n, Y_n)$, where X $\in R^{n_x}$, $Y \in R^{n_y}$, n, n_x, $n_y > 0$
- Let $x_i \in X$, $i = 1 \ldots n_x$ be a feature in the input space, and V_i, $i = 1 \ldots n_x$ the set of its real values found within the training set
- 1. Set $S = S_{tr}$
  2. For each x_i and $v \in V_i$
     (a) Split S in two datasets $S_<$ and $S_>$, corresponding to the sets where $x_i \leq v$ and $x_i > v$
     (b) Estimate the MSE from Eq. (3.4) for both datasets, $MSE_<$ and $MSE_>$
     (c) Select the split that minimises $MSE_< + MSE_>$
     (d) Recurse with $S = S_<$ and $S = S_>$
     (e) Terminate when termination criteria are met (only one data point on each leaf, desired tree depth reached e.a.)

Random forests

CARTs are a very intuitive and simple model for prediction. Their main disadvantage is that their method of operation divide the input space with a "sum of squares" surface, and so they tend to overfit to the data, especially when the tree depth is large. A way to solve this problem and reduce prediction variance is by applying *bagging* (bootstrap-aggregating). When *bootstraping* with CARTs, new trees are trained in new datasets resulting from sampling with replacement in the original dataset. By *aggregating* the prediction of all such trees, this set of trees provides low-variance predictions. However, trees created by fitting bagged datasets tend to have similar structure due to the fact that the all the input features are used for the prediction. A *Random Forest* (RF) further decreases variance by using subsets of the input feature space and training on bagged datasets. This results in a large number of trees (forest) with varying structure. The averaging of all the trees is the prediction result. The number and depth of trees, the feature space partitioning, and the bootstrap sampling method are some of the training hyperparameters affect the fitting and prediction result. Since in this thesis the regression has more than one target variables, the learning methods are *Multi-Output-Regression Random Forests* (MORRFs).

While technically any regression algorithm could be modified to accommodate a multi-objective case, Random Forests were selected for a variety of reasons. The usage of multiple trees on bagged datasets reduces the error variance across the testing set, as the prediction on a new input is passed through a large number of weaker estimators (CARTs), and the results of all estimators are averaged. This also makes RFs resistant to noise on the input feature. Typically, increasing the number of trees reduces the variance, but after a specific number, further increase of the tree number does not induce significant reduction in variance. To achieve this robust performance, RFs require large amounts of data, typically in a range of thousands. As will be shown in the next sections, large amounts of data were required for inertial parameters estimation with pushing, and so RFs were preferred. Finally, RFs operate by subdividing the input space into sums of "hyper-rectangles", meaning that the decision surfaces around each tree are rectangular in the dimension of the input space. Because of this property, RFs tend to perform well when the input is within the range of the input limits used for the training process, but not well when the input is out of range, limiting the generalisation capabilities. The data gathered in this Chapter come from a large range of inertial parameters, and RFs are used to train on this range. By gathering data from such a large range and using an RF, the need for generalisation is not required, as generalisation would mean that the robot is estimating the parameters of a very light, or very heavy (upushable) object.

3.3.4 Estimation on pushes

The data collected by pushing the cubes (input) together with the know n inertial data (output) of the instance cubes are used for training a multi-output regression model. A MORRF is used for the estimation, which is robust to noise and outliers, as shown by Borchani et al., 2015. This is critical as the experiments include both uncertainties and measurement noise. (Fig. 3.4).

The data collected during pushing are split in training and testing sets. The training set includes data from 22 cubes and the testing set from the remaining 8 cubes. The cubes were selected at random for the training and testing sets. All measurements are expressed in the O_{W} frame. To tackle the measurement noise and pushing uncertainties, the measurement signals are considered to be random. A useful way to extract information from the noisy signals is to represent them with their statistical properties. As such, the mean and variance of each measurement signal are calculated and are concatenated to be fed as feature vector to the random forest. Thus, the input feature vector $\bar{v}$ for each push consists of 12 scalars. The output vector consists of the mass and rotational inertia of each cube, i.e. 2 scalars. The random forest maps a 12D space to a 2D space. To optimise the hyperparameters of RF (number of trees, depth and feature space partition) a combined grid search with 10-fold cross-validation was performed. The cross-validation sets included all the pushes from 3 cubes, i.e. 30 pushes. The optimal number of parameters was 1000 trees of depth equal to 7. The feature space was partitioned to 75% , i.e. at every tree of the forest 9 out of 12 features were used for training.

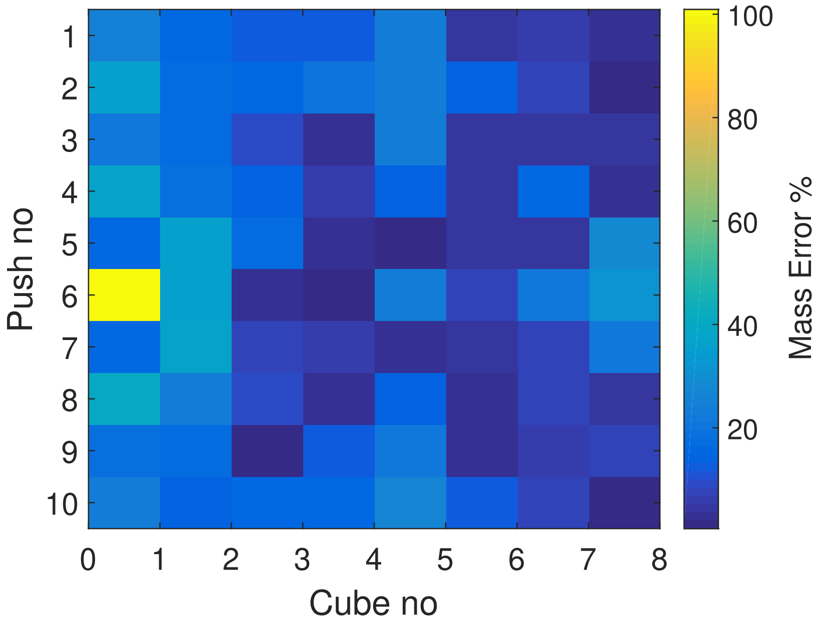
3.4 Experimental results and discussion

Large ranges of cube sizes, masses and frictions are included in the training set. The performance of the algorithm is measured by the relative error of the ground truth m_{gt}, I_{gt} and estimated m_{est}, I_{est} values for mass and inertia. This leads to the following 2 error metrics, i.e.

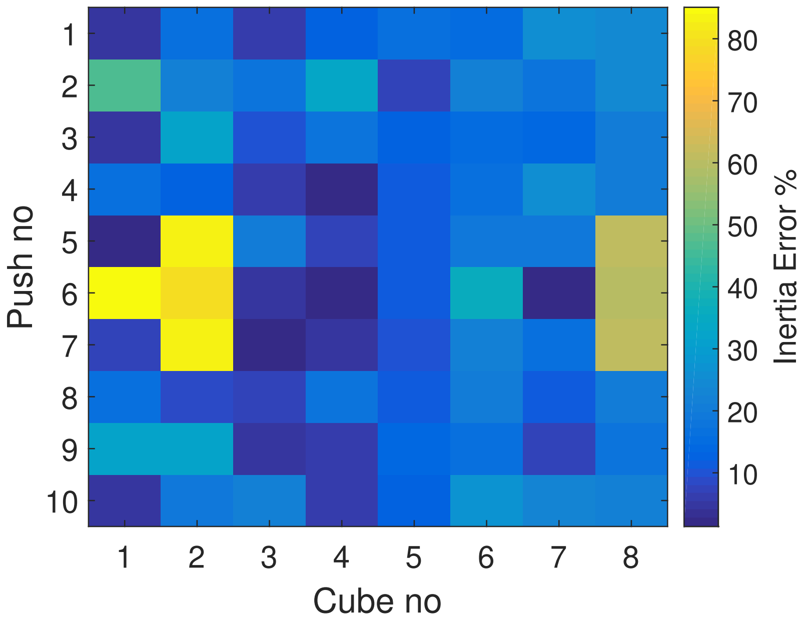
$$e_m = \frac{|m_{est} - m_{gt}|}{m_{gt}} * 100\ ,\ e_I = \frac{|I_{est} - I_{gt}|}{I_{gt}} * 100. \tag{3.5}$$

These error metrics represent the difference between the real value and the estimated value as a percentage of the real value. The use of percentage enables an error description that is does not depend on the magnitude of the real value. Consequently, the performance can be measured along different masses and lead to more descriptive conclusions. The errors can also be averaged to give an overall performance on the testing set. The ground truth for each cube is calculated as follows: Each cube has a mass and an edge value assigned, and assuming uniform density, the inertia is calculated from the formula $I_{gt} = \frac{1}{6}ms^2$, where m is the mass in kg and s is the edge in meters.

The results of estimating the mass and inertia of the 8 testing cubes are shown in Figs. 3.5a and 3.5b. It can be seen that in general, pushes in the middle of the cube have lower mass errors than pushes on the sides of the cube. In contrast, the pushes on the sides yield lower errors for the inertia. In fact, when the robot pushes close to the middle, the object does not rotate and the rotational acceleration is not observed; hence, the inertial modality of the system is not excited by the applied input (as per Eq. (3.1)) and an imprecise estimation of the inertia is expected. On the other hand, the pushes applied to the point close to the side of the cube yield larger angular velocity (and smaller liner velocity) for the object resulting in a better estimation of inertia (as per Eq. (3.2)). The overall error performance for each cube is shown by averaging the rows for each error image. The results are shown in Fig. 3.6a, and suggest that each cube has varying estimates both in mass and inertia. This is attributed to every cube having varying combinations of size, mass and friction coefficient, factors that affect the pushing and the estimate. Likewise, averaging the columns of each image results in a metric for the performance of every pushing side,



(a)

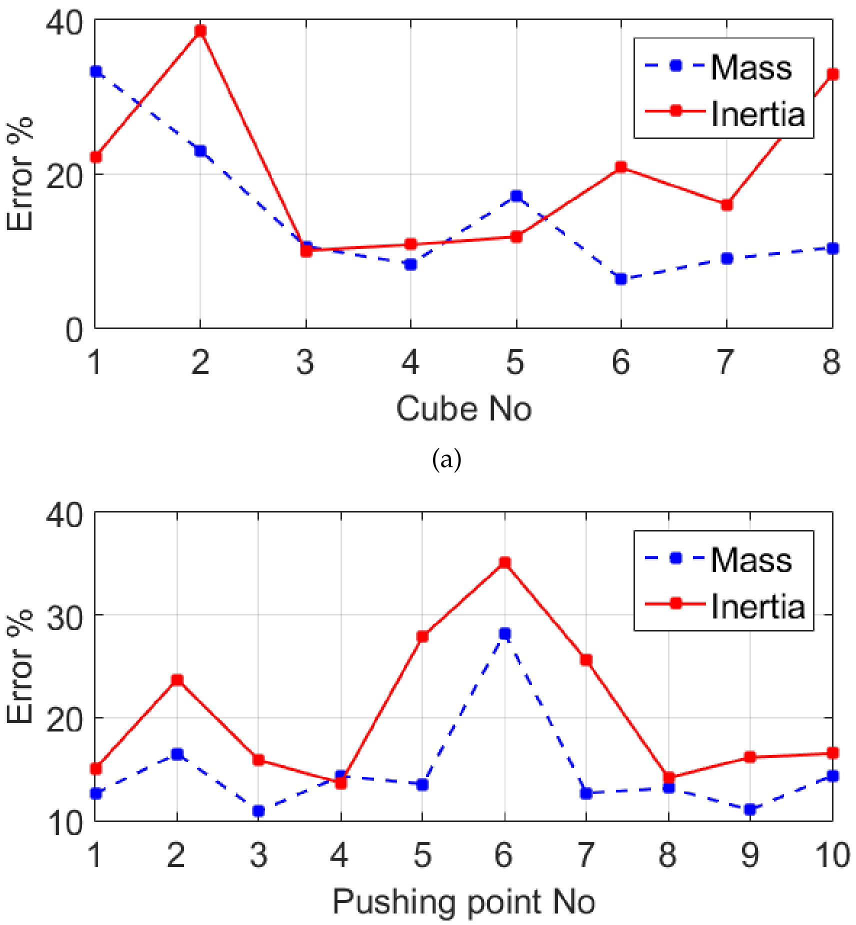


(b)

Figure 3.5: Error images for all the pushes applied to the cubes in the test set. The plots show the errors of estimation for each cube, and for each push at different points along the cube edge. It can be seen that in general, pushing closer to the middle of the cube provides a better estimate of mass, and the pushes closer to the edges yield smaller error of inertia estimate. (a) Mass error (b) Inertia error.

as shown in Fig. 3.6b. In general, pushing close to the sides of the cube leads in better estimate of inertia and slightly worse estimate of mass.

Apart from the estimation accuracy, the influence of each parameter of a cube on



(a)

(b)

Figure 3.6: The average of the testing set error values (shown in Fig. 3.5): (a) The average errors of the inertia (red line) and mass (blue line) estimation across all the pushes for each cube. Cube No. 3 yields the smallest error of mass and relatively small error for inertia. (b) The average errors of the inertia (red line) and mass (blue line) estimation across all the cubes for each push. Pushing points closer to the side of the cube, namely No. 3, 8 and 9, yields the smallest error of mass and inertia estimation.

the performance of the approach is evaluated. To do so, the cube associated with the minimum estimation error of mass and inertia is selected, i.e. the cube No. 3 in Fig. 3.6a. From this cube, a second set of cubes is generated by using its parameters as nominal: the size denoted by ā, the mass denoted by m̄ and the friction coefficient denoted by $\bar{\mu}$. While the two parameters are kept constant, the third is changed in a range of -30% to +30% of the nominal value. For each parameter 6 uniformly distributed variations of the nominal values are taken, leading to 18 new cubes in total. This time, a single push is applied at each obtained cube, at a location with a relatively low error for both mass and inertia based on the results showed in Fig. 3.6b. The obtained results are shown in Fig. 3.7.

Fig. 3.7a shows that different sizes of the cube and friction coefficients yield error of mass equal to almost 12% with very small variation. Likewise, different masses of the cube and friction coefficients yield almost an error of inertia of 15% with small

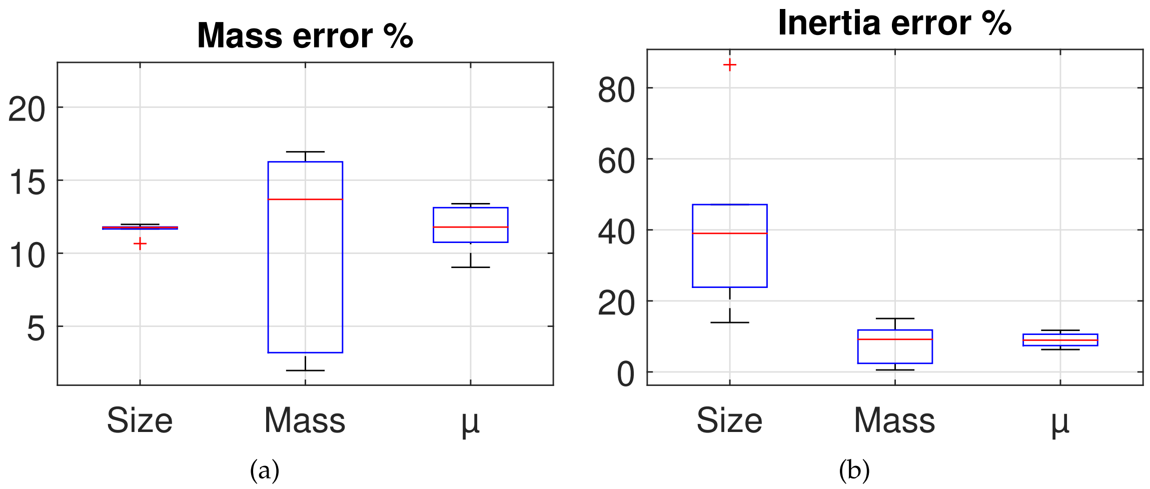


Figure 3.7: Box plots of mass and inertia error in the second set of cubes. The plots show how the cube size, mass and friction coefficient correlate with the mass and inertia estimation errors for a single push. (a) The mass estimation error stays less than 20% for all changed parameters where the median of the obtained errors illustrates that one can always expect almost 12% error. Changes in cube mass lead to large variations in estimated error of mass, but the overall error values are still relatively low. (b) The inertia estimation error is highly affected by changes in the cube size, with a relatively high value of error median. Changes in mass and friction coefficient lead to a low value of errors with small variations.

variations. In contrast, changing the size of the cube results in a large value of inertia estimation error with a large variation. Similarly, changing the mass of the cube yields a reasonably small error of mass estimation with a relatively large variation. Since inertia is proportional to square of length and the cube length is changed, it can be reasonable to see a significantly large inertia estimation error as per the change of size. For all the other parameters the observed errors are small.

3.4.1 Simulation with novel objects

Finally, the generalisation capability of the random forest is evaluated by estimating the mass and inertia of unseen objects, which belong to a completely dissimilar object

Table 3.2: Inertial properties of objects shown in Fig. 3.8 for final testing.

Object	Mass $[kg]$	Inertia $[kgm^2 \times 10^{-4}]$
Ellipse	1	80.982
Hexagon	1	41.700
Triangle	1	41.214
Saucepan	0.8	70.381
Tank	0.5	13.947
Rock	2.5	77.723

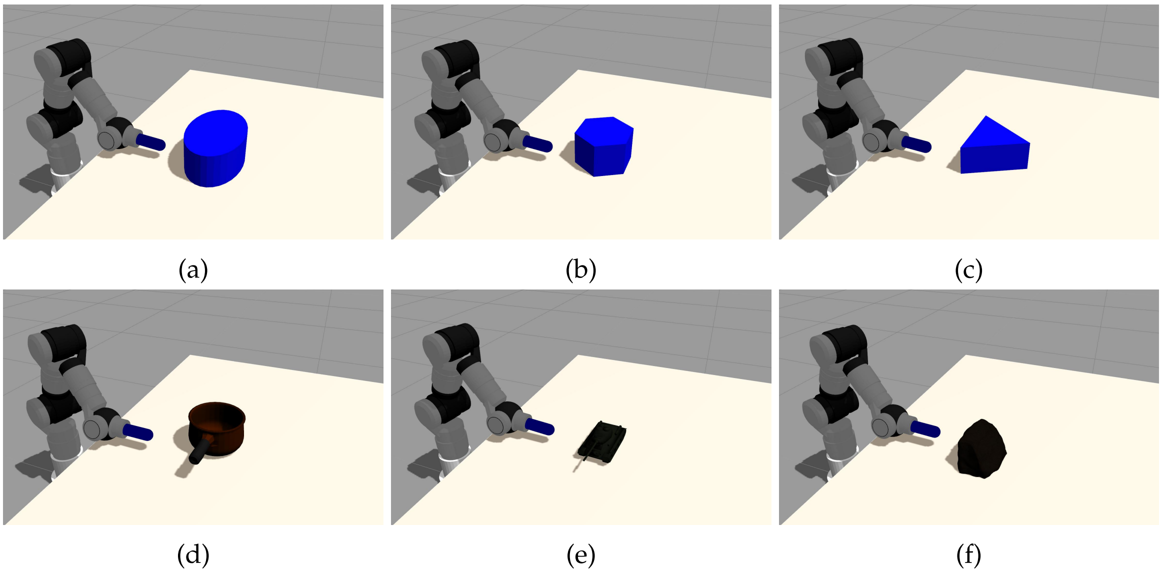
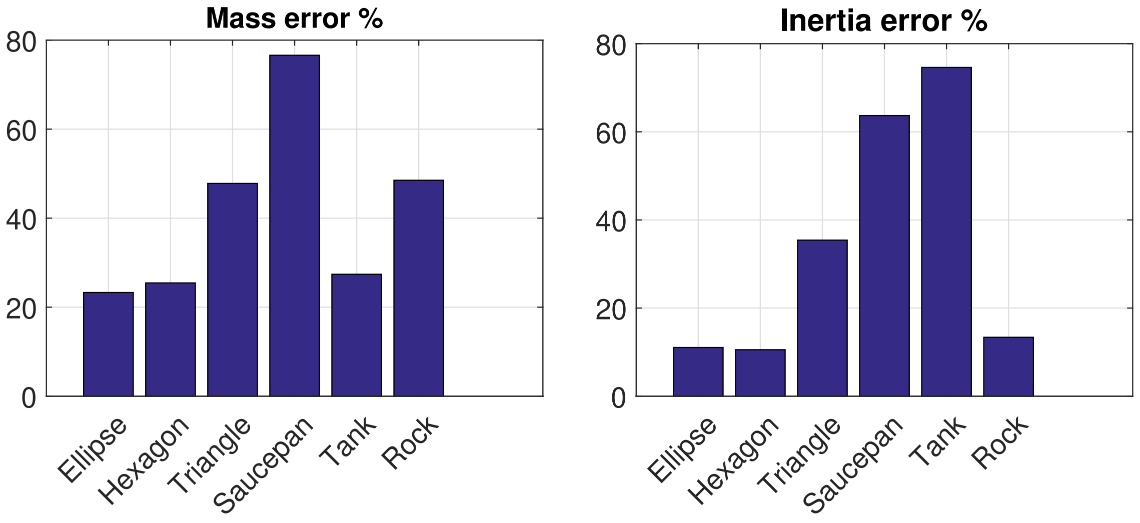


Figure 3.8: The set of objects generated for evaluating the generalisation capability of the proposed method: at the third evaluation stage, 6 new objects from the inertial dataset are selected and the push experiment is performed. These objects include: an ellipse primitive (a), a hexagon primitive (b), a triangle primitive (c), a saucepan (d), a toy tank (e) and a sample rock (f).

class. This property could be crucial to estimate the mass and inertia of a real world object utilising the model trained in simulation. First, set of objects from the inertial dataset is used. The set includes an ellipse, a hexagon, a triangle, a saucepan, a toy tank and a rock, with assigned masses and friction coefficients. The inertias are calculated in Meshlab and scaled to match the assigned mass. The objects are shown in Fig. 3.8a-3.8f, and their inertial parameters in Table 3.2. Again, a single push is applied at a point close to each object's edge. The regression model trained by the cube data is used for estimating the mass and inertia values of the unseen objects. The resulting mass and inertia estimation error are shown in Fig. 3.9. These results suggest that the model performs better with objects that have size and mass within the range of those used in the training. Indeed, the mass errors are lower in the ellipse,hexagon and toy tank. The objects with parameters significantly out of the range used for training, namely the rock and the saucepan, show larger error values. In the case of inertia, heavier objects yield smaller error where lighter object results in significantly large errors. The estimation error represented as the error does not imply an absolute error value. For instance, the estimated value of the mass of the rock is 1.28 kg resulting in almost 50% mass estimation error of the rock weighing 2.5 kg. While the estimation has low accuracy, the results suggest that by making the training dataset bigger and even more diverse, the performance of the approach for estimating the mass and inertia of unseen objects and object classes will improve.

3.5 Large-dataset estimation

The conducted experiments were a good starting point in the designing of a data-driven estimation system. The final results brought forward a number of shortcomings of the method, that needed to be addressed for further improvement. The results



(a) (b)

Figure 3.9: Mass and inertia estimation errors for the unknown object simulation. The plots show the error for every object, with only one push applied. (a) The mass error is lower for objects that resemble the training cubes in similar shape and mass values. Heavier objects (rock) or objects with peculiar shape (saucepan) have higher error values. (a) The inertia error is again lower for objects that are similar to the training ones in shape, as well as larger objects. The errors are small for heavier object, namely for the rock.

demonstrated that the MORRF method offers a more accurate prediction on target values that were part of the training phase. The limited generalisation capabilities manifested with larger error values on unseen objects. In combination with the relatively small number of cubes and pushes that composed the training set, this means that there is a need for larger, more diverse datasets for training and testing.

Furthermore, the feature extraction process generated motion-based features suitable for data-driven learning, but not representative of the physical laws that characterise the pushing motion. A more careful feature extraction and selection process is needed, where the features represent quantities that appear in the well-defined laws of pushing motion. The inspiration behind this is found in earlier works on pushing and inertial parameter estimation, where the authors used measurements applied directly in physical motion laws and generated the prediction. Notable examples are Tanaka et al., 2004 and Yu et al., 2005 where the authors measured the applied forces and tracked the object's motion, and estimated the inertial parameters from Newton's Laws of Motion. By enriching the Random Forest method with physically meaningful input, one can achieve the *meta-modelling* of the pushing process, i.e. expressing a mathematical model under another model. In the robot pushing case, the analytical models of motion are re-expressed with a data-driven method.

Another shortcoming of the method described previously is the lack of testing with data extracted from a real-robot pushing scene. Such testing would add extra fidelity to the method, and show how it can cope with real issues, such as sensor noise and pushing uncertainties.

To address all these shortcomings a new solution was designed, so that its input features stem from the analytic laws of quasi-static pushing of objects. This way, the pushing law are expressed with a data-driven model, leading to more accurate estimations. For training and testing, two datasets are used: one extracted from simulated pushing, and the MCube Lab pushing dataset (Yu et al., 2016). The simulated dataset has an extensive range of object dynamics, and the M-Cube dataset consists of pushes on real objects. Using these two datasets leads to higher fidelity of the estimation process. The results suggest that in both cases one can achieve low-error and low-variance estimations, with only a single push on an object.

3.6 Quasi-static robot pushing mechanics

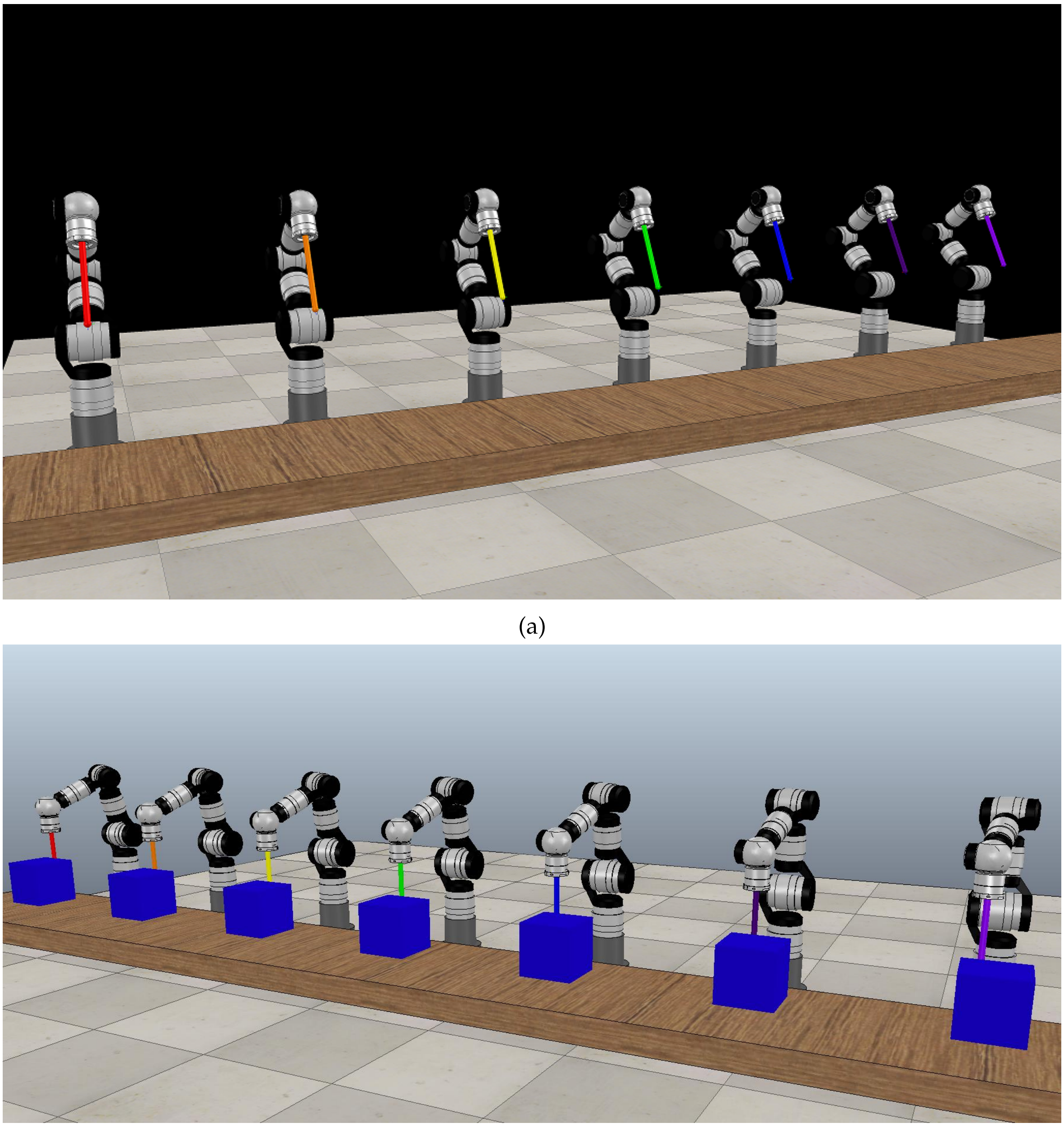
Let an object lying on a surface and a robot applying a push at one of its edges (Fig. 2). The coordinate frames of the world, finger-object contact and object's centre of mass are noted with W,CP and CM respectively. The distance vector between CP and CM is noted with $\boldsymbol{r}$. The direction of all z axes are towards the reader. When the robot applies a push, the object motion falls into one of three categories: rest, quasi-static and accelerating.

During rest, the robot applies an infinitesimal amount of force that can not overcome the static friction, and the object stays still. In accelerating motion, the robot applies enough force for the object to overcome the static friction, and start accelerating. The accelerating motion is characterised by the inertial parameters of the object, the applied force, and the friction coefficient between the pusher and the object, as well as the object and the surface. In this section, the interest is in in quasi-static pushing, where the robot applies just enough force for the object to match the frictional force. The object moves with low and constant velocity, and is not accelerating. The quasi-static pushing mechanics have been analysed in depth in a series of works Mason, 1986 Goyal et al., 1991a Goyal et al., 1991b, Lynch et al., 1991, Yu et al., 2015. A brief description of the necessary notions is provided.

When the object is in contact with an underlying surface, a *frictional load* (f_x, f_y, m) is applied on the support surface, and the object's bottom side. The principle of maximal dissipation indicates that the frictional load depends on the sliding direction and angle. The set of all frictional loads that can be applied on an object form a convex set in the 3D space, called *limit surface*.

To find an analytical expression for the limit surface, let dA be an infinitesimal surface patch on the bottom side of the object, that lies in $\boldsymbol{r_a}$ distance from a fixed reference frame. When the object is sliding, it rotates around an *instantaneous centre of rotation* $\boldsymbol{r_c}$. If the motion is purely rotational, $\boldsymbol{r_c}$ is located in the *centre of friction* in the the support surface. If the motion is a pure translation, $\boldsymbol{r_c}$ lies at infinity. The surface patch dA moves with instantaneous velocity $\boldsymbol{u_a}$, and the friction coefficient is μ. The frictional load applied on the whole surface A, is given by Mason, 1986:

$$\begin{aligned} \boldsymbol{f} = (f_x, f_y) &= \int_A -\mu \frac{\boldsymbol{u_a}}{|\boldsymbol{u_a}|} p(\boldsymbol{r_a})\, dA \\ m &= \int_A -\mu(\boldsymbol{r_a} - \boldsymbol{r_c}) \times \frac{\boldsymbol{u_a}}{|\boldsymbol{u_a}|} p(\boldsymbol{r_a})\, dA \end{aligned} \tag{3.6}$$



(a)

(b)

Figure 3.10: Extraction of the simulated dataset. 7 Schunk LWA4D robots are set in the V-REP simulator, with pushing tips and F/T sensors. The pushed cubes have different masses, dimensions, friction coefficients and rotational inertias. For each cube, the robot executes an open-loop push along the cube surface with given velocity, and measures the force and torque applied as well as the pusher and cube linear and rotational velocities. In total, the dataset includes 48000 pushes for a large variety of inertial parameters and friction coefficients. (a) The robot models used for pushing. (b) The robots pushing a spawned set of cubes. For each push, they measure the applied force and moment, the pusher velocities, and the object velocities.

where $p(\cdot)$ is the object's pressure distribution on the plane. In practice, it is very difficult to calculate the limit surface, because the pressure distribution, centre of

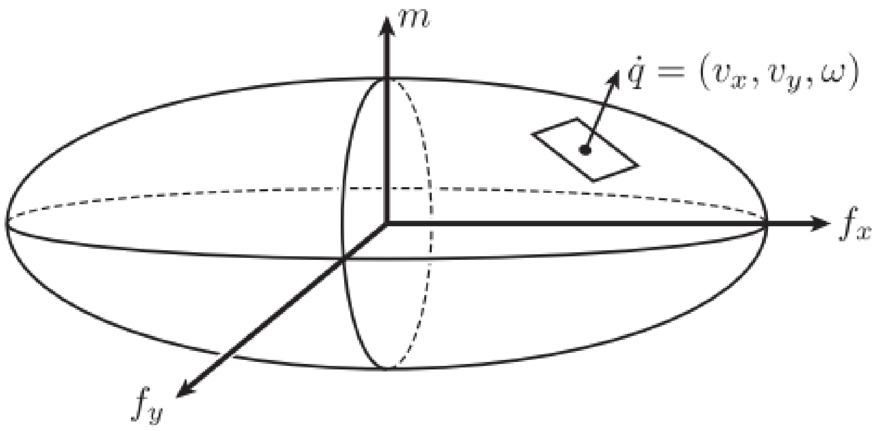


Figure 3.11: Limit surface approximation as an ellipsoid. The exact calculation of the limit surface may not be possible, and by approximating it as an ellipsoid the velocity. Points on the limit surface represent the total frictional load applied on the object. Due to the principle of maximum dissipation and the smoothness of the limit surface, the velocity of each point on the surface $\dot{\boldsymbol{q}} = (u_x, u_y, \omega)$ must be orthogonal to it. Image from [27].

rotation, and supporting surface can be unknown or even varying during motion. In Lee et al., 1991, it is suggested to approximate the limit surface as an ellipsoid (Fig. 3.11). The ellipsoid semi-principal axes and equation are found as follows:

- The maximum force that can be applied by the surface is $f_{max} = \mu M g$. This force is applied in the case of pure translation.
- The maximum moment that can be applied by the surface is $m_{max} = \int_A -\mu|\boldsymbol{r_a}|p(\boldsymbol{r_a})\, dA$. This moment is applied in the case of pure rotation by the projection of the object's CoM on the support surface, which is considered the moment reference point. The fraction of maximum torque to maximum force is noted with $c = \frac{m_{max}}{f_{max}}$.
- The limit surface equation is ellipsoid (Fig. 3.11):

$$L(f_x, f_y, m) = (\frac{f_x}{f_{max}})^2 + (\frac{f_y}{f_{max}})^2 + (\frac{m}{m_{max}})^2 = 1 \tag{3.7}$$

Due to the principle of maximum dissipation and the smoothness of the limit surface, for a given frictional load on the limit surface, the object velocity $\dot{\boldsymbol{q}} = (u_x, u_y, \omega)$ must be orthogonal to it Goyal et al., 1991a. The orthogonality is imposed, by making $\dot{\boldsymbol{q}}$ parallel to $\nabla L(f_x, f_y, m)$. This leads to the following relationships between applied forces and object velocity:

$$\begin{aligned} \frac{u_x}{\omega} &= c^2 \frac{f_x}{m} \\ \frac{u_y}{\omega} &= c^2 \frac{f_y}{m} \end{aligned} \tag{3.8}$$

The next step after Eq. (3.8) is to express the motion of the object as a function of the motion of the robot pusher. This is done by introducing the *motion cone* Mason, 1986. The motion cone spans the object velocity, the same way the *friction cone* spans the applied force. The left and right limit forces of the friction cone result in the generalised object velocities $\dot{\boldsymbol{q}}_{\boldsymbol{l}} = [u_{lx}, u_{ly}, \omega_l]$ and $\dot{\boldsymbol{q}}_{\boldsymbol{r}} = [u_{rx}, u_{ry}, \omega_r]$. The contact velocities from this motion are the limits of the motion cone and are given by $\boldsymbol{v_l} = [u_{lx} - \omega_l r_y, u_{ly} + \omega_l r_x]$ and $\boldsymbol{v_r} = [u_{rx} - \omega_r r_y, u_{ry} + \omega_r r_x]$. If one notes with $\boldsymbol{u_p}$ the velocity of the contact point on the pusher, and $\boldsymbol{u_o}$ the velocity of the contact point, the motion of the object is dictated by whether $\boldsymbol{u_p}$ lies within the motion cone.

If the pusher velocity is within the motion cone, then the contact is *sticking* and $\boldsymbol{u_o}$ = $\boldsymbol{u_p}$ holds. The pusher velocity $\boldsymbol{u_p}$ and the object velocity are related by:

$$\begin{aligned} u_x - \omega r_x &= u_{px} \\ u_y - \omega r_y &= u_{py} \end{aligned} \tag{3.9}$$

The moment applied by the pusher is $m = \boldsymbol{r} \times \boldsymbol{f} = r_x f_y - r_y f_x$. This equation, along with Eqs. (3.9) and (3.8) give the velocity of the object as a function of the pusher velocity:

$$\begin{aligned} u_x &= \frac{(c^2 + {r_x}^2) u_{px} + r_x r_y u_{py}}{c^2 + r_x^2 + r_y^2} \\ u_y &= \frac{(c^2 + {r_y}^2) u_{py} + r_x r_y u_{px}}{c^2 + r_x^2 + r_y^2} \\ \omega &= \frac{r_x u_y - r_y u_x}{c^2} \end{aligned} \tag{3.10}$$

If the pusher velocity is not within the motion cone, then the contact is *sliding* and $\boldsymbol{u_o}$ is on one of the two boundaries of the motion cone $\boldsymbol{u_b}$. A part of the pusher velocity is lost due to slippage, and the rest contributes to the object motion. The fraction that is transferred is $\boldsymbol{u_o} = k\boldsymbol{u_b}$, with $k = \frac{\boldsymbol{u_p n}}{\boldsymbol{u_b n}}$, and $\boldsymbol{n}$ the contact normal. The object velocity is given by substituting the new $\boldsymbol{u_o}$ to Eq. (3.10), as if the object was pushed by a sticking pusher with reduced velocity. It is clear that in all cases, Eqs. (3.9) and (3.10) are crucial to fully characterise the object's motion from the pusher data.

3.6.1 Simulated dataset extraction

For the simulated data extraction, a robot pushing scene is set up using the V-REP simulator. Again, the Bullet physics engine was used without significant change in the estimation result. The scene includes 7 Schunk LWA4D robot manipulators with force sensors and pushing sticks attached. The scene is shown in Fig. 3.10b. The robots are pushing cubes that appear in front of them. The cubes have variable masses, edge size, inertias and coefficients of friction with the supporting surface. For the cube parameters 20 evenly-sampled masses are selected from a range of $[0.5, 5]kg$, 6 friction coefficients from a range of $[0.2, 0.6]$ and 3 cube edge sizes from $[0.1, 0.2]m$.

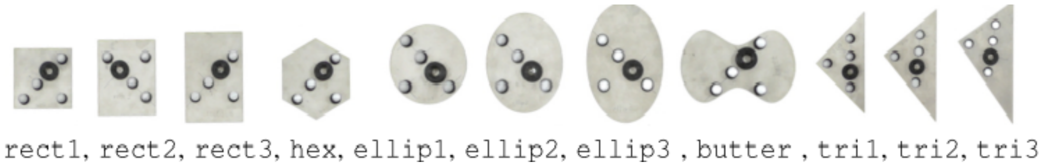
The rotational inertia for each cube is calculated according to the size and mass of the cubes, with the assumption of uniform density. The CoM of each cube is located in its geometric centroid, and the 2D distance vector is extracted from the first contact point to the cube's CoM (the $\boldsymbol{r}$ vector in Fig. 3.10b). Since the CoM needs a reference point, this 2D vector is used as CoM prediction in the training and testing phases of the algorithm. This way, the robot learns to provide an estimate of the object's CoM w.r.t. the first contact point, enabling predictions for various mass distributions. Reasons for selecting cube objects to form the dataset are their easiness of modelling, their planar contact surface with the supporting surface, and because their straight edges can be pushed without breaking contact. The choices for the mass are made to simulate objects that are heavy enough to produce measurable force signals above potential noise, but not extremely heavy to prevent smooth pushing. The friction coefficients are selected to simulate a range of every day contacts (e.g. wood to plastic, wood to wood, metal on plastic e.t.c) but not extreme cases such as icy terrains or very sticky contacts. The cube edge sizes are selected to generate variety in the inertia values.

Each cube is pushed under a different pushing profile. The profile consists of an initial and final point on the horizontal (y) axis along the cube edge (that can be interpreted as a pushing angle w.r.t. the cube's horizontal edge) and a constant pushing velocity. Each push has a length of about $0.15m$. 5 initial and 5 final points are selected along the horizontal edge of the cube and 4 pushing velocities $0.01, 0.02, 0.04, 0.06m/s$. The pushing velocities were selected to be low and constant, in accordance with the described quasi-static modelling. The robot applies a push for every combination of cube parameters, velocities and pushing angles to get 48000 pushes in total, in about 30 hours of simulation.

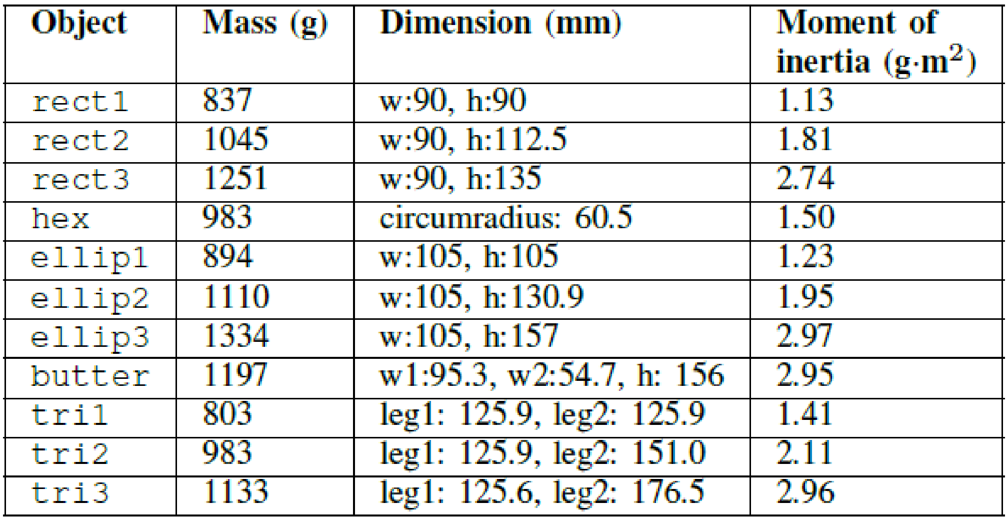
For each of the 48000 pushes, the applied forces are measured from the pushing stick to the cube for the duration of the push, as well as the cube's linear and angular positions and velocities.Each sample of every measured size was infused with artificial Gaussian noise of zero mean and $\sigma = 0.05/3 * sample\ value$ to achieve a ±5% spread around the sample value. Different values of spread were also tested, up to 12%, but they were found to pose negligible difference in the final experimental results. This noise addition makes the estimation procedure more applicable to a real world scenario, since in real pushing there exist many different sources of uncertainty such as unknown object pressure distribution, stiction, tracking and measurement errors etc.

3.6.2 Real object dataset

To evaluate the method with data from real robot pushing, the high fidelity pushing dataset presented in Yu et al., 2016 is used, and for clarity is is noted as the M-dataset throughout the Chapter. The M-dataset consists of a large number (over 200K) of straight pushes, executed by a real robot manipulator on a set of objects. It contains different combinations of pushing velocities, angles, and pushing points on the objects, and their execution is consistent with the quasi-static analysis mentioned earlier. The pushed objects are 11 in total, and are shown in Fig. 3.12 along with their properties. The inertial properties of the objects vary, although the mass range is lower compared to the simulated dataset. Four different surfaces are used for the pushing,


rect1, rect2, rect3, hex, ellip1, ellip2, ellip3 , butter , tri1, tri2, tri3

(a)



Object	Mass (g)	Dimension (mm)	Moment of inertia (g·m^2)
rect1	837	w:90, h:90	1.13
rect2	1045	w:90, h:112.5	1.81
rect3	1251	w:90, h:135	2.74
hex	983	circumradius: 60.5	1.50
ellip1	894	w:105, h:105	1.23
ellip2	1110	w:105, h:130.9	1.95
ellip3	1334	w:105, h:157	2.97
butter	1197	w1:95.3, w2:54.7, h: 156	2.95
tri1	803	leg1: 125.9, leg2: 125.9	1.41
tri2	983	leg1: 125.9, leg2: 151.0	2.11
tri3	1133	leg1: 125.6, leg2: 176.5	2.96

(b)

Figure 3.12: The M-dataset objects. Images from Yu et al., 2016 © 2016 IEEE. (a) The 11 objects in the M-dataset have variable shapes. These objects are pushed in the training and testing phases of the estimation method. (b) The inertial properties and corresponding size of the M-dataset objects.

resulting in different friction coefficients. For more information on the M-dataset, the reader is encouraged to study Yu et al., 2016.

3.6.3 Feature extraction and training

The next step after acquiring the data, is to extract meaningful features for training. The parameter estimation process is essentially a regression problem. After acquiring the pushing dataset, the next step is to extract features from the dataset measurement signals that are fed into a regression algorithm and provide a value for the inertial parameters.

As per the quasi-static analysis, there exists a non linear relationship between applied forces and moment by the robot, object velocities, and the c parameter. The c parameter is the division of the f_{max} and m_{max}. In the division, the friction coefficient is removed, and the type of surface does not have any effect on the motion. So, c includes information about the object's mass and pressure distribution. The moment of inertia I_{zz} intuitively manifests itself through the pressure distribution from the object to the underlying. The pressure distribution is affected, among others, by the weight distribution of the object on the surface. Since g the gravitational is constant, the distribution of mass along the object's horizontal surface (i.e. moment of inertia) affects the pressure distribution, and so the overall motion even though the model is quasi-static. From the non-linear equations described before it can be seen that an closed-form expression between inertial parameters, applied forces and moment, and object velocities is tough to obtain. It is instead modelled it as a non-parametric regression problem as:

$$\boldsymbol{\theta} = F(f_x, f_y, m, u_x, u_y, \omega, u_{px}, u_{py}) \tag{3.11}$$

with $\boldsymbol{\theta} = [M, I_{zz}, r_x, r_y]$ the inertial parameter vector. For each the 8 quantities in the right part of Eq. (3.11) a signal is extracted, and treated as random signal. This is because the signal shape in time depends on the pushing profile. Each signal is split in 3 windows, to catch possible significant variations in time. The window size resulted from hyperparameter tuning. Larger window sizes resulted in higher feature dimensions and more complex learning models, and smaller window were not able to capture the variations in the measurement signal waveforms. The mean, standard deviation, and RMS value of each window are extracted, which leads to a $8 \times 3 \times 3 = 72$-dimensional feature vector for each push.

The feature vectors extracted from the datasets are used to train a Multi-Output Regression Random Forest (MORRF). The MORRF was selected because it performs well with larger amounts of data and have reduced variance along the predictions. Additionally, the mass range for the simulated data is within the operational limits of many arms. Items above 5 kg can be difficult to push and manipulate, and items below 0.5 kg are very light. Selecting a learning algorithm that could generalise well beyond these limits would be unnecessary. A Random Forest can fit well the range of the target variable during the training phase, even with poor generalisation.

3.7 Experimental results

To test the approach two estimation experiments are set up, one with each dataset. In all experiments, the effectiveness of the algorithm is measured using the *Average Percent Difference* e_{apd} of each predicted value d_{pred} and its corresponding ground truth d_{gt}:

$$e_{rpd} = \frac{2(d_{gt} - d_{pred})}{|d_{gt}| + |d_{pred}|} * 100\% \tag{3.12}$$

This metric expresses the difference of two values as a percentage of their absolute magnitude. This metric is chosen over the more well known Relative Percent Difference, because in this case dividing by the ground truth can skyrocket the error value. This happen s when the ground truth is close to zero, as it is the case with small inertias and CoM distances, corrupting the evaluation process. With the average difference, the error is bounded to ±200%. Over the testing set, the mean and standard deviation of the error is calculated. The use of a percentage instead of absolute quantities enables the mean and standard deviation to characterise the performance within the whole range of the target values with the same fidelity.

Table 3.3: Prediction results on the simulated testing set

Parameter	Error Mean %	Error Standard Deviation %
Mass	7.59	11.04
Inertia	13.49	10.20
Com x	12.59	8.62
Com y	20.61	8.12

3.7.1 Simulated dataset estimation

In the simulated dataset, the feature vector described above is used to train the MORRF. The predictor is the feature vector and the target vector is $\boldsymbol{\theta}$. The dataset is split in training and testing sets. The testing set is 10% of the dataset size, namely 4800 samples. The 48000 samples are sorted by mass size. The next step is to extract 1% of the lowest, 1% of the second lowest and 1% of the third lowest masses, $0.5, 0.736, 0.973kg$ respectively, and do the same with the higher masses ($4.526, 4.763, 5kg$), for a total of 6%. The rest 4% came from evenly sampling the rest of the masses. In the end, this gives us a 10% testing set, that over-represents the two limit cases (heavy and light objects). The remaining samples are the training size. The model then trains mostly in medium-range masses and tests mostly in limit-range masses. Again, a combined grid-search hyperparameter optimisation was conducted, with a 10-fold cross-validation. The validation split was 10% on the training set. The optimal number of parameters was 1500 trees of depth equal to 5. The feature space was partitioned to 75%. The training time is about 212 secs, on a Intel Core i7-8750H CPU @ 2.20GHz and 16 GB RAM laptop. The results are shown in Table 3.3, and an instance of a resulting tree is shown in Fig. 3.13.

It is evident that the presented training method achieves low error means and relatively low error standard deviations. The low standard deviation is inherent in ensemble algorithms, and the low error means are achieved by increasing the depth of the Random Forest. These properties further justify the selection of a MORRF as a learning algorithm in the presented setup. An exception would be the y-dimension of the CoM, where the mean appears slightly increased. Despite the selection of error metric limiting the error value to ±200%, the low values of the denominators in the error metric can still lead to frequent occurrences of the limit values, affecting the overall performance.

In Section II, it was mentioned that the surface type does not play a role in quasi-static motion, because the c^2 parameter consists of a fraction that deletes the friction coefficients. To test whether the model follows this principle the performance mean and standard deviation was calculated for each of the four friction coefficients in the dataset. The results are shown in Fig. 3.14. It can be seen that the performance remains quite robust to the changes in the friction coefficient, with some minor variations of the error mean occurring. As expected, the method also shows better performance towards the middle of the mass range, because the middle masses were more represented in the training set. The variations of the error standard deviation are almost negligent. The presented method was also proven quite resistant to noise. The final performance was not significantly affected by the added Gaussian noise, as error variations of about 0.2% at most were observed.

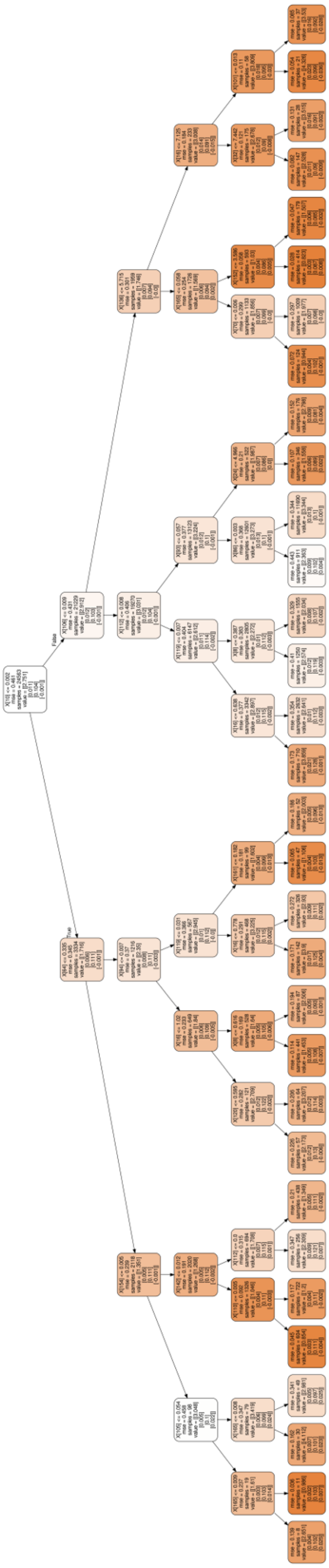
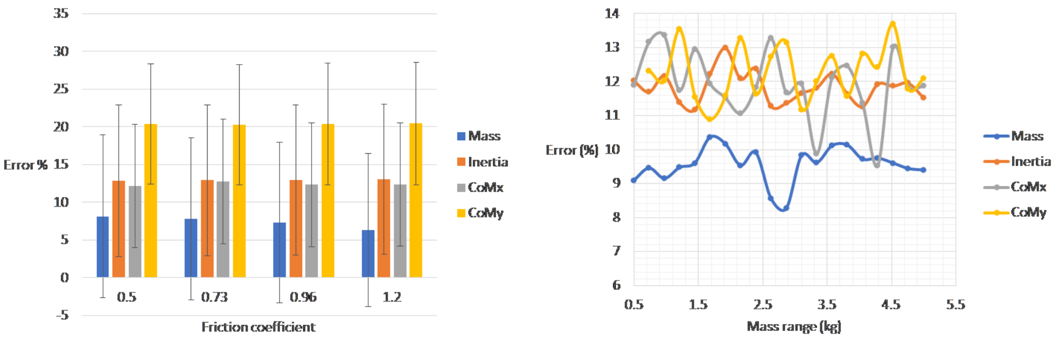


Figure 3.13: An example tree of the MORRF for large simulated data training.



(a) (b)

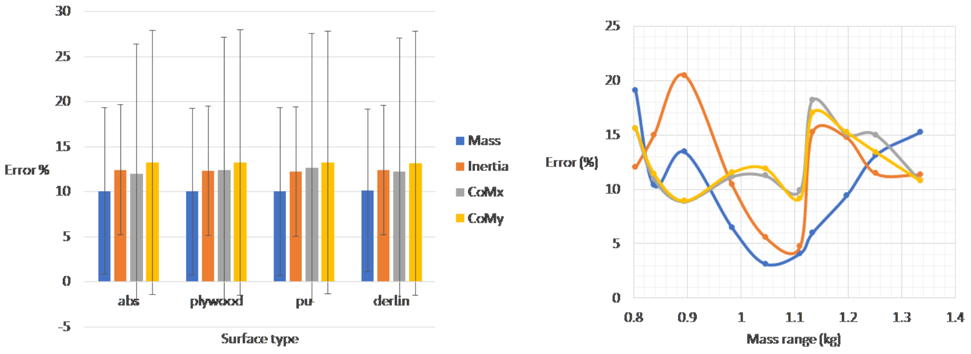
Figure 3.14: Mean performance of the proposed learning algorithm in the simulated dataset, plotted over the friction coefficient and mass range. It is in accordance with the overall performance in Table 3.3. (a) Mean performance across varying surface friction coefficients. The error means are noted with the bars, and the standard deviation with the vertical lines. (b) Mean performance across the mass range of the test set. The standard deviations are omitted for clarity.

Table 3.4: Prediction results on the M-datset testing set

Parameter	Error Mean %	Error Standard Deviation %
Mass	10.05	7.09
Inertia	12.44	7.38
Com x	12.90	15.05
Com y	13.32	15.31

3.7.2 M-dataset estimation

For the M-cube dataset, objects rect3, hex, butter, tri3 and ellip3 were left out for testing. Training and validation was conducted with all the pushes for the remaining objects. A MORRF is trained with same testing and validation percentages, 10-fold cross-validation, on the same laptop. Due to the larger size of the M-Dataset, the training time was about 1236 secs. The results are shown in Table 3.4. Again, the system can estimate the object's inertial parameters with low error, due to the low mean and relatively low standard deviation of the average percent difference. The performance is again checked for every one of the four different surfaces provided in the M-dataset. The results are shown in Fig. 3.15. Again, a smooth and robust performance is observed along varying surface materials and mass ranges, indicative that the learning model can properly describe the quasi-static pushing mechanics. Even though the middle range masses are again more represented in the training, the large number of samples in the dataset mean that the robot will be trained with enough samples from each range to warrant robust performance.



(a) (b)

Figure 3.15: Mean performance of the proposed learning algorithm in the M-Cube dataset, plotted over the surface type and mass range. It is in accordance with the overall performance in Table 3.4. (a) Mean performance across varying surface types. The error means are noted with the bars, and the standard deviation with the vertical lines. (b) Mean performance across the mass range of the test set. The standard deviations are omitted for clarity.

3.7.3 Comparison with an analytical method

To further validate the usefulness of the data-driven estimation method, an experiment showing a comparison with an analytical method was set up. More specifically, the MORRF method is compared with the analytical approach presented in Yu et al., 2005. The reason for selecting the method in ibid. of comparing, is that it demonstrates an algorithm for estimating all the 2D inertial parameters (mass, 2D CoM, rotational inertia) of a pushed object using analytical laws of motion and least squares error reduction. Model-based methods are known to be very accurate, but with limited generalisation to new objects. The goal is to compare the estimation errors and demonstrate the ability of generalisation of the MORRF-based method against a model-based one. The work in ibid. is heavily dependent in the scene configuration, object type, and robotic hardware used. In it, the authors conducted a real robot experiment using a customised end-effector with two force sensing fingers, They estimated the parameters of a single cuboid object by applying a number of pushes, substituting tracked motion parameters to physical motion laws, and using least-squares estimation for estimating each parameter. As the exact hardware and overall experimental configuration were not easy to replicate, a V-REP simulation was instead set up. In it, a 2-fingered mechanism with force sensors at each fingertip was created. The distance between the two fingers was set to 10 cm, as described in the experimental part of ibid. The mechanism was mounted on a Schunk LWA4D robot model. The setup is shown in Fig. 3.16.

The testing set for the comparison consists of a set of simulated cuboids to be pushed. The cuboids have two possible sizes, small and large, with dimensions 0.15 × 0.2 × 0.15 cm and 0.2 × 0.3 × 0.2 cm respectively. The mass values for the cuboids are 1, 2.5 and 4 kg, and the friction coefficient values are 0.6 and 0.5, in order to simulate typical values of friction between wooden and plastic surfaces. This results in 2*3*2 = 12 cubes in total for testing. The cuboids are shown in Fig. 2., and their properties in Table 1.

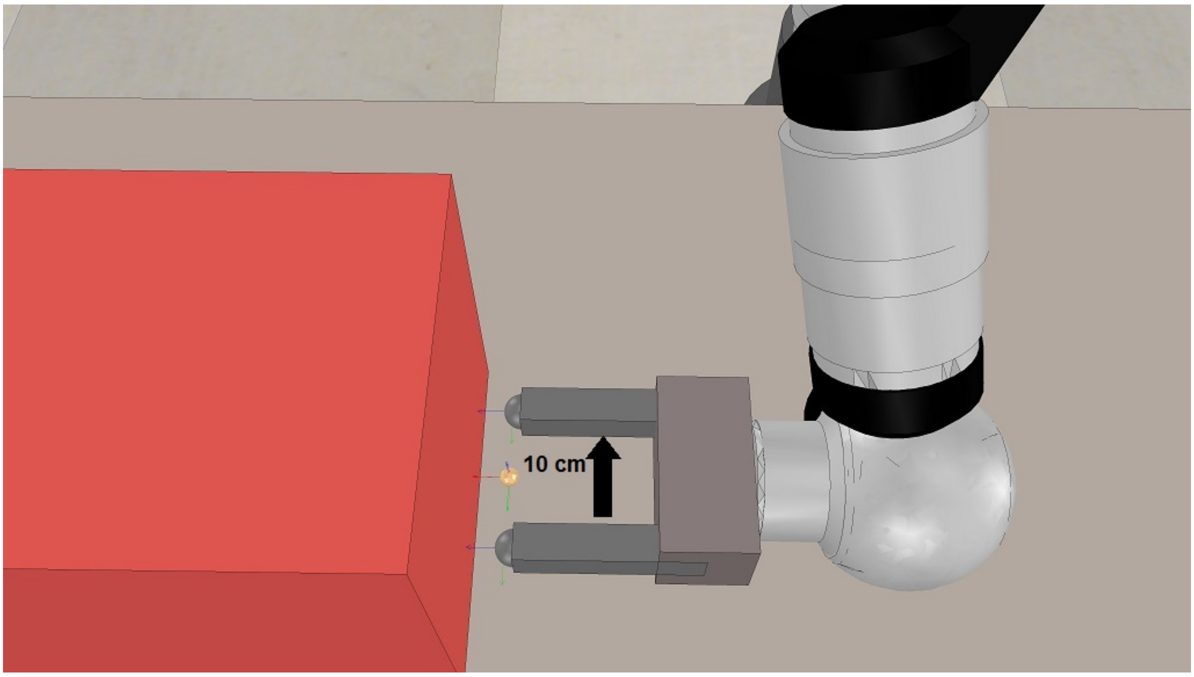


Figure 3.16: The 2-fingered mechanism mounted on the Scunk LWA4D arm, in the V-REP simulation. Each finger has force sensing capabilities. The distance between the fingers is 10 cm, as in Yu et al., 2005.

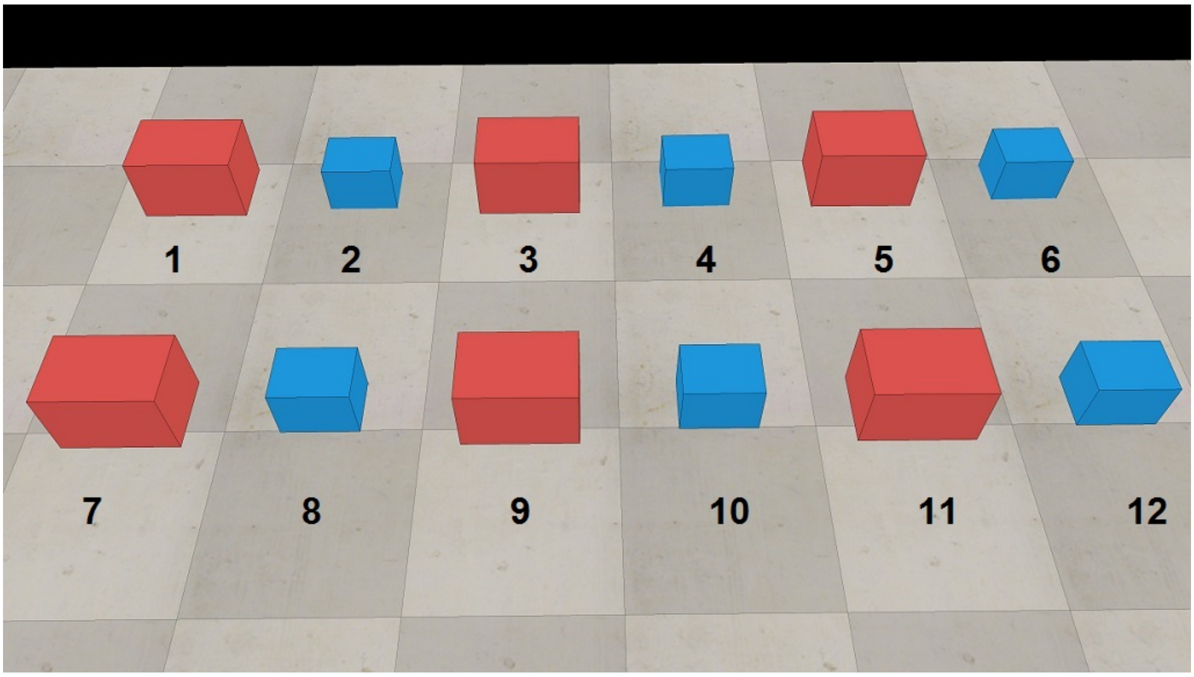


Figure 3.17: In total, 12 cuboids with different size, mass and material were used to test the method in Yu et al., 2005.

The method in Yu et al., 2005 requires a minimum number of 12 pushes for estimation, 4 for the mass, 4 for the CoM, and 4 for the inertia. In the original paper the authors conducted their experiment with 8 pushes of different accelerations for each size to be estimated. In the presented V-REP simulation 4 pushes were applied

Table 3.5: Properties of the 12 simulated cuboids for testing.

Cuboid index	Mass (kg)	Inertia ($kg \times m^2$)	Size	Friction coeff
1	1	0.01083	large	0.3
2	1	0.005208	small	0.3
3	1	0.01083	large	0.5
4	1	0.005208	small	0.5
5	2	0.02708	large	0.3
6	2	0.01302	small	0.3
7	2	0.02708	large	0.5
8	2	0.01302	small	0.5
9	4.5	0.04333	large	0.3
10	4.5	0.02083	small	0.3
11	4.5	0.04333	large	0.5
12	4.5	0.02083	small	0.5

for each size. For the mass estimation, 4 straight pushes are applied near the centre of long edge of each cuboid. For the CoM, 4 straight pushes are applied at the long edge of the cuboids, and 4 at the short side, both near the edge centre. For the inertia, 4 pushes are applied at the right side of the long edge, rotating slowly to 30^o. Each set of 4 pushes includes accelerations of 0.2, 0.4, 0.6 and 0.8 m/s^2. The collected data consist of measured finger forces, finger positions and linear velocities, as well as object position, linear velocity and angular velocity, on a 20 Hz sampling rate. The accelerations needed in Yu et al., 2005 were calculated from the measured velocities and simulation timestamp (50 ms).

For the data-driven part, the MORRF trained with the simulated cubes of Subsection 3.7.1 was used. The robot pushed every cube 5 times, two at $\pm 0.04m$ from the object's centre along the object's face, two at $\pm 0.08m$ and one on the centre (Fig. 3.18). The pushing distance is $10cm$ and the pushing velocity $0.02m/s$. The extracted measurements are processed as described in Subsection 3.6.3 and fed to the MORRF. The prediction estimate is the average of the 5 pushes.

The results for mass, inertia, CoM x dimension and CoM Y dimension are shown in Figs. 3.19, 3.20, 3.21 and 3.22, both for the analytical method and the MORRF. It can be seen that the analytical method has lower error means over the whole dataset in the estimation of mass and y dimension of CoM. The proposed MORRF algorithm yields lower estimation means for the inertia and the x dimension of CoM. The MORRF also yields high standard deviation on inertia, suggesting that more pushes may be required for a better estimation result. The analytical method results in high error in the inertia estimation, because in the V-REP simulation the 2-fingered end-effector can not sufficiently rotate the object so as to get a richer measurement for prolonged time without losing contact on one finger. Overall, the comparison experiment shows the difficulty of replicating the controlled environment that analytical methods require, and the resulting lack of versatility. The MORRF method can reach overall performance comparable to the analytical method, while being trained on a different dataset of cubes. The data-driven approach in inertial parameter estimation shows overall promising results for future extensions.

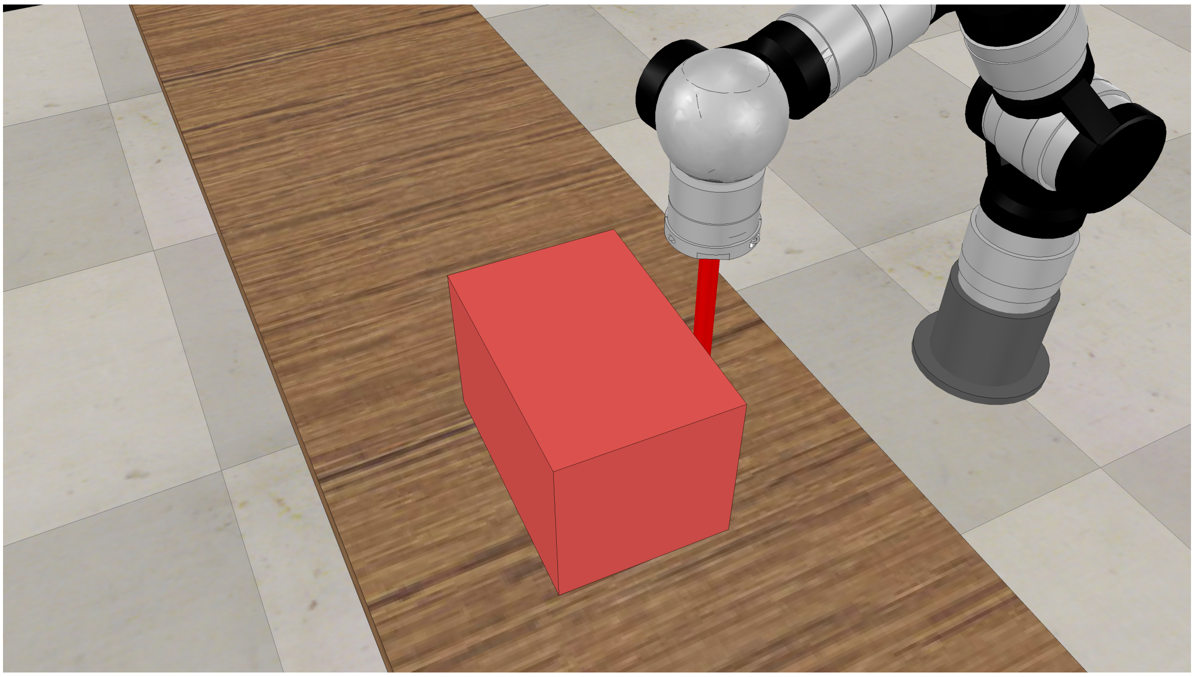


Figure 3.18: The robot pushing a cube for the comparison experiment.

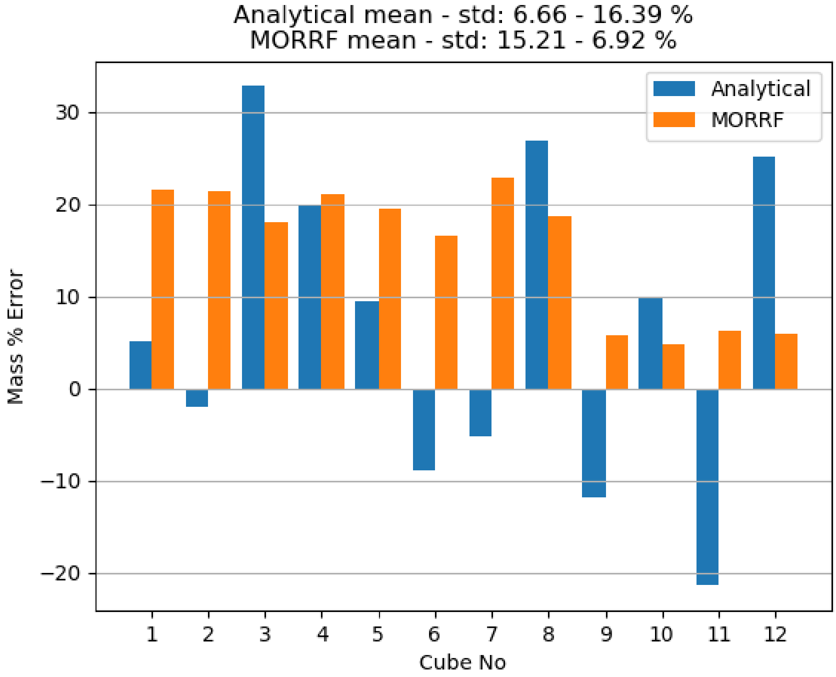


Figure 3.19: Estimation errors for the mass of the 12 simulated cuboids.

3.8 Discussion

This chapter presented a first attempt to formalise and conduct an experiment in data-driven estimation of inertial parameters by robot pushing. Such method is attempted for the first time in literature. The inertial dataset is introduced as a testbed for data extraction and evaluation. A set of 30 simulated cubes with different physical

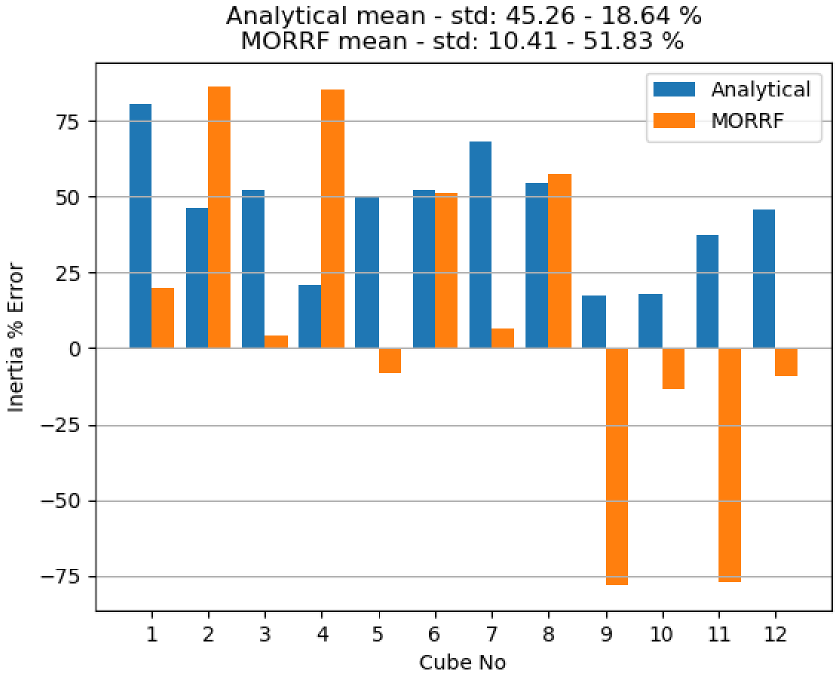


Figure 3.20: Estimation errors for the inertia of the 12 simulated cuboids.

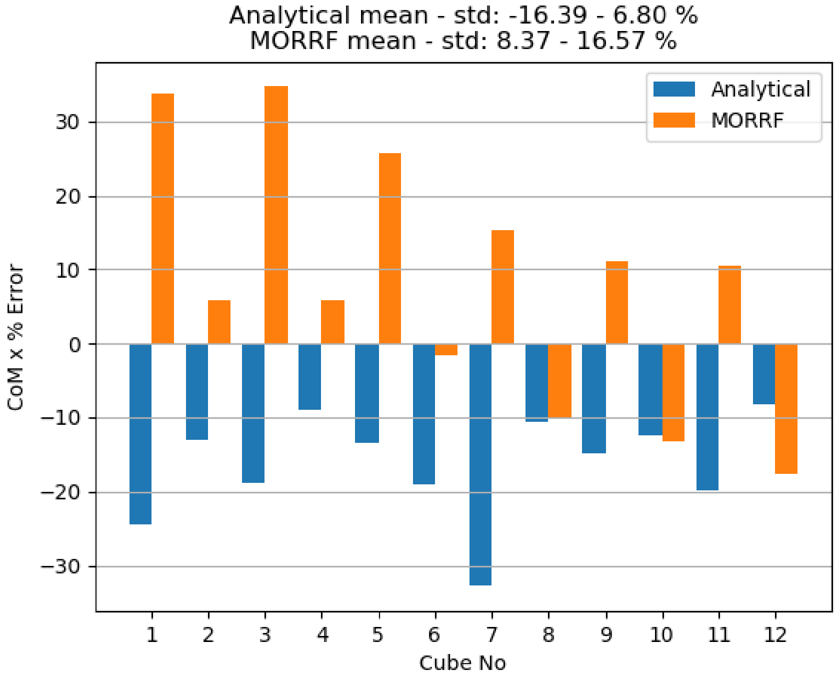


Figure 3.21: Estimation errors for the x CoM dimension of the 12 simulated cuboids.

properties was created, and 10 pushes at each cube were performed. The measured applied wrenches and cube displacements were used to train a random forest for regression. The results show that the performance is better on objects that match the training set characteristics.

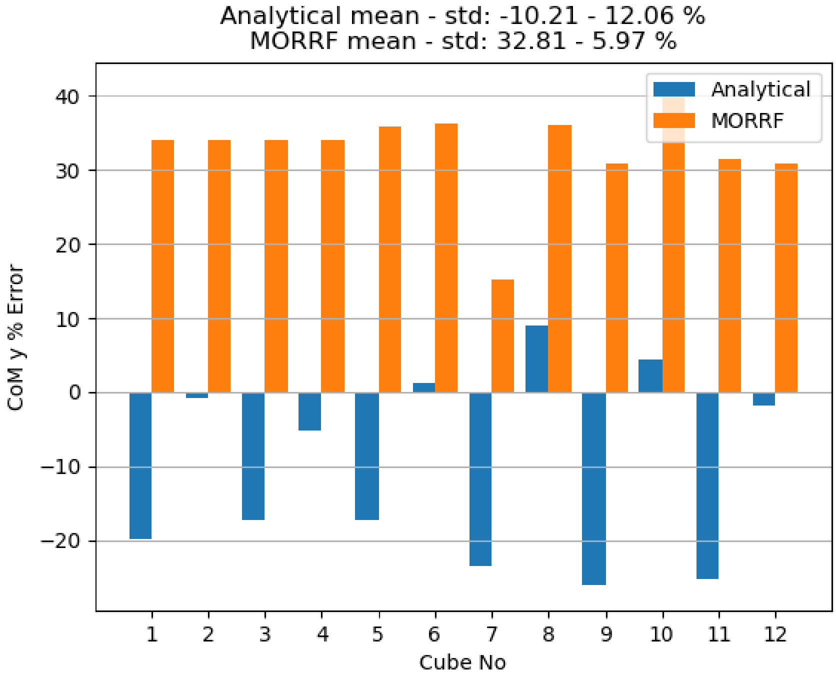


Figure 3.22: Estimation errors for the y CoM dimension of the 12 simulated cuboids.

The presented experiment suggest that the data-driven pushing estimation is possible if the methodology is carefully designed, as it is the case with all learning approaches. The data collection process needs to include large ranges of object properties, and larger amounts of data need to be collected. The features from the measurements need to be descriptive and characteristic of the size they represent. Different learning algorithms can be implemented on the data, and the training and testing process need to be refined. These issues were addressed in the large-dataset experiments that followed.

The results from the experiments in both large-datasets suggest that the learning method can accurately describe the quasi-static motion of an object while pushed by a robot finger. The MORRF can accurately capture the non-linearities of Eqs. (3.8) and (3.10), leading to quite accurate predictions of the inertial parameters.

The simulated dataset was extracted because it enables easier testing on a larger variety of objects with large range of inertial properties. The number of pushes the M-Dataset contains is enough for training and testing, however the objects provided have low ranges on their inertial parameters, and generation of extra objects was necessary.

One of the main advantages of the presented method is the robust performance under variations of the surface friction coefficient. In real conditions, achieving exact quasi-static motion can be very difficult. It requires constant application of a crucial velocity, just enough to break the object's rest but not actively accelerate it. A good approximation of ideal quasi-static motion can be achieved for most objects by applying a very low pushing velocity. Even then, the motion of the object can be jittery, due to the object momentarily sticking on the surface and being pushed again. For this reason the measured sensor signals were divided in 3 windows, and the mean, standard deviation and RMS value of each window were extracted as features. This leads to a smoothing on the signal jitter, reducing the uncertainty in quasi-static motion.

An obvious step when using two datasets is cross-testing, i.e. training on one dataset and testing on the other. The preferred option would be to train on the simulated dataset and test on the M-Dataset. An initial attempt resulted in model performance with error rates of 40-60%. The reasons for the poor performance are inherent to sim-to-real approaches. The frictional loads applied on the object from the surface can be controlled and measured in simulation, but they are inaccurate. On the other hand, in a real dataset frictional loads are physically accurate but difficult to measure. The control over the cross-dataset friction coefficients and the pressure distribution is also another parameter that makes sim-to-real testing difficult. The experimental conditions, such as pushing starting and ending points and sensor frequencies also affect the performance and are difficult to replicate. The goal for this Chapter is to prove and test that the quasi-static pushing mechanics can be incorporated in a data-driven estimation model and not to provide a complete fit-for-all system. For that reason, the cross-testing analysis is left out of the Chapter's scope. Solving the cross-testing problem would open new frontiers in sim-to-real estimation and enable a system to be trained in simulation and deployed in the real world.

Finally, a comparison simulation was set up between the analytical method in Yu et al., 2005 and the large-dataset MORRF. The comparison suggests that both approaches have strengths and drawbacks and yield comparable results, which further solidifies the goal of the Chapter to establish a data-driven estimation method with similar performance with existing analytical ones.

Chapter 4

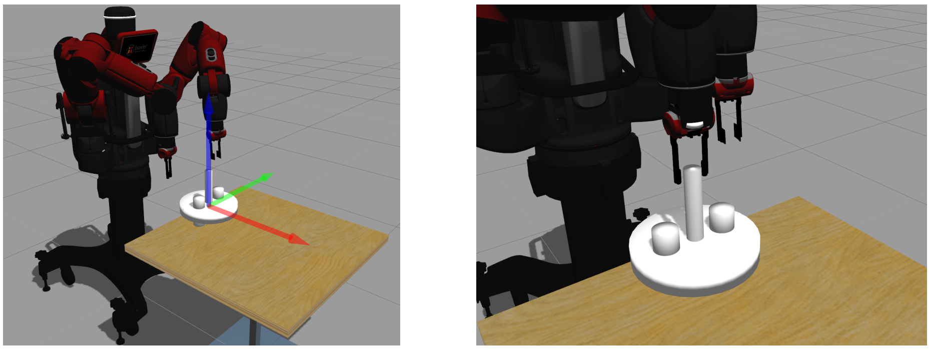
Exploitation of Objects' Inertial Parameters in Efficient Robot Manipulation

4.1 Introduction

In Chapter 2, it was established that there is a need for estimation algorithms to provide accurate values of the inertial parameters, so that they can be used as a resource in autonomous robotics. Numerous works were presented, that use the inertial parameters in different tasks, such as generating robot grasps for unknown objects, as well as incorporating them in dynamic models for more accurate control. In addition, many different approaches have been proposed for achieving autonomous grasping, based on various properties of the object to be grasped. The majority of the grasping literature focuses on the object's geometry or visual appearance, typically derived from 2D RGB camera images or 3D point cloud images, e.g. Kopicki et al., 2015; Saxena et al., 2008.

In this thesis the focus was placed in a novel research area between grasping and manipulation, namely task-informed grasping. Task-informed grasping is the process of evaluating a set of given robot grasps on an object, and selecting the grasps that minimise a manipulation criterion, under a known task to be executed.This chapter explores the use of the object's mass distribution and inertia tensor, in order to help choose an appropriate location on the object for grasping. Unlike the vast majority of the previous grasping literature, this addresses the problem of choosing grasps which improve the robot's ability to move the object along a desired path *after* a successful grasp has been achieved. Specifically, the availability of a state-of-the-art grasp planner such as the one in Kopicki et al., 2015 is assumed, which uses a machine learning approach to output a number of different possible grasps, based on observed 3D geometry of the object, coupled with prior knowledge of the surface geometries that are suitable for grasping by a particular robot hand. Given a variety of grasps proposed by such geometry-based planners, this chapter shows how to select *one* of those grasps which will minimise the motor torques, needed for the robot arm to move the object along its post-grasp trajectory.

Dexterous grasping and manipulation are one of the interesting characteristics which distinguish primates (especially humans) from most other species. Humans are inherently capable of learning sophisticated grasping and manipulation skills (Castiello, 2005), and as described in Chapter 2, they are able to intuitively exploit



(a) (b)

Figure 4.1: Simulation experiment. The Baxter robot grasps a non-uniform, disk-like object, (a). The robot uses its two-finger parallel-jaw gripper to vertically grasp (from above) the tall, thin, cylindrical "handle" attached to the centre of the disk, (b). Two cylindrical weights are attached to the disk, which cause the mass and inertia of this object to exert different torques on the robot, when it is grasped in different orientations. The local coordinate frame is attached to the grasped object's CoM.

the inertial parameters of objects in their everyday manipulations. They do this for a number of tasks, such as grasping and perception of geometrical properties.

Despite the extensive use by humans (and presumably other animal species also) of mass and inertia information for grasping, comparatively little robotics literature explores grasp planning based on the inertial properties of the grasped object. In contrast, this chapter exploits information about the mass and inertia of the object, to choose the grasp that is best in terms of minimising the joint-torques needed for the robot to move the object along a desired post-grasp trajectory. To do so, it is shown how an augmented dynamic model (Khatib, 1987) can be used to combine the dynamics of both the robot and the grasped object within an operational space formulation. This augmented dynamics model is used to estimate the joint torques needed to move the object along a desired post-grasp trajectory. Based on these predicted joint torques, the robot can choose one out of many (geometrically) possible grasps, for achieving post-grasp trajectories with minimal effort by the robot arm's motors.

4.2 Operational space trajectory of a task

Let the point ${}^{gc}\mathbf{x} \in SE(3)$ as the operational point, attached to the end-effector, which will come into contact with the grasped object, once a successful grasp is achieved. $SE(3)$ denotes the group of 3D poses (3D position and 3D orientation). Operational space trajectory ${}^{gc}\zeta$ refers to a vector of successive poses, of a frame attached to this point, defining a desired trajectory for the object. The world reference frame is noted with by ${}^{r}X = \{{}_{r}O, {}_{r}x, {}_{r}y, {}_{r}z\}$. A trajectory to be followed by the manipulated object implies that the local frame ${}^{c}X$, attached to the CoM of the object, follows a sequence of poses:

$$\begin{gathered} {}^{c}\zeta = {}^{c}\mathbf{x}(t) \\ 0 \leq t \leq \mathrm{T} \end{gathered} \tag{4.1}$$

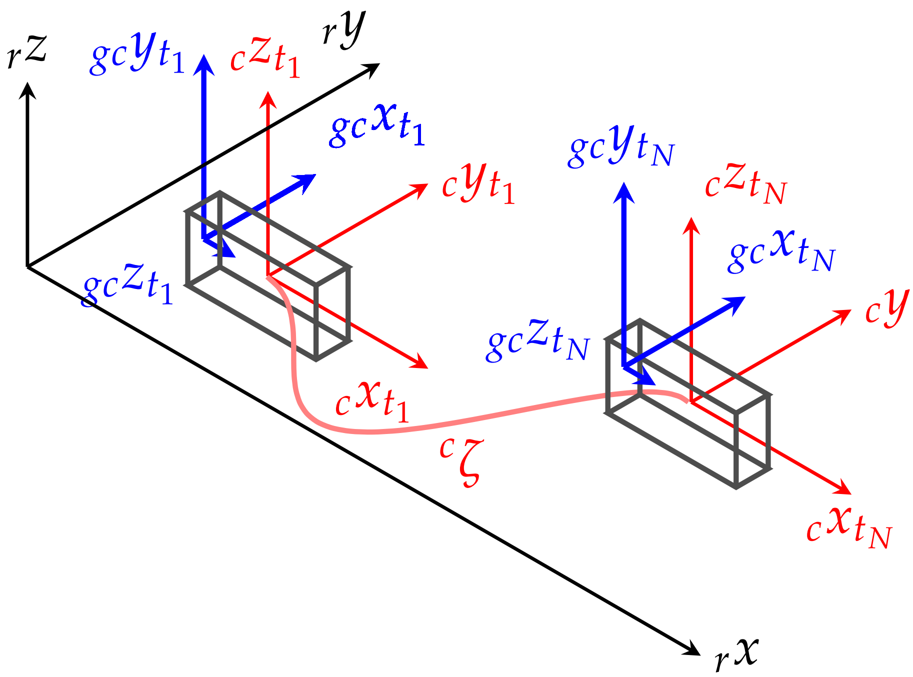


Figure 4.2: An object in the global coordinate frame ${}^r\boldsymbol{x} = \{{}_rO, {}_rx, {}_ry, {}_rz\}$, shown in black. A local coordinate frame ${}^c\boldsymbol{x} = \{{}_cO, {}_cx{}_cy, {}_cz\}$ is attached to the center of mass of the object, shown in red color. This frame follows a trajectory ${}^c\boldsymbol{\zeta}$ during manipulation. ${}^c\boldsymbol{x}_{t_1} = \{{}_cO_{t_1}, {}_cx_{t_1}, {}_cy_{t_1}, {}_cz_{t_1}\}$ and ${}^c\boldsymbol{x}_{t_n} = \{{}_cO_{t_n}, {}_cx_{t_n}, {}_cy_{t_n}, {}_cz_{t_n}\}$ denote this frame at the initial and terminal point of the manipulation trajectory with the corresponding frame of grasp candidate ${}^{gc}\boldsymbol{x}_{t_1} = \{{}_{gc}O_{t_1}, {}_{gc}x_{t_1}, {}_{gc}y_{t_1}, {}_{gc}z_{t_1}\}$ and ${}^{gc}\boldsymbol{x}_{t_n} = \{{}_{gc}O_{t_n}, {}_{gc}x_{t_n}, {}_{gc}y_{t_n}, {}_{gc}z_{t_n}\}$ shown with blue color.

where t denotes a particular time during the motion, and T is the total time that the robot needs to complete the desired manipulation task[1]. ${}^c\mathbf{x}(t)$ determines a complete pose of the grasped object at time t. Although there is a variety of different possible representations of orientation, for the sake of simplicity here the conventional transformation matrix is used.

Consider a local frame ${}^c\mathbf{x} = \{{}_cO, {}_cx, {}_cy, {}_cz\}$. This frame can be described by a transformation matrix[2] from the global reference frame $\{{}_rO, {}_rx, {}_ry, {}_rz\}$ into the local frame $\{{}_cO, {}_cx, {}_cy, {}_cz\}$:

$$\begin{gathered} {}^c\mathbf{x}(t) = {}^c_rT(t) \\ {}^c_rT(t) = \left[\begin{array}{cc} R_{3\times3}(t) & \boldsymbol{d}_{3\times1}(t) \\ \hdashline 0_{1\times3} & 1 \end{array}\right]. \end{gathered} \tag{4.2}$$

Given a vector of times, $\{t_1, ..., t_n\}$, a discrete-time trajectory, corresponding with Eq. (4.1) is represented by a sequence of homogeneous transformations:

$${}^c\zeta = \{{}^c_rT_1, {}^c_rT_2, ..., {}^c_rT_n\} \tag{4.3}$$

[1] Throughout this chapter, $Y(t)$ denotes a continuous function of time, where Y_i is a corresponding value of $Y(t)$ at time $t_i \ \forall\ i = 1, ..., n$, where $t_1 = 0,\ t_n =$ T and $0 \leq t_i \leq$ T denotes discrete sampling time. Y_t is also used as a shorthand of $Y(t)$ where necessary. ${}^*\boldsymbol{\zeta}(t)$ and ${}^*\zeta(t)$ are continuous and discrete trajectory of poses of a frame attached to point $*$ of object in Figs. 4.2 and 4.3.

[2] In general, ${}^{(..)}_{(.)}T \in \mathbb{R}^{4\times4}$ denotes a transformation matrix from local frame (.) into local frame (..).

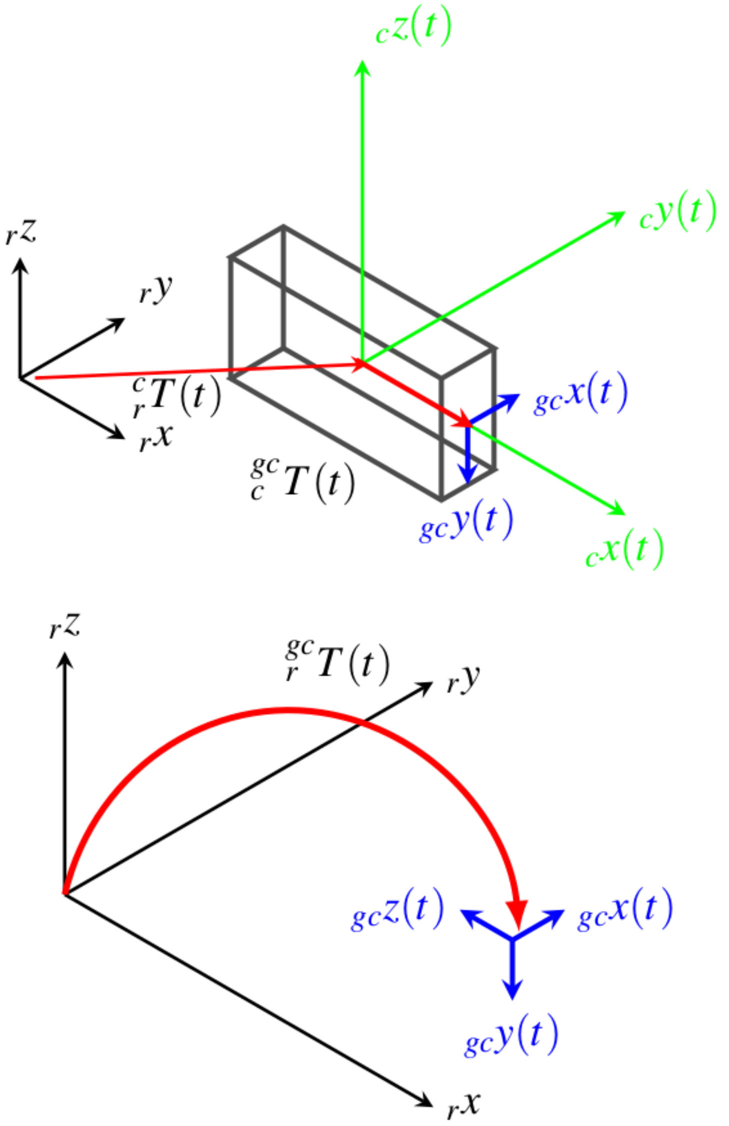


Figure 4.3: Top: a non-deformable object is shown in the global frame. At time t, ${}^c\boldsymbol{x}(t) = \{{}_c o(t), {}_c x(t), {}_c y(t), {}_c z(t)\}$ is attached to the centre of mass of the object to be manipulated. At every time, a frame attached to the object ${}^{gc}\boldsymbol{x}(t) = \{{}_{gc} o(t), {}_{gc} x(t), {}_{gc} y(t), {}_{gc} z(t)\}$ can always be expressed by a single homogeneous transformation ${}^{gc}_{c}T$ from ${}^c\vec{x}(t)$ into ${}^{gc}\boldsymbol{x}(t)$. Note that ${}^{gc}_{c}T$ is not a function of time t. Transformation from ${}^{gc}\boldsymbol{x} = \{{}_{gc} o, {}_{gc} x, {}_{gc} y, {}_{gc} z\}$ fixed on a point of the object to be manipulated into ${}^c\boldsymbol{x} = \{{}_c o, {}_c x, {}_c y, {}_c z\}$ and a transformation from ${}^c\boldsymbol{x} = \{{}_c o, {}_c x, {}_c y, {}_c z\}$ into reference frame ${}^r\boldsymbol{x} = \{{}_r o, {}_r x, {}_r y, {}_r z\}$; Bottom: a total transformation from ${}^r\boldsymbol{x} = \{{}_r o, {}_r x, {}_r y, {}_r z\}$ into ${}^{gc}\boldsymbol{x} = \{{}_{gc} o, {}_{gc} x, {}_{gc} y, {}_{gc} z\}$.

where, ${}^c_r T_i$ is the transformation matrix at time t_i, with corresponding rotation $R_i = R(t_i)$ and translation $\boldsymbol{d_i} = \boldsymbol{d(t_i)}$. The corresponding local frames are ${}^c X_i = {}^c\mathbf{x}(t_i)\ \forall\ i = 1, ..., n$, where, $t_1 = 0$, $t_n = \mathrm{T}$. It should be again noted that the object is considered rigid. Consider also a local frame attached to the robot end-effector, which is the "operational point", by ${}^{gc}\mathbf{x} = \{{}_{gc} O, {}_{gc} x, {}_{gc} y, {}_{gc} z\}$. Because the object is rigid, any candidate robot end-effector pose can be expressed by a fixed transformation matrix ${}^{gc}_{c}T$

from ${}^{c}\mathbf{x}$ into ${}^{gc}\mathbf{x}$, as in Fig. 4.3:

$$\begin{aligned} {}^{gc}\mathbf{x} &= {}^{gc}_{c}T^{c}\mathbf{x}, \\ \forall\, {}^{i_{gc}}\mathbf{x} &\in {}^{gc}\mathscr{X}, \\ i_{gc} &= 1, ..., n_{gc} \end{aligned} \tag{4.4}$$

where ${}^{gc}\mathscr{X}$ is a set of candidate wrist poses. Such poses can be computed by a variety of well known grasp planning algorithms (e.g. Ding et al., 2001; Ferrari et al., 1992; Kopicki et al., 2015; Miller et al., 2004; Saxena et al., 2008, or other grasp-planners as the user prefers). The current state-of-the-art grasp planners, typically generate a set of various possible stable grasp configurations. In the works by Kopicki et al., 2015; Saxena et al., 2008 these are based on learned relationships, between features of the object's surface geometry and appropriate configurations of various parts of the robot hand. Alternatively, a set of potential stable grasps can be computed using grasp simulation software (Miller et al., 2004), force-closure analysis (Ferrari et al., 1992), or form closure (Ding et al., 2001).

Using ${}^{c}\boldsymbol{\zeta}$ in Eq. (4.1), the trajectory of poses followed by ${}^{c}\mathbf{x}$, and Eq. (4.4), the trajectory of ${}^{gc}X$ is written as follows:

$$\begin{aligned} {}^{gc}\boldsymbol{\zeta} &= \{{}^{i_{gc}}_{c}T^{c}\mathbf{x}_1, {}^{i_{gc}}_{c}T^{c}\mathbf{x}_2, ..., {}^{i_{gc}}_{c}T^{c}\mathbf{x}_n\} \\ {}^{gc}\zeta &= {}^{i_{gc}}_{c}T^{c}\zeta, \end{aligned} \tag{4.5}$$

Consider ${}^{gc}\zeta \in {}^{gc}\mathscr{Z}$, where ${}^{gc}\mathscr{Z}$ denotes a group of trajectories corresponding with both the task of manipulating the object $\mathscr{O}$ in Fig. 4.2 and different grasp candidates $\forall\, {}^{gc}\mathbf{x} \in {}^{gc}\mathscr{X}$ in Fig. 4.3. Given a grasp candidate ${}^{gc}\mathbf{x}$ and object's CoM ${}^{c}\mathbf{x}$, one can readily compute ${}^{gc}_{c}T$. Therefore, ${}^{gc}\mathscr{Z}$ can be compactly represented by:

$$ {}^{gc}\mathscr{Z} = {}^{gc}_{c}\mathscr{T}^{c}\boldsymbol{\zeta} \tag{4.6}$$

where, ${}^{gc}_{c}\mathscr{T}$ is a group of transformation matrices from ${}^{c}\mathbf{x}$ into ${}^{gc}\mathscr{X}$.

4.3 Manipulator dynamics under load

To keep the notation simple, from this section onwards the grasping point ${}^{g}c\mathbf{x}$ will be simply noted as $\mathbf{x}$. In this chapter the dynamic model of the robot is assumed known in advance, and it is expressed in the robot joint space. Here, the interest lies in choosing one of many grasp candidates, which will result in minimum post-grasp effort of the robot when executing the desired post-grasp trajectory. To find a grasp which results in minimal effort, an augmented equation of motion is computed, i.e. a combined equation of motion for both the robot and its grasped object, in the robot's joint space.

Murray et al., 1994 define the joint space dynamic model of an n-degree-of-freedom (DOF) manipulator:

$$M(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + C(\dot{\boldsymbol{\theta}}, \boldsymbol{\theta}) + N(\boldsymbol{\theta}) = \boldsymbol{\tau} \tag{4.7}$$

where $\boldsymbol{\theta} \in \mathbb{R}^n$ and $\boldsymbol{\tau} \in \mathbb{R}^n$ are the vectors of joint positions and joint torques, respectively, and $M(\boldsymbol{\theta})$ is the manipulator inertia matrix.

$$C_{ij}(\dot{\boldsymbol{\theta}}, \boldsymbol{\theta}) = \frac{1}{2} \sum_{k=1}^{n} \left(\frac{\partial M_{ij}}{\partial \boldsymbol{\theta}_k} + \frac{\partial M_{ik}}{\partial \boldsymbol{\theta}_j} - \frac{\partial M_{kj}}{\partial \boldsymbol{\theta}_i} \right) \dot{\boldsymbol{\theta}}_k \tag{4.8}$$

represents the Coriolis and centrifugal force terms.

$$N(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \frac{\partial V}{\partial \boldsymbol{\theta}} \tag{4.9}$$

defines a gravitational force term, where $V(\boldsymbol{\theta})$ is potential energy due to gravity. The governing equation of motion in operational space is well studied in pioneering work (Khatib, 1995). The operational point is defined as a point of interest in which the inertia properties are evaluated. The operational coordinate $\mathbf{x}$ is a vector defining the 6-DOF pose of the robot's end-effector. The dynamics of the robot in operational space are represented using this operational coordinate $\mathbf{x}$ as follows:

$$\Lambda(\boldsymbol{\theta})\ddot{\mathbf{x}}(t) + \mu(\dot{\boldsymbol{\theta}}, \boldsymbol{\theta})\dot{\mathbf{x}}(t) + \pi(\boldsymbol{\theta}) = F(t) \tag{4.10}$$

where:

$$\Lambda = J^{-T}(\boldsymbol{\theta}) M(\boldsymbol{\theta}) J^{-1}(\boldsymbol{\theta}), \tag{4.11}$$

$F = J^{-T}(\boldsymbol{\theta})\boldsymbol{\tau}$, $\mu(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$, $\pi(\dot{\boldsymbol{\theta}})$ are the applied end-effector force caused by the robot actuating torques, the gravitational and Coriolis terms in operational space, and $J(\boldsymbol{\theta})$ is the robot's Jacobian. For further details refer to the classic work of ibid.

The joint space dynamic equation of motion of the robot needs to be determined, "augmented" by incorporating the dynamics of the grasped object. Let M_g denote the generalised inertia matrix of an object.

$$M_g = \begin{pmatrix} mI_{3x3} & 0 \\ 0 & I_{CoM} \end{pmatrix} \tag{4.12}$$

where m and I_{CoM} denote the object's mass and inertia tensor w.r.t. the CoM. This inertia tensor can be expressed in a frame attached to the end-effector of the robot as follows:

$$M_o(\mathbf{x}) = E^{-T}(\mathbf{x}) M_g E^{-1}(\mathbf{x}) \tag{4.13}$$

where $E(\mathbf{x})$ is the matrix transforming the linear and angular velocities of the object's CoM to generalised velocities in the frame attached to the end-effector. A trajectory of the operational point attached to the end-effector is:

$$\boldsymbol{\zeta} = \mathbf{x}(t) \tag{4.14}$$

while the object's CoM follows ${}^{c}\boldsymbol{\zeta}$. The augmented dynamics in the end-effector space are:

$$\Lambda_{tot}(\boldsymbol{\theta}) = \Lambda(\boldsymbol{\theta}) + M_o(\mathbf{x}) \tag{4.15}$$

and when projected to the joint space:

$$M_{tot}(\boldsymbol{\theta}) = J^T(\boldsymbol{\theta})\Lambda_{tot}(\mathbf{x})J(\boldsymbol{\theta}) = J^T(\boldsymbol{\theta})\Lambda(\boldsymbol{\theta})J(\boldsymbol{\theta}) + J^T(\boldsymbol{\theta})M_o(\mathbf{x})J(\boldsymbol{\theta}) = M(\boldsymbol{\theta}) + \tilde{M}_o \tag{4.16}$$

where $\tilde{M}_o(\boldsymbol{\theta})$ is the grasped object's inertia tensor representation in the joint space. The resulting inertia matrices represent the dynamic behaviour of the robot (both in joint space and operational space), augmented with the dynamics (inertia matrix) of the object that has been grasped by the robot.

Next to be computed is the Coriolis term of the augmented robot's dynamic equation of the motion in the joint space in Eq. (4.7). From Eq. (4.8), (4.13) and (4.15) the following holds:

$$C_{tot}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \frac{1}{2}\sum_{k=1}^{n}\left(\frac{\partial M_{tot,ij}}{\partial \boldsymbol{\theta}_k} + \frac{\partial M_{tot,ik}}{\partial \boldsymbol{\theta}_j} - \frac{\partial M_{tot,kj}}{\partial \boldsymbol{\theta}_i}\right)\dot{\boldsymbol{\theta}} \tag{4.17}$$

Finally, the gravitational term of the dynamics of the grasped object in the robot's joint space can be defined using Eq. (4.9).

$$N_{tot}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \frac{\partial V_o}{\partial \boldsymbol{\theta}} \tag{4.18}$$

where $V_{tot} = (m + m_{robot})gh_o$ and h_o can be computed using the forward kinematics of the robot and the transformation matrix, from a frame attached to the end-effector to a frame attached to the CoM, represented by ${}^{gc}_{c}T(t)$ in Fig. 4.3. m_{robot} is the total mass of the robot arm.

4.4 Joint torque minimisation along the post-grasp trajectory

The joint values, velocities and corresponding dynamics terms, including inertia, Coriolis and gravity (Eq. (4.7)) of the robot, can be measured while the robot is performing a task. These signals, along with knowledge of the grasped object's mass distribution, (Eq. (4.15)) can be used to compute dynamics terms corresponding to the object inertia as per Eq. (4.13), object Coriolis term as per Eq. (4.17) and object gravity term as per Eq. (4.18)) in the manipulator joint space. The torques, needed at every joint to perform the desired manipulative motion, can be computed using the governing equation of motion of the robot and the grasped object:

$$M_{tot}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + C_{tot}(\dot{\boldsymbol{\theta}}, \boldsymbol{\theta}) + N_{tot}(\boldsymbol{\theta}) = \boldsymbol{\tau}_{tot}(t) \tag{4.19}$$

where M_{tot} can be computed using Eq. (4.16).

At every time step, the L_2 norm of the applied torques is computed. Given a particular object, and a desired manipulation task, the "effort" of a manipulative task is defined as a cost function for characterising the suitability of any candidate grasp for the post-grasp motion. This "effort" metric is computed in terms of the L_2 norm of the joint torques, as follows:

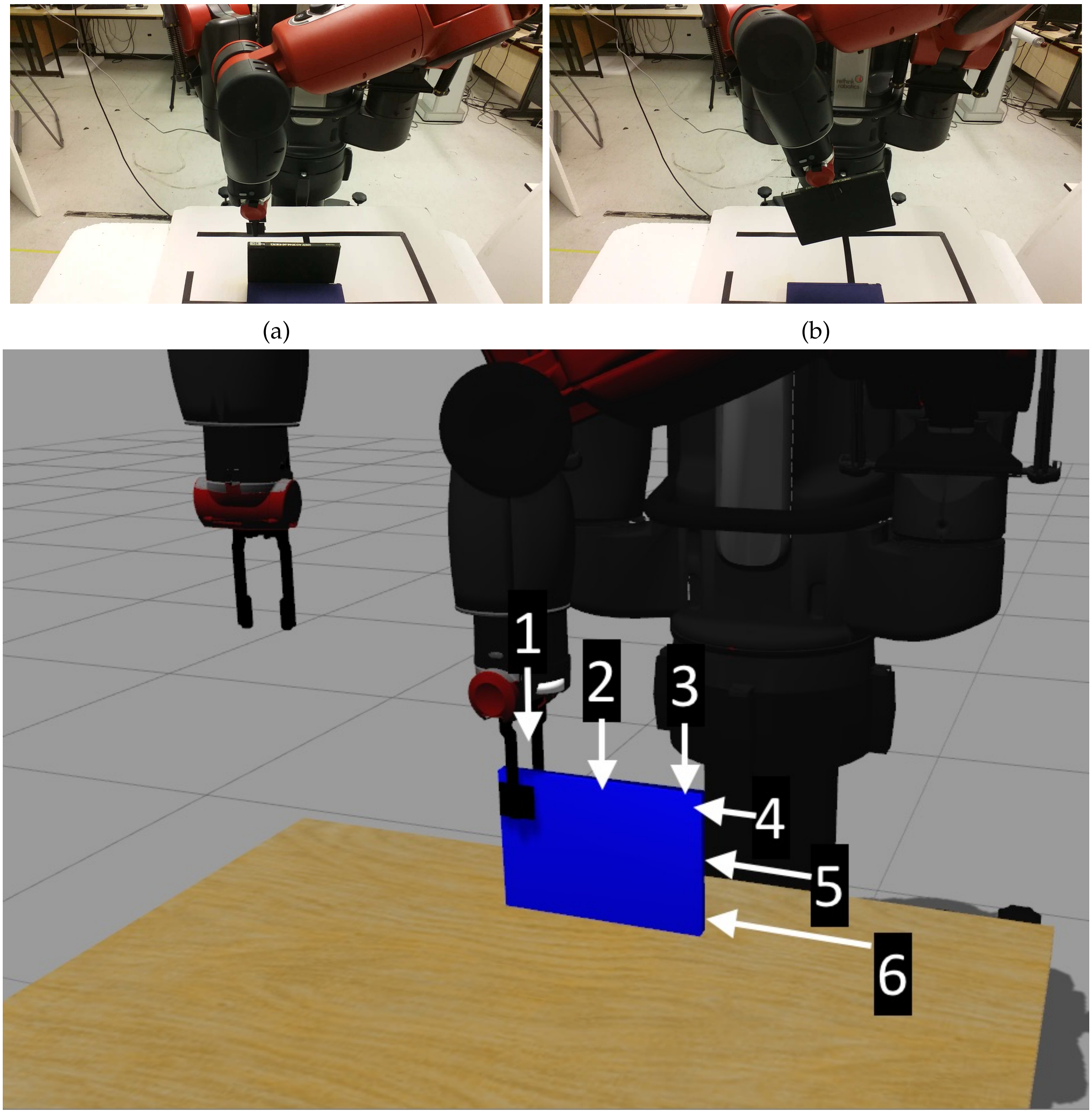
$$\hat{E}(\mathbf{x}) = \sqrt{\sum_{i=0}^{n_T} |\boldsymbol{\tau}(t_i)|^2} \tag{4.20}$$

where $\boldsymbol{\tau}(t_i)$ is a torque at time t_i computed using Eq. (4.19), where $0 \leq t_i < T$, t_i is the i_{th} sampling time, and n_T is the number of samples in the time domain. If Δt is the sampling time and T is the overall time to complete the task, then $n_t = \frac{T}{\Delta t}$. $\hat{E}$ represents the overall effort, in terms of the torques which a robot uses to move the grasped object along a commanded post-grasp manipulation trajectory, corresponding with a candidate grasp $\mathbf{x}$. The goal is to find the grasp that minimises the manipulation effort along the executed task. To sum up, each grasp candidate with known transformation $E(\mathbf{x})$ yields different yields different object dynamics (Eqs. (4.13)), which in turn yield different combined dynamics (Eqs. (4.16)-(4.17)-(4.18)) and corresponding joint torques (Eq. (4.19)). The torques of the combined model are calculated for each point in the known end-effector trajectory, and the effort metric (Eq. (4.20)) is calculated for the whole trajectory. The grasp with the minimum effort metric is selected as the best grasp for the task.

4.5 Experimental results

This section presents the results of experiments using a 7-DOF redundant manipulator, grasping objects of different shapes. Firstly, both simulations and empirical measurements on a real robot are presented, which show that different grasp positions on the same object do indeed lead to significantly different amounts of effort to move that object along a post-grasp trajectory. Secondly, the accuracy of the augmented dynamics formulation combined with a physics simulator is evaluated, regarding its ability to predict post-grasp joint torques, by comparing predicted torques against torques measured empirically during task execution on a real robot. The results show that, even though predicted and measured torques are not identical, they are sufficiently well correlated to enable correct selection of the best grasp to enable the least-effort post-grasp motion. Finally, the grasp selection strategy on objects of two different shapes is demonstrated, and it is shown that the method successfully chooses the best grasp in both cases.

In all experiments, a *Rethink Robotics* Baxter® robot is used. Baxter includes a torso, and two 7-DOF arms. Each arm has seven rotational joints and eight links (including the base). The Baxter Software Development Kit (SDK) is used along with Python Kinematic Dynamic Library (PyKDL) and Gazebo simulation. The experiments were executed in the Gazebo simulation environment, as well as on the real robot. The aim of these experiments is to analyse different grasps for objects with known inertial properties, and compute which, out of a set of possible candidate grasps, would result in the minimum torque effort when moving the grasped object along a desired post-grasp manipulation trajectory. The shown examples consist of choosing grasps for two different objects with very different shapes and inertial properties. For each candidate grasp on each object, a post-grasp trajectory of poses



(a)

(b)

(c)

Figure 4.4: The experimental setup: (a) The robot is approaching the object; (b) The robot performs a primitive rotation movement about x axis; (c) Several possible grasp poses on a book object. The robot is tasked with choosing one of these potential grasps, which minimises the effort of the robot arm's motors when executing a desired post-grasp trajectory.

is computed as per Eq. (4.5) such that the object's CoM follows the corresponding desired movement as per Eq. (4.2), in accordance with discussion presented in section 4.2.

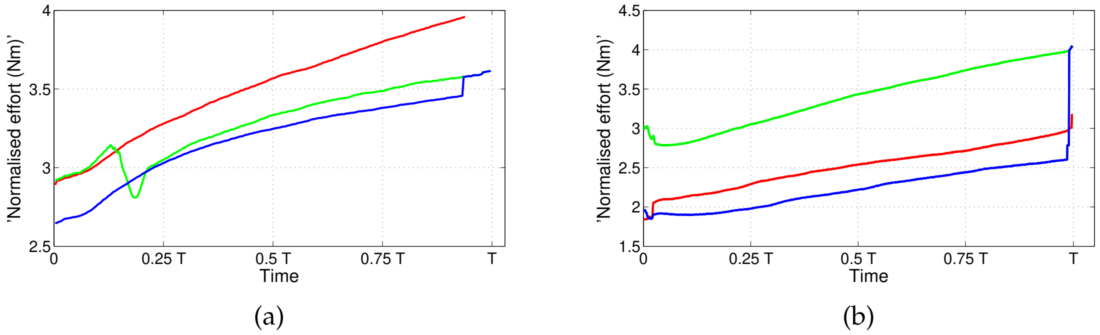


Figure 4.5: Example of total torque signals for the task of rotation about y axis, computed using the augmented dynamic model of object and manipulator. Red, green and blue lines represent the overall torque at each time-step for: (a) First, second and third grasp poses (b) Forth, fifth, sixth grasp poses shown in Fig. 4.4c.

4.5.1 Variation of post-grasp effort with choice of grasp and choice of post-grasp trajectory

This experiment is designed to confirm that different choices of grasp pose, on the same object, will result in significantly different amounts of torque effort being required to perform a post-grasp manipulation. In this experiment, the robot's left arm is used to pick up a $22 \times 15 \times 1.5$ cm hardcover book, weighing 0.34 kg. The book is initially balanced along its long edge, on top of a table in front of Baxter, and six possible candidate grasps are considered on the book, as shown in Fig. 4.4c. Each candidate grasp is associated with a different inertia tensor of the object, and is also associated with a different initial robot pose at the beginning of the post-grasp manipulation. For each choice of grasp, the results of performing six different post-grasp movements were measured . After grasping the book, the robot initially lifts it 10cm above the table, and then performs one of six experimental post-grasp movement primitives. These experimental motion primitives consist of three translations (10cm along each Cartesian axis) and three rotations (20 about each Cartesian axis).

Fig. 4.5 shows the evolution of overall torques at each time-step along a single post-grasp object trajectory (rotation about y axis), given each of the six candidate grasps on the book object. These torque evolutions were estimated using the augmented dynamic model, Eq.(4.19). These results confirm the proposed argument that different choices of grasp lead to significantly different torque efforts required to move the object along a desired post-grasp trajectory.

Fig. 4.6 shows each of the six experimental post-grasp motions. Red, green and blue lines denote translation movements (Fig. 4.6a) and rotation movements (Fig. 4.6b) with respect to x, y, z Cartesian axes respectively. For each motion, Fig. 4.6 plots the overall torque "effort" integrated along the entire trajectory (according to Eq. (4.20)). Solid lines denote effort predicted by the augmented dynamic model, whereas dashed lines denote effort measured directly from the robot's joints during experiments on the real Baxter robot. Note that it is not strictly correct to perform the linear interpolation between each candidate grasp, as shown in Fig. 4.6, however in many cases it is reasonable to expect that a new grasp, placed somewhere between two measured candidate grasps, might yield a torque effort lying somewhere between the efforts needed for each of the measured grasps.

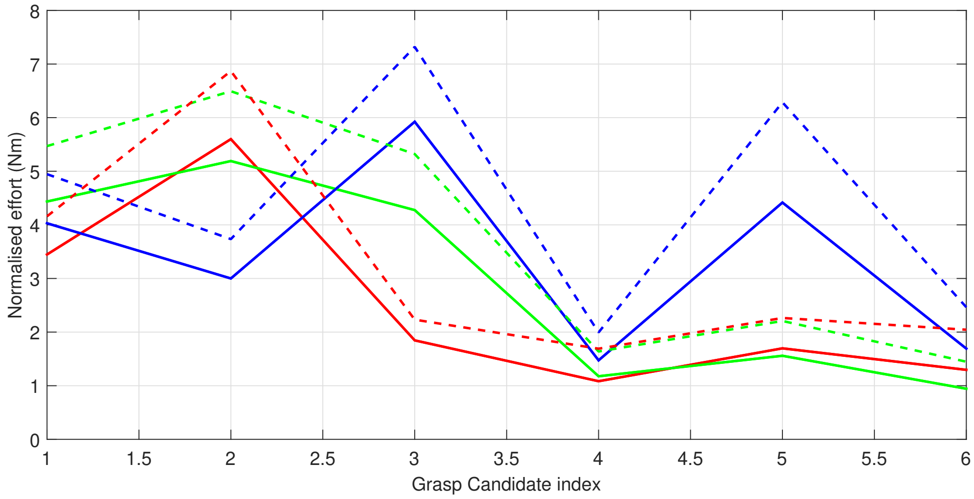
Note that the effort predicted by the augmented dynamic model, does not exactly match the effort reported by querying the real robot during task execution. From these preliminary experiments, it is not clear whether this discrepancy arises as a result of: errors in the torque estimation by the real robot; failure of the dynamics equations to completely model reality; or a combination of both kinds of error. The Baxter robot is provided with Series Elastic Actuators (SEAs), which use position of each joint, along with knowledge of the spring stiffness, to indirectly estimate the torque at each joint. It is possible that some industrial-type robots, which directly measure torques using torque sensors at each joint, might yield different results, shedding light on the above questions. Nevertheless, despite these discrepancies, the predicted effort and effort reported by the robot are still very highly correlated, and are sufficiently similar that it is still possible to use the predicted effort to correctly choose the best (minimum effort) out of a set of possible grasp candidates.

As an example of how the proposed method can be applied, note that the forth grasp pose results in minimum effort of the manipulator while it grasps and translates the book along the three Cartesian axes (Fig. 4.6a). In contrast, the second grasp pose results in minimum effort for post-grasp manipulations that involve rotating the book about the x and z axes, while the sixth grasp is best for achieving post-grasp rotation of the book about the y axis. Clearly, the proposed method can be usefully employed to choose the best grasp, from a set of grasp candidates, in order to execute a particular desired motion of the object after it has been grasped, with minimum effort.

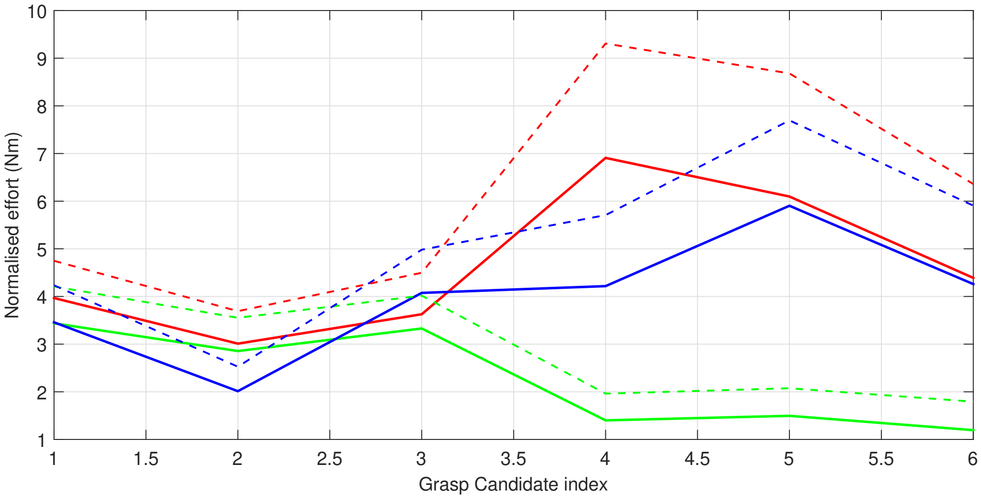
4.5.2 Using the augmented dynamics model to choose an optimal grasp for an oddly shaped object

This section describes the results of an additional experiment performed in simulation. In the simulation, the object to be manipulated is a disk with radius $15cm$, and height $5cm$, shown in Fig. 4.7. Four (short and fat) cylindrical weights are attached to the disk, two on the top surface and two on the bottom surface (one of the bottom cylinders is just visible in Fig. 4.7b). Each of these cylindrical weights has radius $5cm$ and height $3cm$. The disk also has a (tall thin) vertical cylindrical handle attached to its centre, with a radius of $2cm$ and a height of $20cm$. The total structure weighs $0.3kg$.

The robot gripper approaches the handle from above, aligned with the z-axis shown in blue in Fig. 4.7. By rotating the gripper in increments of 20^o about the z-axis (from 10^o to 170^o) the robot is able to generate a uniformly sampled space of nine initial grasping poses. Each such candidate grasp corresponds to a different rotation of the object's inertia tensor with respect to the robot, leading to changes in torque loading and overall augmented dynamics of the robot and its grasped object. Four different post-grasp manipulation tasks were defined. The first task comprises a series of translation motions of the grasped object. The robot first lifts the object $10cm$ in the z direction. Next it translates the object $10cm$ along the y direction. Finally, it translates the object by $10cm$ along both x and y axes simultaneously. The remaining three post-grasp motion tasks comprise primitive rotations about each of the x, y, z Cartesian axes (after first lifting the object through $10cm$ in the z direction). Fig. 4.8 shows the torque effort required to perform the first (piecewise translation) task, for



(a)



(b)

Figure 4.6: Total torque effort, integrated along each of six experimental post-grasp motion trajectories for the book object, for each of six candidate grasps. Vertical axis denotes effort, while horizontal axis denotes the space of possible grasps. Red, green and blue denote translation (a) and rotation (b) with respect to x, y, z Cartesian axes respectively. Solid lines denote efforts estimated by the augmented dynamics model, whereas dashed lines denote efforts measured by querying the real robot's joint sensors during task execution.

each grasp candidate. For this task, the grasp oriented at 50^o about the z-axis will result in minimal post-grasp effort.

Fig. 4.9 shows the effort required to perform primitive rotation tasks about each Cartesian axis. The 10^o grasp is best for x rotation, the 110^o grasp for y rotation and the 150^o grasp for z rotation.

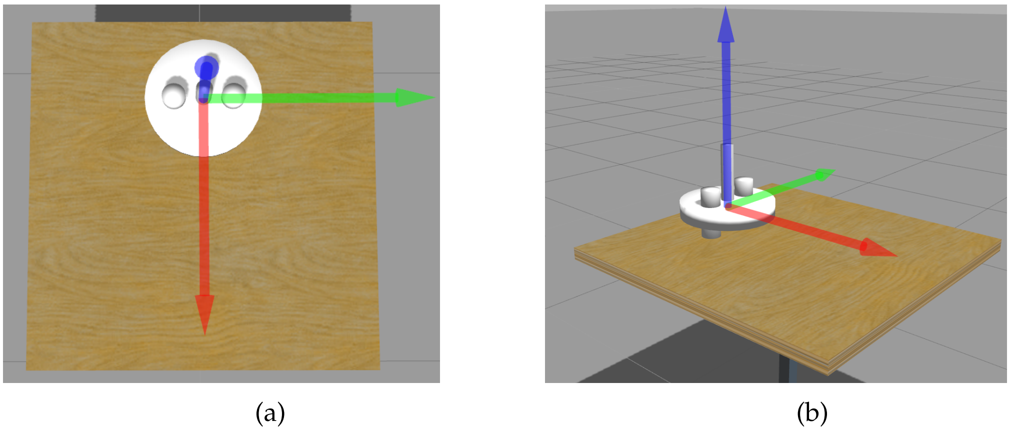


Figure 4.7: Oddly shaped disk object, with attached cylindrical weights and vertical handle. The object is shown in its initial pose prior to manipulation. (a) Birds-eye view from above. (b) Perspective view from the side. The robot gripper approaches the handle from above, aligned with the *z*-axis shown in blue. The robot samples nine possible candidate grasps, by rotating the gripper in 20^o increments about the *z*-axis (from 10^o to 170^o). The robot is tasked with deciding which of these nine grasps will result in the minimum torque effort needed to move the object along a desired post-grasp trajectory.

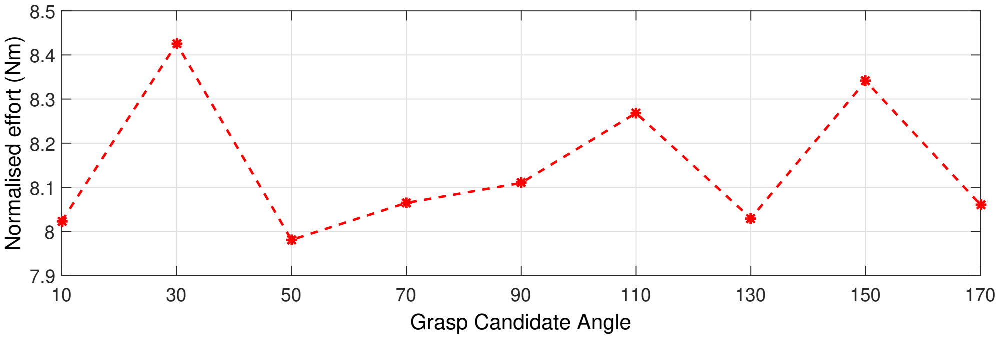


Figure 4.8: Overall torque effort that each grasp causes during first post-grasp manipulation task (piece-wise translation), for the oddly shaped object shown in Fig. 4.7. Vertical axis denotes overall effort during the post-grasp motion. Horizontal axis denotes the grasp space, with successive grasp candidates separated by 20^o increments

4.6 Discussion

Both simulation data and empirical measurements on a real robot, confirm that different stable grasp positions on an object do indeed lead to significantly different amounts of effort to move that object along a post-grasp trajectory. In addition, these results suggest that the augmented dynamics representation of the robot and its grasped object, combined with a physics simulator, do not exactly predict the resulting joint torques (as measured empirically using torque readings from the real robot's sensors), Fig. 4.6. However, predicted torques and real torques do correlate sufficiently well that these predictions are able to correctly choose the best grasp for minimising post-grasp robot effort. Where a grasp planner, such as Kopicki et al.,

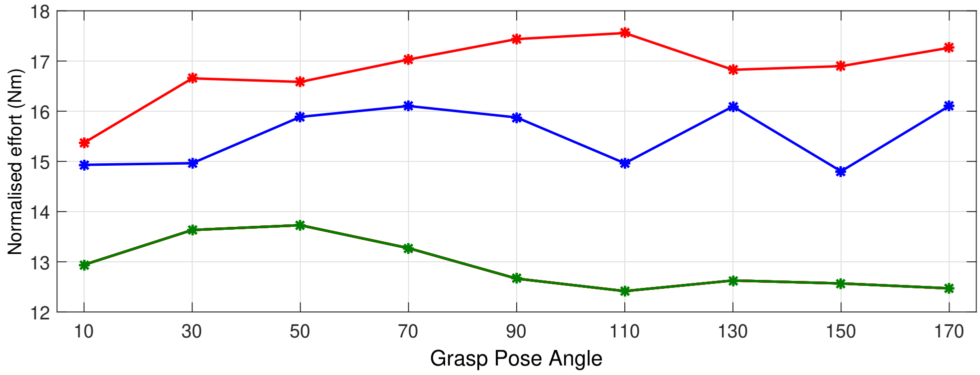


Figure 4.9: Torque effort needed to achieve post-grasp rotations of the grasped object about x, y and z axes, denoted by red, green and blue lines respectively. Vertical axis denotes effort. Horizontal axis represents the space of candidate grasps.

2015, outputs many geometrically feasible grasps, the presented torque-based cost function can be used to plot an "effort map" over the surface of the object.

The results have offered several intuitive and easy to understand examples, in order to illustrate the potential applications of the proposed method. For the rotation about the x-axis task, Fig. 4.9, the best grasp candidate is the one with a 10^0 rotation angle, while the grasp candidate that gives maximum effort is located near the 90^o neighbourhood. Since the heavy cylinders of the object are located along the x-axis, Fig. 4.7a, the robot must exert a greater effort to rotate them about the x-axis when grasping from the 90^o grasp, and minimal effort when grasping from the 0^o angle. For the post-grasp translation tasks, Fig. 4.8, the best grasp candidate for y-axis translation is the 90^o grasp. For the z-axis translation task, the minimal post-grasp effort is achieved by selecting a grasp close to the 0^o angle, because this angle places the end-effector closer to the disk weights, making the rotation easier. It is evident that the proposed method chooses grasps which minimise post-grasp effort in intuitively sensible and physically plausible ways.

The Baxter robot has two 7-DOF arms. Hence, to perform a task in a 6-DOF Cartesian workspace, the robot has a redundant degree of freedom. Although there are always more than one joint trajectories corresponding with an end-effector trajectory, this effect was not studied in the experiments and it can be an interesting future work.

The results of these experiments suggest that different choices of grasp pose, relative to the object frame, do indeed result in significantly different efforts from the robot's motors in executing desired post-grasp object trajectories. In addition, the augmented dynamics model, of the robot and its grasped object, can predict post-grasp joint efforts sufficiently well to enable optimal grasp selection. Finally, the grasps selected by this proposed method are useful in an intuitively meaningful and physically plausible manner. The method could provide a useful way to evaluate raw grasps resulting from grasp planners such as Kopicki et al., 2015.

Chapter 5

Exploitation of Objects' Inertial Parameters in Safe Robot Manipulation

5.1 Introduction

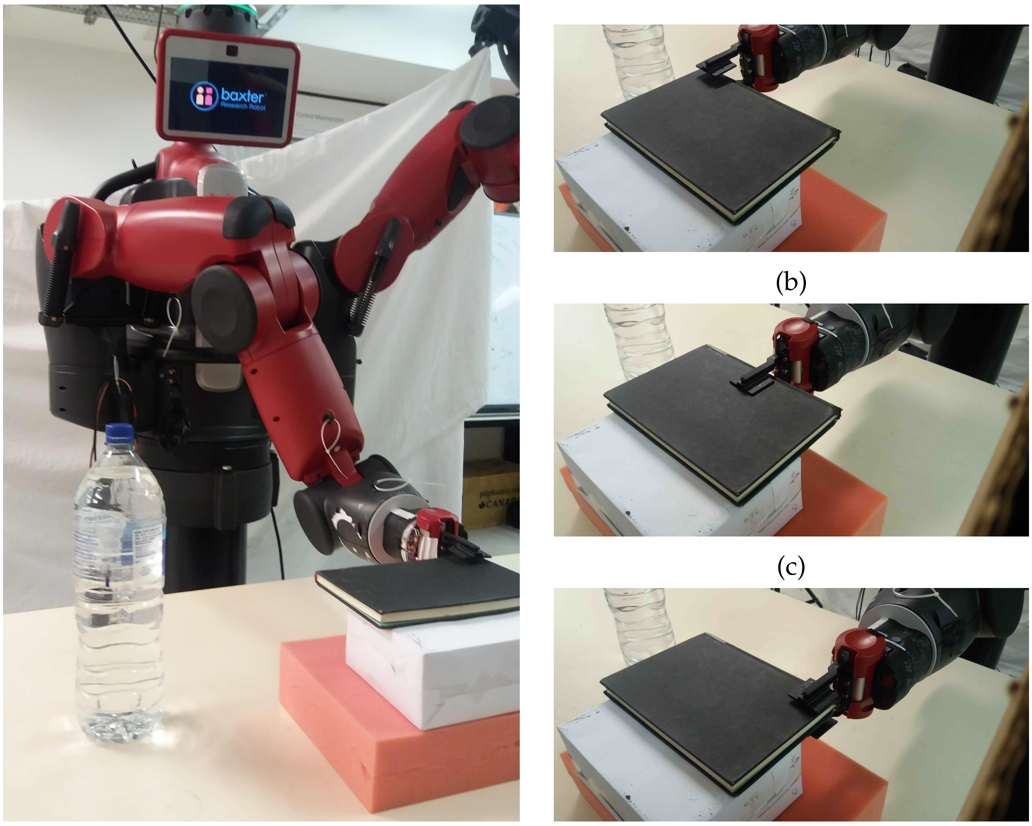
The idea that different grasping points on an object contribute to alterations of the total dynamics under a given task, is a novel idea that has been introduced in this thesis. These alterations are primarily dependent on the inertial parameters of the object, which further solidifies the main research of the thesis. The core idea of the dependence of the total dynamics on the grasping point and inertial parameters, leads to new possibilities in the field of grasp evaluation and selection, and novel manipulation metrics.

The research presented in the previous chapter showcased how the inertial parameters of an object are combined with the dynamics of the robot, and how this combination depends on the location of a specific grasping point on the object. As a result, a robot can reduce the amount of effort it requires for a given task, thus reducing its power requirements. In order to continue the orientation of this thesis towards manipulation in extreme environments, this chapter presents a variation of the grasp selection idea that increases the robot's safety towards its surroundings.

More specifically, this chapter extends previous ideas on "task-informed grasp selection", by proposing a method for choosing grasps that maximise safety during post-grasp manipulations. Such method can be used in dynamic and uncertain environments where collisions are both likely and also safety-critical. Motivating applications include human-robot collaborative working, or remote manipulation in highly cluttered and high-consequence environments such as nuclear decommissioning and robotic surgery (Alambeigi et al., 2011; Wilkening et al., 2017).

Making robotic manipulation safer for humans and surroundings is a domain that is receiving increasing attention. Many approaches have been proposed to tackle this problem. For example, Ikuta et al., 2003 define three categories of robot safety measures, namely collision avoidance, impact minimisation and post-impact suppression methods. A large proportion of the research literature has been devoted to the collision avoidance category, e.g. Flacco et al., 2012 and Beetz et al., 2015. In contrast, in this chapter the focus is on the category of impact minimisation.

Early work on impact minimisation was by Walker, 1990, where the authors described the impact force delivered by the robot's end-effector, as a function of its



(b)

(c)

(a)

(d)

Figure 5.1: Pick-and-place experiment with a real robot. The robot is tasked with grasping and then moving the book (which has same dimensions and mass as used in simulation experiments described later in this chapter). During the post-grasp manipulation, a collision occurs between the grasped object and an obstacle (the water bottle) and force values in the end-effector are measured. The main idea of this Chapter with enabling the robot to choose from several feasible grasps, to minimise impact forces during such post-grasp collisions. In this experiment, the impact forces are measured for each possible grasp on the book. (a) Overall experimental configuration. (b) First grasp choice. (c) Second grasp choice. (d) Third grasp choice.

velocity, type of impact, direction of impact and manipulator posture. They use the manipulator's redundancy to alter its posture in order to minimise the impact force. Impact force minimisation is also studied by Lin et al., 1995, where the authors designed a compliance controller to minimise impact and reduce post-collision bouncing effects. Walker, 1994 and Barcio et al., 1994 defined the impact ellipsoid of a manipulator as the variation of the end-effector impact force w.r.t. variations in the joint velocities. Kim et al., 2000 extended those notations and proposed new impact force measures related to the robot's directional velocity. Furthermore, Heinzmann et al., 2003 defined *impact potential* as a quantification of the maximum impact force a robot can exert in a collision with a stationary object. They implemented controllers to directly control the impact potential. Hu et al., 2016 expressed the impact ellipsoid as a series of inertia quasi-ellipsoids for a space robot and object model, and a pre-impact configuration is designed to inflict minimum impact forces before grasping an object. The ideas of Hu et al., 2016 are related to the work presented in this chapter, in the sense that ibid. use an object's inertial properties to minimise impact *before* grasping. However, this chapter is concerned with understanding inertial properties to minimise impact *after* grasping.

Khatib, 1995 analysed the inertial properties of robotic manipulators and introduced the terms *effective mass and inertia*. These terms were used to describe the mass and inertia felt by the environment during a collision with the robot's end-effector. Kang et al., 2010 minimised the effective mass for the case of a mobile manipulator, by using both the mobile platform's and the manipulator's kinematic redundancies. Haddadin et al., 2012 used variable effective mass, inertia, and robot velocity to approximate collision with human tissues. They generated a database that describes the effects of different collision configurations on human tissue, embedding injury knowledge in the robot's motion planning and control systems. Petersen et al., 2014 and Petersen et al., 2016 used the redundancy of a surgery robot to follow a minimum effective mass and inertia trajectory while under the surgeons control. Another strategy in impact minimisation was presented by Rossi et al., 2015, where the authors designed a robot path controller to constrain the dissipated energy in case of inelastic collision. Finally, Ragaglia et al., 2014 proposed an integrating approach that combined visual and sensory input, as well as minimisation of reflected mass and robot velocity regulation to estimate a severity index when a person is nearby.

While all of the above-mentioned approaches explore ways of making the robot motion safer, they do not consider the safety of the manipulator while it is holding an object. Nonetheless, robots are intended to manipulate objects. Holding an object alters the dynamics of the manipulator and consequently affects the impact forces experienced in a collision. The impact forces on the end-effector will change when holding the same object with different grasp poses, because each grasping pose corresponds to a different transformation of the object's inertia tensor with respect to the grasp frame.

Although there have been many studies on robotic safety during manipulation, they consider the problem of safety during manipulative motion only. In contrast, this work is different in that safety is considered *prior to grasping*, and incorporates notions of post-grasp safety into the grasp selection process. The effects of different grasp pose choices on post-grasp impact force and robot safety have not previously been studied, and previous grasping methods have not incorporated metrics of post-grasp safety into the grasp planning process.

The main contributions of this Chapter are as follows: Extending the results of the previous Chapter, this Chapter provides an analysis of the effect of different object grasping poses on the augmented robot-object dynamics, but now evaluating their performance in the case of post-grasp collisions. A simple but fundamental methodology is developed in an under-explored aspect of robotic safety. A novel grasp selection criterion is proposed and evaluated, that enables a robot to choose a grasp which minimises the resulting impact force in the event of a post-grasp collision with the robot's end-effector.

To show the effectiveness of this approach, a series of empirical experiments using both simulated and real robots are conducted and their results are presented, which evaluate and illustrate how the impact force is related to both the object's inertial properties and the robot's choice of grasp.

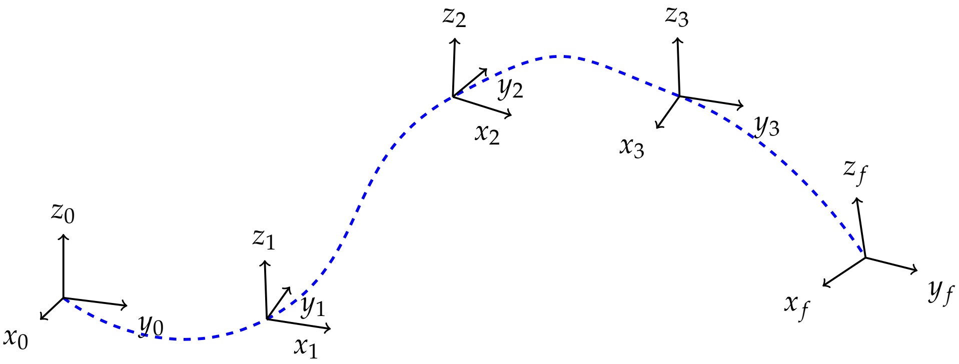


Figure 5.2: End-effector trajectory example. The desired trajectory, which the robot is commanded to follow post-grasp, is considered known prior to grasping. The robot needs to move from an initial 6D pose to a target 6D pose, $\mathbf{x}(0)$ and $\mathbf{x}(t_f)$ respectively, within a finite amount of time $t_f = 2$ sec. A quintic function is selected to represent this trajectory in continuous time. A sampling rate $\Delta t = 0.04$ sec is selected and the end-effector's trajectory is sampled to get a total of $N = \frac{t_f}{\Delta t} = 50$ in-between poses. The effective mass of the robot holding the object is evaluated for each of these N poses.

5.2 Problem formulation

Let $\mathbf{x}(t) \in \mathrm{SE}(3)$ be a pose of the end-effector at time t where $[0 \leq t \leq t_f]$, $\mathrm{SE}(3) = \mathbb{R}^3 \times \mathrm{SO}(3)$ and $\mathrm{SO}(3)$ denotes the group of rotations in three dimensions:

$$\mathrm{SO}(3) = \{\mathrm{R} \in \mathbb{R}^{3\times 3} \ : \mathrm{RR}^T = \mathrm{I},\ det(\mathrm{R}) = \pm 1\}.$$

It is assumed that an initial and a target pose of the object to be manipulated are given (e.g. recognised by a computer vision algorithm). Consequently, the poses of the robot's end-effector trajectory corresponding to those of the object are computed. A rigid object and a fixed grasp are considered, so that the transformation from the object's centre of mass (CoM) to the end-effector is unique for every point of the object's trajectory. For simplicity, all the analysis is conducted directly in the end-effector space. As by Ramos et al., 2013, a polynomial function is used to generate a trajectory of end-effector poses necessary to move the object from the initial to the target pose. For example consider the pick-and-place task of the book shown in Fig. 5.1 which is desired to be placed on the other side of the bottle. The initial and final poses are represented with $\mathbf{x}(0)$ and $\mathbf{x}(t_f)$, where $x(t)$ is a 6D pose vector consisting of Cartesian coordinates for translation and Euler angles for rotation. A quintic polynomial is used to generate the end-effector's trajectory. This polynomial has enough free parameters to ensure that the boundary conditions of position, velocity and acceleration are satisfied while the trajectory and its first and second order derivatives are continuous. An example of a trajectory is shown in Fig. 5.2.

$$\begin{aligned} \mathbf{x}(t) = \mathbf{x}_0 + \mathbf{x}_1 t + \mathbf{x}_2 t^2 + \mathbf{x}_3 t^3 + \mathbf{x}_4 t^4 + \mathbf{x}_5 t^5 \\ \mathbf{x}(0) = x_0,\ \mathbf{x}(t_f) = \mathbf{x}_{t_f} \\ \dot{\mathbf{x}}(0) = \dot{\mathbf{x}}(t_f) = \ddot{\mathbf{x}}(0) = \ddot{\mathbf{x}}(t_f) = 0 \end{aligned} \tag{5.1}$$

The augmented object model, as proposed by Khatib, 1987 is used. The *augmented object model* describes the dynamics of a manipulator while holding an object. In the previous chapter, this augmented model was used to select grasps which minimised the post-grasp torque effort.

Consider the dynamic model of a robotic manipulator in the operational space expressed in Eq. (4.10). While the analytical dynamics may not be known with large detail in real applications, it is reasonable to assume that an approximation of the robot's dynamic model can be identified or known in advance.

$$\Lambda(\mathbf{x})\ddot{\mathbf{x}} + \mu(\mathbf{x}, \dot{\mathbf{x}}) + p(\mathbf{x}) = \mathrm{F} \tag{5.2}$$

where $\Lambda(\mathbf{x}), \mu(\mathbf{x}, \dot{\mathbf{x}})$ and $p(\mathbf{x})$ are the mass matrix, Coriolis matrix and gravity force vector respectively, all expressed in the operational space. F is the generalised force in the operation space including external wrench of force and torques applied to the manipulator from the environment.

While one can compute the force at every point of interest of the manipulator by writing the corresponding operational space equation, the kinetic energy matrix $\Lambda(\mathbf{x})$ can be used to compute the impact force during a collision without the need to solve the second order differential equation in Eq. (5.2). However, this only represents the effect of the manipulator's dynamics on the impact force. According to previous studies in human-robot collaboration and safety, the impact force, as perceived by a human during collision with a manipulator, can be represented by the "effective mass", defined by Khatib, 1995 as:

$$m_{eff} = \frac{1}{\mathbf{v}^T \Lambda_{\mathbf{u}}^{-1}(\mathbf{x})\mathbf{v}} \tag{5.3}$$

where $\mathbf{v}$ is a unit vector in the direction of motion and $\Lambda_{\mathbf{u}}$ is the kinetic energy corresponding with translation only.

The aim is to quantify the impact force of the manipulator while it is moving an object with known inertial properties. Hence, the augmented object model is again used. Consider the kinetic energy matrix of the object, in the object's CoM coordinate frame, denoted by $\mathcal{F}_{com}$, given by:

$$\Lambda_{O_{com}} = \begin{pmatrix} m\mathrm{I}_{3x3} & 0 \\ 0 & I_{CoM} \end{pmatrix} \tag{5.4}$$

where m and I_{CoM} are the object's mass and inertia tensor and I is the unit matrix. This kinetic energy can be also expressed in the frame attached to the end-effector at a desired grasping pose, denoted by $\mathcal{F}_{gp}$ by using a transformation T, which transforms linear and angular velocities from $\mathcal{F}_{com}$ to $\mathcal{F}_{gp}$. Let r be the vector that connects $\mathcal{F}_{gp}$ to $\mathcal{F}_{com}$ and $\hat{r}$ the cross-product operator for r, then T is given by:

$$T = \begin{pmatrix} I & \hat{r} \\ 0 & I \end{pmatrix} \tag{5.5}$$

The kinetic energy matrix $\Lambda_{O_{gp}}$ expressed in $\mathcal{F}_{gp}$ is thus given by:

$$\Lambda_{O_{gp}} = T^T \Lambda_{O_{com}} T \tag{5.6}$$

$\Lambda_{O_{gp}}$ needs to be transformed to an appropriate operational space coordinate representation. This is achieved by using a matrix $E(\mathbf{x})$ which relates operational generalised velocities to linear and angular velocities notation. $E(\mathbf{x})$ is only dependent on the choice of the variables to represent position and orientation in $\mathcal{F}_{gp}$ e.g, Cartesian position and Euler angles Khatib, 1995. The kinetic energy matrix of the object, expressed in the end-effector frames and in operational coordinates representation is given by:

$$\Lambda_o(\mathbf{x}) = E^T(\mathbf{x})\Lambda_{O_{gp}}E^{-T}(\mathbf{x}) \tag{5.7}$$

As the kinetic energy matrices $\Lambda_o(\mathbf{x})$ and $\Lambda(\mathbf{x})$ are now represented in the same coordinate frame and operational coordinate representation, the kinetic energy of the whole system comes from the addition of the two matrices:

$$\Lambda_{tot}(\mathbf{x}) = \Lambda_{obj}(\mathbf{x}) + \Lambda(\mathbf{x}) \tag{5.8}$$

A more detailed explanation of the augmented model can be found by Khatib, 1987. Eq. (5.8) shows that the total kinetic energy matrix is simply the addition of the manipulator and the object energies expressed in $\mathcal{F}_{gp}$. As $\Lambda_{tot}(\mathbf{x})$ has the same form and meaning as for the manipulator, namely $\Lambda(\mathbf{x})$, its inverse always exists and it has the following form:

$$\Lambda_{tot}(\mathbf{x})^{-1} = \begin{pmatrix} \Lambda^{-1}_{u_{tot}}(\mathbf{x}) & \Lambda_{uw_{tot}}(\mathbf{x}) \\ \Lambda^{T}_{uw_{tot}}(\mathbf{x}) & \Lambda^{-1}_{w_{tot}}(\mathbf{x}) \end{pmatrix} \tag{5.9}$$

where $\Lambda_{u_{tot}}(\mathbf{x})$ represents the inertial properties of the augmented model associated with translation, $\Lambda_{w_{tot}}(\mathbf{x})$ is the inertial properties associated with rotation, and $\Lambda_{uw_{tot}}(\mathbf{x})$ represents a measure of coupling between angular and linear parts.

As mentioned in the introduction, the aim is to select a grasp that produces minimum impact force in case of a collision, as compared to other possible grasps. It is assumes that a set of possible grasps for the object have been generated by a grasp synthesizer, e.g. Kopicki et al., 2015 or several other recently proposed methods. Since the object's CoM is known, every synthesized grasp represents a pose transformation between the CoM frame and the robot's end-effector frame.

It has been shown that a manipulator during a collision is perceived according to its effective mass (Eq. (5.3)). In analogy, the effective mass of the total system is defined as

$$m_{eff,tot} = \frac{1}{\mathbf{v}^T\Lambda^{-1}_{\mathbf{u}_{tot}}(\mathbf{x})\mathbf{v}} \tag{5.10}$$

The total effective mass is dependent on the object's inertia tensor w.r.t $\mathcal{F}_{gp}$. This means that grasping from different poses results in different kinetic energy values of the corresponding augmented models. Hence, there may exist one grasp choice which could result in an impact force lower than that for all other possible grasps. In the next section it is shown that grasp poses have a significant effect on a safety measure of effective mass in the context of human-robot collaboration.

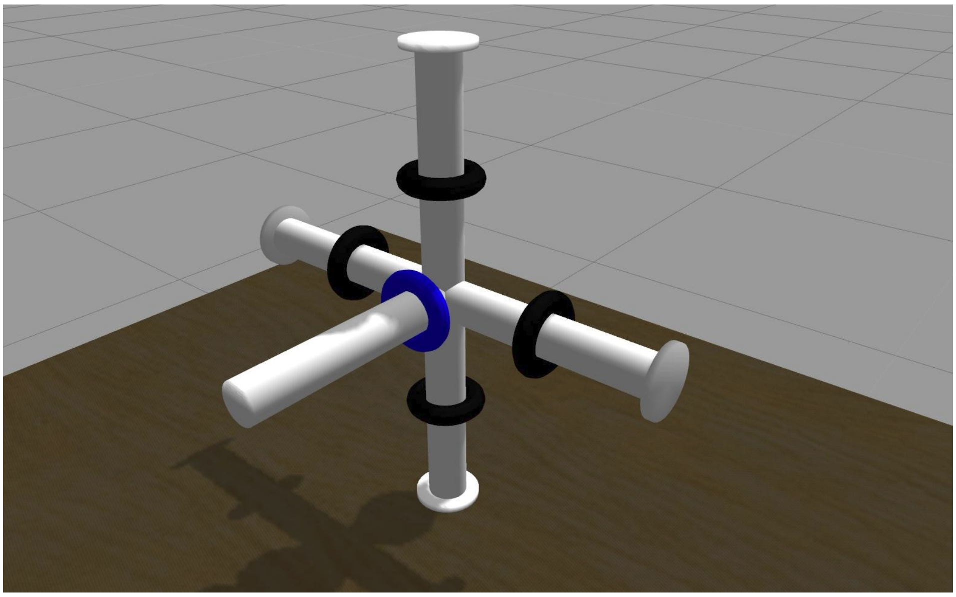


Figure 5.3: The tensor object. The caps of the cylinders have negligible mass. The black and blue weighted rings are able to move along the cylinders and latch when needed. By changing a ring's position, the inertia tensor of the object and the perceived inertia tensor at the contact point are changing. The ring on the x axis (handle) is coloured blue so that it can be distinguished easily.

5.3 Experimental results

This section presents the results of several experiments including two experiments in simulation environments and one experiment using a real robot. The experiments are designed such that the alteration of the robot's dynamics with each grasping point is illustrated. In particular, the pre-calculated effective mass along a given trajectory for each grasp, and the impact force that the grasp will produce in a collision scenario, are discussed in the following to show the effectiveness of our approach.

Again, the 7-DOF Baxter® robot by *Rethink Robotics* is used for both simulation and real robot experiments. Programming is done in Baxter Software Development Kit (SDK), the Python Kinematic Dynamic Library (PyKDL) and the Gazebo simulation environment which supports object and sensor generation and built-in physics engines.

The continuous end-effector trajectory is sampled with a sampling rate of $\Delta t = 0.04$ sec, to collect a total number of $N = \frac{t_f}{\Delta t}$ intermediate end-effector poses $\mathbf{x}_i$, $i = 1, ..., N$ in Cartesian space. The poses $\mathbf{x}_i$ are used to calculate the dynamic properties of the manipulator during the trajectory.

5.3.1 Simulation with tensor object

In the first simulation, a "tensor object" is used. Tensor objects are widely used in experimental psychology, to demonstrate how changes in an object's inertia tensor affect its perceived properties by a human who is manipulating it (Amazeen et al., 1996). A tensor object consists of five cylinders in the form of a 3D coordinate frame, as shown in Fig. 5.3. One of the cylinders is chosen as the handle for grasping.

Toroidal weights (shown as black and blue coloured rings in Fig. 5.3) are located on each cylinder and can slide along it. The total mass of the tensor object is 0.43 kg. By changing each weight's position along its respective cylinder, the object's inertia tensor w.r.t. the CoM can be modified. Subsequently, the perceived inertia tensor at $\mathcal{F}_{gp}$ changes.

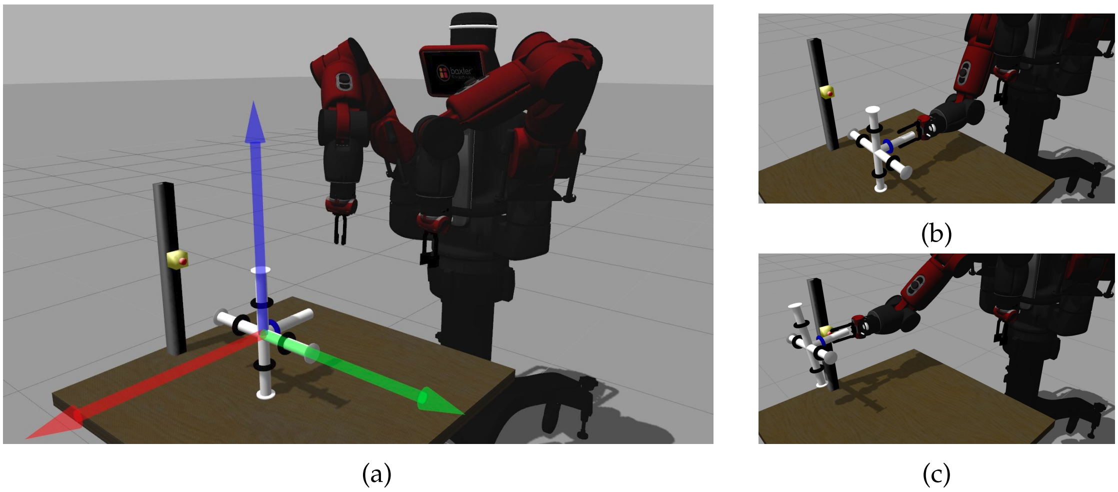
The tensor object for this experiment, shown in Fig. 5.3, is designed such that the robot pose expressed by $\mathcal{F}_{gp}$ relative to the $\mathcal{F}_{com}$ is kept invariant over different grasps, whereas the inertia tensor of the object can be changed. The velocity and type of collision are kept invariant over all experiments. This allows studying the effect of just one variable (inertia tensor of the object being manipulated) on the impact force during a post-grasp collision, while keeping all other conditions invariant throughout the experiment.

20 different grasps were generated by altering the position of the rings on the tensor object. For each grasp a collision scenario is conducted to measure the impact force that is delivered by the robot end-effector. The robot picks and moves the object to its right side according to the task trajectory described in the previous section. The initial and final 3D positions of the end-effector were set to $\mathbf{p}_0 = (1,\ 0,\ 0.03)$ and $\mathbf{p}_f = (1.1,\ -0.38,\ 0.16)$ respectively, where distances are measured in meters. For simplicity the orientation is unchanged during the trajectory, although possible changes in orientation would result in the same form of trajectory polynomial. The trajectory total duration is 2 seconds, leading to 50 trajectory samples. After a brief time, the end-effector collides with the force sensor on the vertical pillar (Fig. 5.4).

The proposed approach in the previous section is used to pre-calculate the effective mass of the augmented model corresponding to each grasp, and the results are shown in Fig. 5.5. The results show that the value of effective mass significantly varies between different grasps. This figure indicates the safest grasp with the minimum effective mass. It should be noted that the effective mass is a function of the manipulator's trajectory, which slightly changes when the robot reaches different grasps on a static objects. The effect of this trajectory change in the total effective mass was negligible compared to the change attributed to the object dynamics.

To further validate the proposed hypothesis, that grasps with different effective mass produce significantly different impact forces, a collision was simulated while recording the impact forces for various different grasps.

Fig. 5.6 plots the impact forces resulting from three different grasps: the grasp with maximum effective mass in Fig. 5.5; the grasp with minimum effective mass; and a grasp with moderate effective mass. The robot is commanded to cease its effort 2 seconds after the collision, so that the peak impact force and the steady state contact force values can be easily evaluated. Because the collision happens roughly 1.5 seconds after the start of the trajectory, the total measurement time was roughly 3.5 seconds as shown in Fig. 5.6 . The impact forces are clearly visible in Fig. 5.6 as the large peak in the force signals at the collision time. It can be seen that the values of these peak forces are in accordance with the computed effective mass, i.e. the greater the effective mass, the greater the peak value of the impact force. These measurements validate the presented hypothesis (Fig. 5.5) and suggest that the proposed approach can make a useful contribution to improving robot safety.



(a)

(b)

(c)

Figure 5.4: The Baxter robot and tensor object simulation configuration. An object with variable inertia tensor is used to demonstrate the effect of different grasping positions on the severity of collisions with the environment. The executed experiments consist of the robot approaching the object, and grasping it as shown. The robot then lifts the object and transports it along the task trajectory. At some point along this trajectory, a rigid pillar equipped with a virtual force sensor is intentionally introduced, shown here as the red button. The robot's end-effector then collides with the force sensor. The purpose is to measure the exerted force on the sensor along the trajectory direction. (a) The simulation configuration. (b) The starting point of the end-effector trajectory. The grasping point identifies with the starting point of the end-effector trajectory. (c) The collision point of the end-effector trajectory.

5.3.2 Simulated grasping and moving of book object

The method is then applied to a more realistic example in which several different grasp locations on an object are generated and, consequently, the robot's configurations and post-grasp trajectories become altered for each new grasp (in contrast to the previous experiment, in which the specially designed tensor object enabled identical robot motions over all experimental runs). The handled object is a book-shaped object with dimensions 22x15x1.5 cm and mass of 0.34 kg (to align with real robot experiments with a real book, described in the next section). This book object is initially positioned in front of the robot on a table. The robot performs three different grasps, on three different parts of the book, as shown in Fig. 5.7. The task setup is similar to the previous experiment, i.e. after grasping the book, the robot collides forcefully with a rigid pillar containing a virtual force sensor. The grasp points are at $-0.1m$, $0m$ and $0.1m$, where these are grasp positions along the spine of the book, which is aligned with the y axis (shown in Fig.5.7a as the green arrow). The pre-calculated effective masses along the post-grasp trajectory, for each of the three grasps, are shown in Fig. 5.8. For each grasp, the force sensor is used to model the resulting impact forces, which are plotted in Fig. 5.9. Again, higher impact forces result from grasps which yield higher effective masses along the post-grasp trajectory, suggesting that computation of effective mass is a useful predictor of collision safety. Furthermore, it is clear that significantly different effective mass, and significantly different impact forces, result from different choices of grasp.

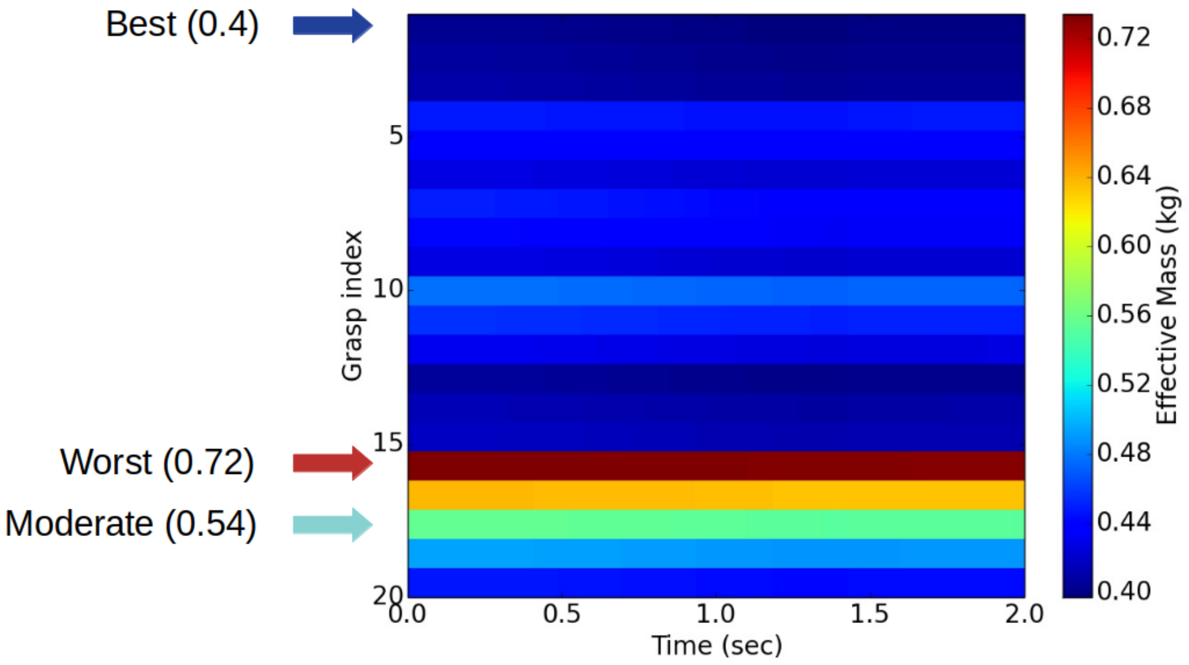


Figure 5.5: Map of the effective mass along the trajectory for all grasps of the tensor object. The vertical axis denotes each of the 20 example grasps, and the horizontal axis denotes time, throughout the duration of the post-grasp trajectory. Colour denotes the magnitude of the effective mass for each grasp at each time step. Each horizontal segment, starting from left to right, plots the magnitude of the effective mass that each grasp produces over the entire time duration of the task trajectory. It can be seen that different grasps can indeed generate different effective masses along the duration of the task trajectory, with significant variations in the effective mass magnitude. The result of these variations is that the methodology can discriminate safer from less safe grasps.

5.3.3 Experiment using a real robot

To further test the method, the simulated book experiment is repeated using a real Baxter robot. The experimental set-up is consistent with that of the simulation experiment, i.e. identical grasp positions and post-grasp trajectories. The real book, shown in Fig. 5.1, has exactly the same mass and dimensions as the virtual book object in the simulation experiment, so that the effective mass computation, Fig. 5.8, is identically valid for our experiment with the real Baxter robot and the real book. To avoid damaging expensive equipment, the robot is made to collide with a plastic bottle full of water, instead of a completely rigid pillar as used in the simulation experiment. During the post-grasp trajectory, the robot collides with the bottle for a brief period of time, knocking it over.

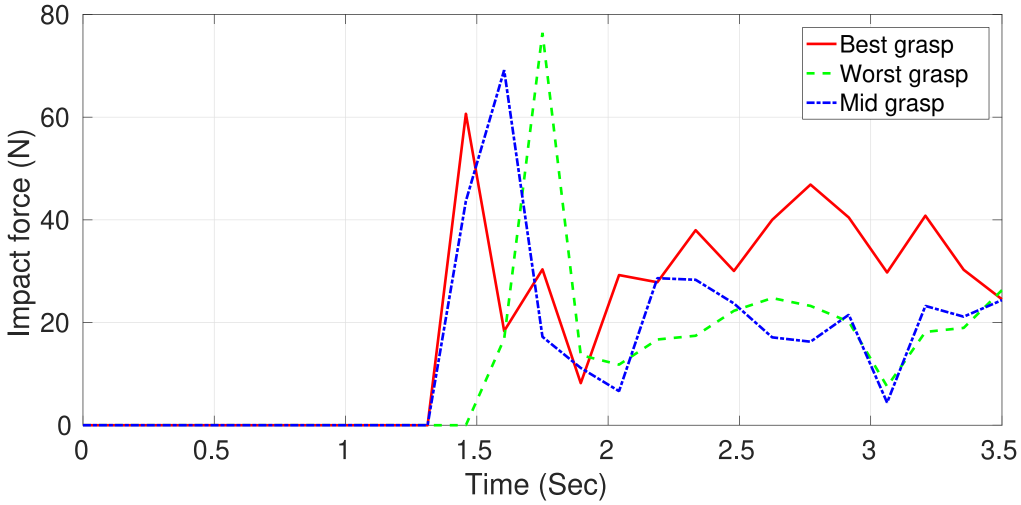
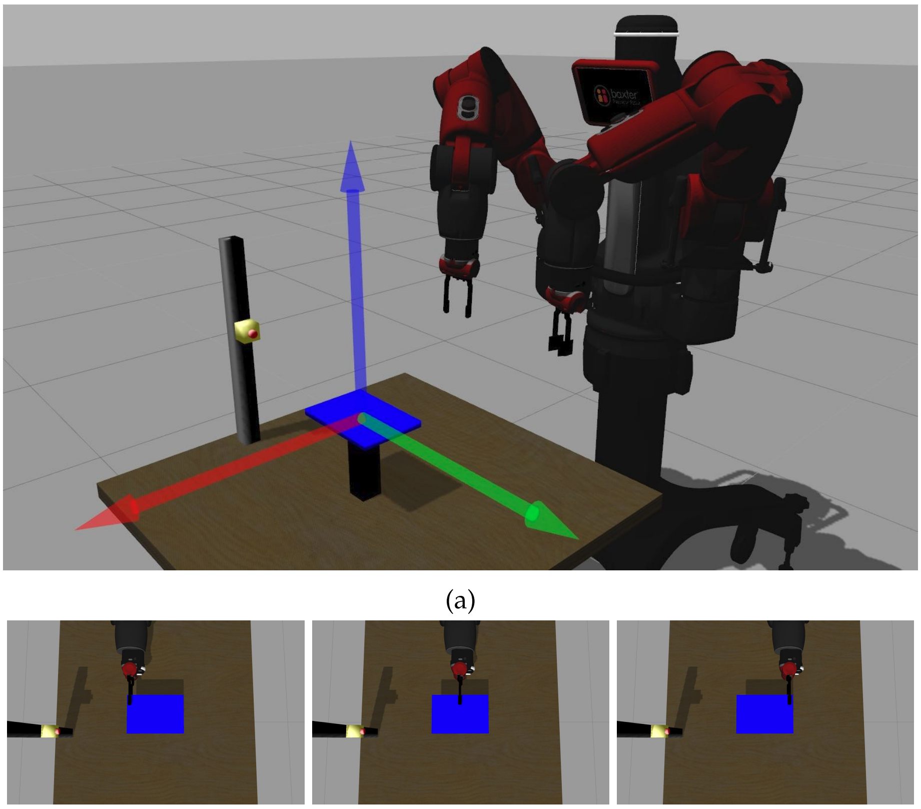


Figure 5.6: Impact force profiles for three different grasps of the tensor object. Red colour corresponds to the minimum effective mass grasp, green to the maximum effective mass grasp, and blue to a grasp with an intermediary value, sampled from the 20 grasps shown in Fig. 5.5. The impact forces are visible as the large initial force peak, and it is clear that the magnitudes of these impact forces vary between the grasps. Since the task trajectory is the same for all grasps and the collision type and geometry remain the same, these variations in the impact forces depend only on the choice of grasp. The relatively high magnitudes are attributed to the fact that the robot collides with a simulated, rigid pillar that has a virtual force sensor. As the robot keeps pushing, the recoil generates oscillations (especially in the spring-loaded series-elastic-actuators of the Baxter robot) which eventually converge to a steady state contact force.

In the simulation experiments, a virtual force sensor was created to measure impact forces. In future work, a high-precision, force-torque sensor can be used for accurately measuring these forces in experiments with real robots. Since such a sensor was unavailable at the time of writing, the Baxter robot's (rather noisy) force estimation system was used, based on a model of its series-elastic actuators combined with joint rotation sensors, to approximately measure the forces experienced by its end-effector. The results are shown in Fig. 5.10.

The time of collision can be clearly seen in Fig. 5.10 as a large peak in the contact force, shortly before 0.5 seconds after motion begins. From Fig. 5.10 it can be seen that the maximum force during impact is different for each grasp, with around 30% difference between safest grasp and least safe grasp. This is consistent with the simulation experiment, where the minimum impact force (safest grasp) is around 100N, while the maximum impact force (least safe grasp) is around 130N.

Note that, in the simulation experiment, the overall magnitude of contact forces was higher, because the robot was colliding with a completely rigid and static obstacle. In contrast, in the real robot experiment, the robot collides with a light-weight, deformable obstacle (full water bottle), which topples over after impact. Nevertheless, the differences in impact force between the safest and least safe grasps were similar in both real and simulation experiments, with approximately 30% difference



(a)

(b) (c) (d)

Figure 5.7: Simulated grasping and moving of a book. The robot was given three grasping points on a book with dimensions 22x15x1.5 cm and mass of 0.34 kg. After grasping the book, the robot hits the force sensor shown on the black pillar, and the impact forces are measured. (a) Experimental setup. The book coordinate frame is visible. (b) First grasp. (c) Second grasp. (d) Third grasp.

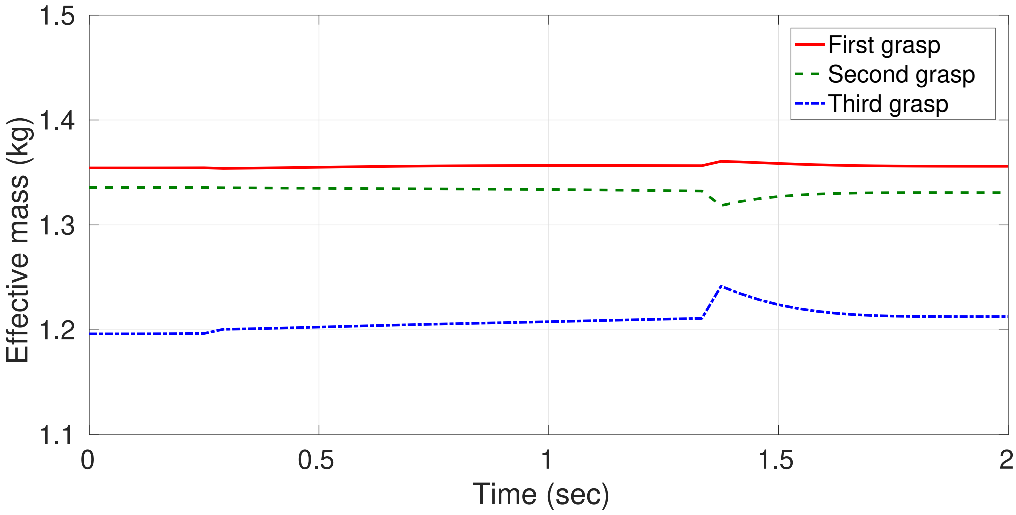


Figure 5.8: Effective mass values, computed along a post-grasp trajectory, for three different grasping points on the book object. It can be seen that different grasps result in different effective masses. Computing effective mass along the desired post-grasp trajectory, can be used to predict the safety of each grasp w.r.t. collisions.

in both cases. In both real and simulation experiments, the grasp with minimum effective mass results in minimum impact force. In both experiments, the most safe grasp was located on the right side of the book (Fig. 5.1d) and the least safe grasp was on the left side (Fig. 5.1b).

5.4 Discussion

The results acquired from all experiments support the proposed methodology. As demonstrated, selection of an optimal grasp can reduce the impact in case of a collision. Grasps that minimise the computed effective mass along a desired post-grasp trajectory, are an effective predictor of grasps that maximise safety with respect to post-grasp collisions. This Chapter has proposed a way of pre-calculating this optimality, by using the effective mass of the augmented dynamics, integrated over the desired post-grasp trajectory. As seen in Fig. 5.5 and Fig. 5.8, the difference between effective mass values can vary significantly between different grasp choices. Furthermore, effective mass differences, between different grasps, result in significantly different impact forces in post-grasp collisions. In all cases, the experimental results consistently support that selection of grasp location can make a robot significantly safer, and pre-calculations of the effective mass, along a desired post-grasp trajectory, are a useful way of identifying the grasp choices that minimise the severity of post-grasp collisions.

The experiments were designed to be simple and able to isolate and measure the effect of each grasp at the instance of collision. As the interest is primarily in minimising the impact force, there was not any optimisation, or further investigation, of

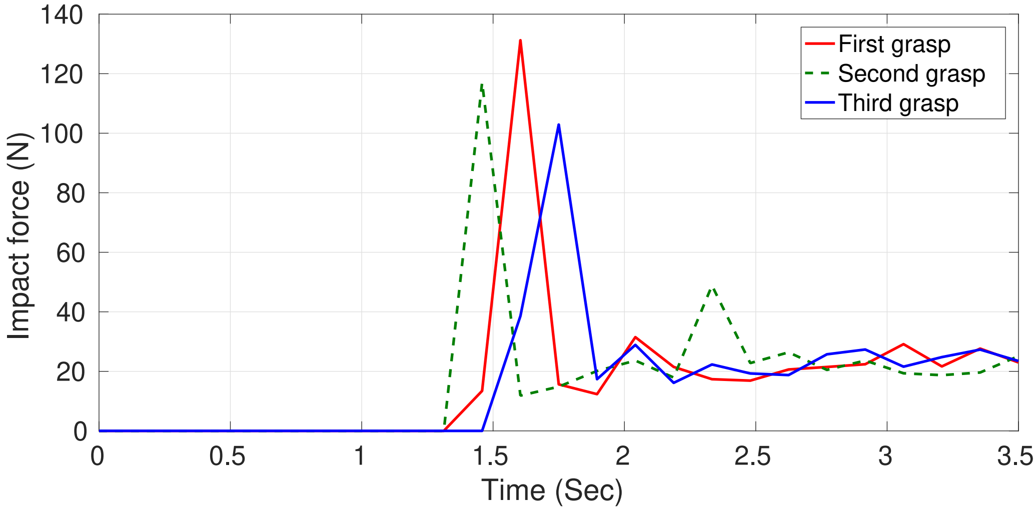
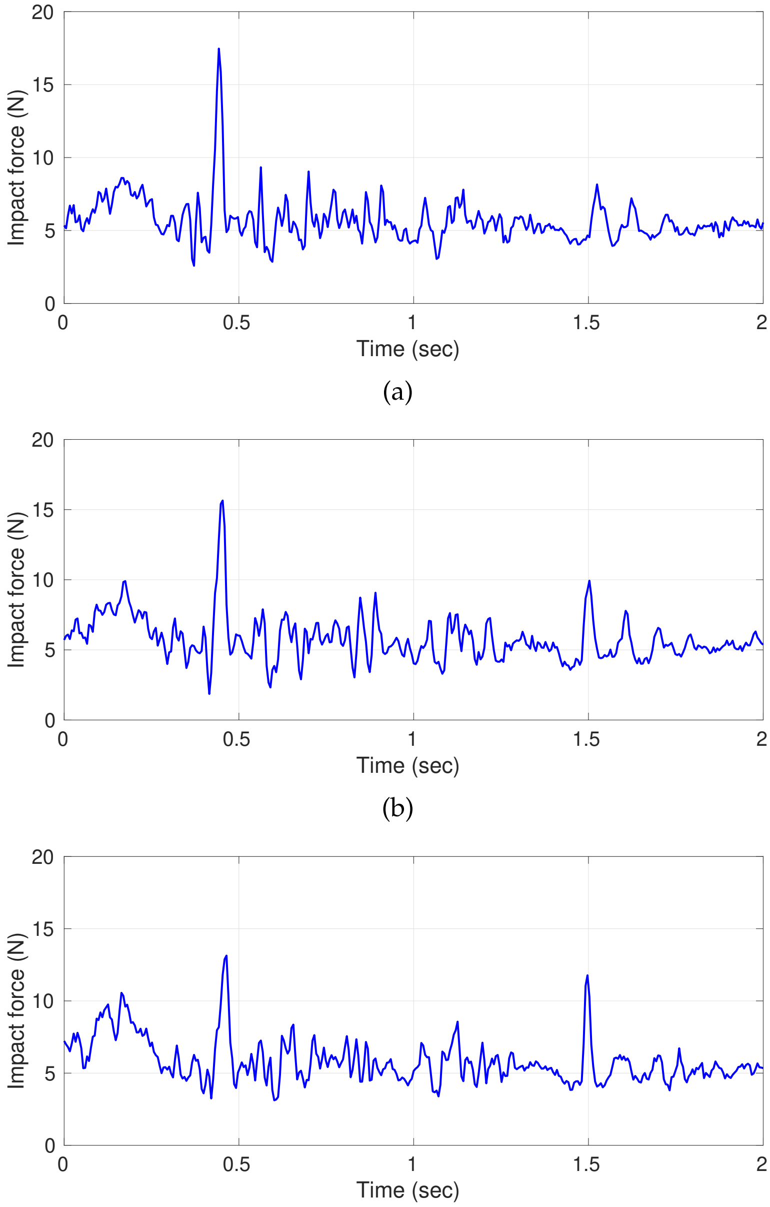


Figure 5.9: Simulated impact force evolutions, between the robot's end-effector and a virtual force sensor, for three different choices of grasp. Similar to the tensor object experiment, the three different grasps on the book object yield significantly different impact force values during the same collision profile. The forces for the first, second and third grasp are shown in red, green and blue respectively. The robot continues pushing after colliding, and the inherent elasticity of the Baxter actuators leads to oscillations which then decay to a steady state contact force. Note that the different timings of the initial impact in each case, are because the robot's wrist begins its motion at three different positions, corresponding to three different grasps on the spine of the book.

post-impact phenomena such as manipulator stability after impact or steady-state response of the contact force. The connection of these phenomena to post-grasp manipulation and the grasped object's inertial properties can serve as an interesting topic for future research. Other ways of potentially enhancing the proposed method in the future include further minimisation of the force by redundancy control, or evaluating how changes in the object's inertia tensor during the task execution, due to e.g. grasp slippage, would perform.

It should also be noted that the dynamics of the robot may not always be available in analytical form, and may need to be identified. The same can be said for the inertial properties of grasped objects, where only approximations can be given, with some margin of error, in many real-world applications. Nevertheless, the proposed methodology is suitable for providing safe grasps, even when using such approximations.



(a)

(b)

(c)

Figure 5.10: Impact force evolutions for three different grasps, performed with a real robot colliding with an obstacle (water-filled bottle). The point of collision is clearly visible on the plots, in the form of a sharp peak in the force measurements. Once again, different grasps lead to different impact forces (observed in the first large peak in contact force, just before 0.5 seconds), consistent with the simulation experiments. The magnitude of the peak collision forces varies from 13N (safest grasp) to 17N (most dangerous grasp), i.e. roughly 30% difference in safety for even this very simple case, with seemingly modest changes in the choice of grasp pose. Note that the contact force fluctuations, after the collision, are most likely attributable to gravity torque compensation behaviour of the real Baxter robot, combined with minor inertial effects (note that the bottle obstacle has toppled over at this point, so no obstacle exists to account for this post-impact fluctuation behaviour). (a) First grasp. (b) Second grasp. (c) Third grasp.

Chapter 6

Generating Manipulation Criteria From the Inertial Parameters

6.1 Introduction

The lessons learnt from the previous chapters suggest that a robot can use the combined arm-object dynamics to generate manipulation criteria for a given task. Since the task to be executed has common mathematical formulation for all manipulation metrics, it comes as following that the aforementioned criteria can be used on top of each other along the task. However, since the criteria are calculated independently from each other, a serious question is posed: is there a possibility that task-relevant objectives conflict? Is there a chance that e.g. a grasp with lower torque leads to a grasp with potentially higher impact force?

The answer to this question is not straightforward to give, and depends on the specific manipulation task and application. The main theme of this chapter is to demonstrate how the values of task-oriented criteria vary under a different selection of grasp poses, even though they are a function of the object's inertial parameters.

6.2 Trajectory definition and post-grasp objectives

In this Chapter, the notation of Section 5.2 is followed for the definition of a task in the object's CoM space, robot's end-effector space, as well as the relative transformations between the object's CoM and the grasping points. Three different post-grasp criteria are considered. On top with the already presented manipulation effort and impact force minimising criteria, the maximisation of the robot's manipulability is considered. It is shown that these objective values are functions of the inertial parameters of the object, the selected grasp pose and a post-grasp trajectory.

6.2.1 Task oriented kinematic velocity manipulability (TOV)

Ghalamzan E et al., 2016 introduced a Task Oriented Kinematic Velocity Manipulability cost function (TOV) to address the problem of jointly planning both grasps and subsequent manipulative actions. This cost function was also used in a mixed initiative, shared control for master-slave grasping and manipulation (Ghalamzan et al., 2017). It was shown that maximising TOV results in significantly reduced joint velocities.

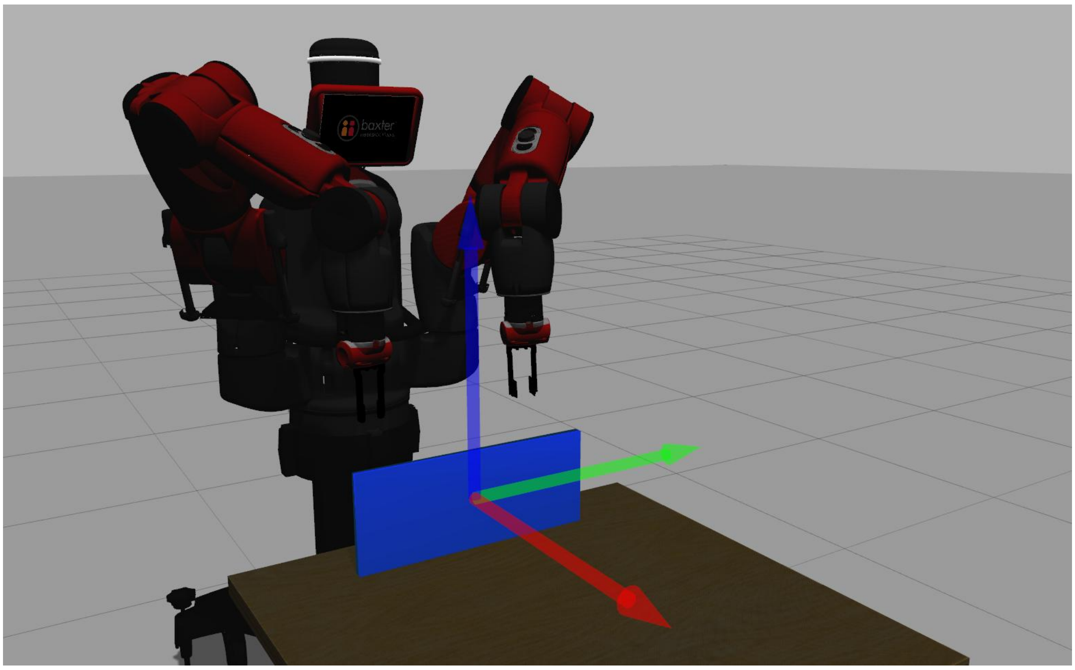


Figure 6.1: Simulation setup with a Baxter robot in Gazebo simulator. The robot is tasked with grasping and then moving the object. The robot manipulates a cuboid object with dimensions $0.5x0.15x0.2m^3$ and mass of $0.4kg$. The coordinate axes of the object's centroid are shown, where red, green and blue correspond to x,y and z axes, respectively. The inertia tensor of the object is known in advance. 10 different grasps are generated for evaluation. Three different Pick-and-Place tasks are provided for the robot to execute, and for each task and grasp, the effective mass, the joint effort and the manipulability along the task trajectory are pre-calculated. The aim is to investigate the performance of each grasping point according to the calculated metrics.

Let $\boldsymbol{\theta} \in \mathbb{R}^n$ be the joint vector of the considered manipulator arm, and

$$\boldsymbol{u} = \left[\begin{pmatrix} v_g \\ \boldsymbol{\omega}_g \end{pmatrix} \right] = J(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \tag{6.1}$$

be the geometric Jacobian relating joint velocities to the end-effector linear/angular velocities $\boldsymbol{u} = (v_g, \boldsymbol{\omega}_g) \in R^6$ in the end-effector frame $\mathcal{F}_g$ (for ease of notation, the superscript g is dropped for the quantities in Eq. (6.1). The kinematic velocity manipulability ellipsoid is defined by

$$\boldsymbol{u}^T(J(\boldsymbol{\theta})J(\boldsymbol{\theta})^T)^{-1}\boldsymbol{u} = 1 \tag{6.2}$$

and it represents the capability of the robot manipulator in generating task space velocities for a given norm of joint velocities, which is a metric of the robot's dexterity. In this Chapter the interest is in in maximising (in an integral sense) a particular *task-oriented* manipulability measure derived from Eq. (6.2): the radius of the manipulability ellipsoid along the tangent vector to the desired path in task space. This is meant to ease as much as possible the execution of the desired trajectory by the robot arm, with the smallest norm of the joint velocities.

Consider $\boldsymbol{\theta}(t)$ being the trajectory in joint space associated to the end-effector trajectory, and generated by the robot inverse kinematics, where $\boldsymbol{u}(t)$ is the corresponding linear/angular end-effector velocity at each time. $\mathbf{u}(t)$ is decomposed as $\mathbf{u}(t) = a(t)\mathbf{u}(\bar{t})$, with $a(t)$ representing the norm of $\mathbf{u}(t)$ and $\mathbf{u}(\bar{t})$ its (unit-norm) direction. From Eq. 6.2 it follows that, along the planned path,

$$a^2(t)\bar{\boldsymbol{u}}^T(t)(J(\boldsymbol{\theta}(t))J^T(\boldsymbol{\theta}(t)))^{-1}\boldsymbol{u}(\bar{t}) = 1. \tag{6.3}$$

It is easy to verify that the quantity $a(t)$ solution of Eq. (6.3) represents the length of the ellipsoid radius along the direction $\boldsymbol{u}(\bar{t})$. The aim is to maximise the quantity $a(t)$ along the whole path as defined in the following integral cost function:

$$H_{TOV}({}^r\boldsymbol{x}_g) = \int \zeta_a \alpha^2({}^r\boldsymbol{x}_g, s)ds = \int \zeta_a \frac{1}{\bar{\boldsymbol{u}}^T(J(\boldsymbol{\theta})J^{-T}(\boldsymbol{\theta}))\bar{\boldsymbol{u}}} \tag{6.4}$$

where $0 \leq s \leq 1$, s is a parametrisation of the path, $s = 0$ indicates $t = 0$, $s = 1$ shows $t = t_f$ and t_f is the time to completion. Let $\bar{\boldsymbol{u}} = \bar{\boldsymbol{u}}({}^r x_g, s)$, $\boldsymbol{\theta} = \boldsymbol{\theta}({}^r x_g, s)$. Ghalamzan et al., 2017, called H_{TOV} Task-Oriented Velocity Manipulability (TOV).

6.2.2 Manipulator dynamics under load

In this section, it is again assumed that the dynamic model of the robot is known, as well as the corresponding governing equation of motion of the manipulator in the joint space. Here, the interest lies in computing the total energy consumption of the robot when executing the desired post-grasp trajectory. Hence, the augmented dynamic model as presented in Chapter 5 is again used. Following the notation of Chapter 5, Eq. (4.19) is used:

$$M_{tot}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + C_{tot}(\dot{\boldsymbol{\theta}}, \boldsymbol{\theta}) + N_{tot}(\boldsymbol{\theta}) = \boldsymbol{\tau}_{\boldsymbol{tot}}(t) \tag{6.5}$$

Manipulation energy consumption

M_{tot} is used to compute the corresponding torque of augmented model of object and manipulator. The energy consumption of the robot to manipulate the object along path ζ_o is defined as:

$$H_{TME}({}^r\boldsymbol{x}_g) = \int_{\zeta_o} \boldsymbol{\tau}_{\vec{\boldsymbol{tot}}}{}^2 ds \tag{6.6}$$

Effective mass definition

While one can compute the force at every point of interest of the manipulator by writing the corresponding operational space equation, one can project the kinetic energy matrix M_{tot} to the end-effector frame as per the analysis of Chapter 6, and extract the end-effector total kinetic energy matrix Λ_{tot}. This matrix is related to the impact force during a collision with a robot's end-effector.

It has been shown in the previous Chapter that during collision, a manipulator is perceived according to its effective mass (Eq. 6.7), denoted by m_{eff}. As in the

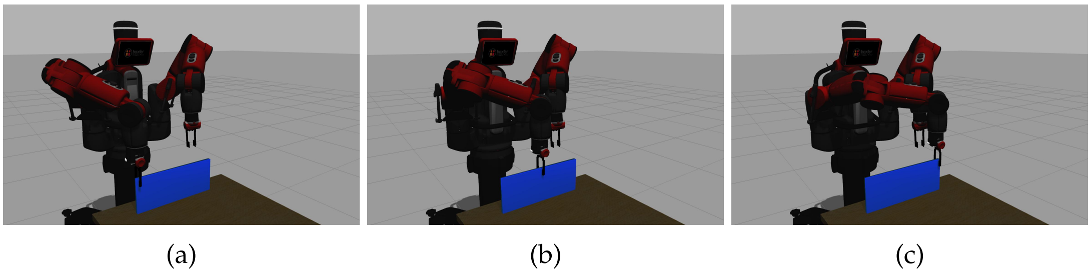


Figure 6.2: In the first task, the robot needs to pick up the object (blue cuboid), which is located on the table, move it 20 cm in line with the negative y-axis and 10 cm in line with positive x-axis and place it on the table. The x and y axes are shown with red and green arrows in Fig. 6.1. All 10 grasping poses are equally distributed on the top edge of the cuboid. Three example grasps on the object are shown where (a), (b) and (c) show the first, fifth and tenth grasping pose.

previous Chapter, the effective mass of the total system is defined as

$$m_e({}^r\boldsymbol{x}_g, s) = \frac{1}{\bar{\boldsymbol{u}}^T \Lambda^{-1}_{\boldsymbol{u},tot}(\boldsymbol{x})\bar{\boldsymbol{u}}} \tag{6.7}$$

and the minimisation criterion along the task trajectory is:

$$H_{TEM}({}^r\boldsymbol{x}_g) = \int_{\zeta_o} m_e ds \tag{6.8}$$

Ideally, a high value of H_{TOV} (or low value of $\frac{1}{H_{TOV}}$) and low values of H_{TEM} and H_{TME} are desired. Although a native approach to minimise all can be achieved by an affine combination of all objectives, this chapter shows that this approach is not sophisticated and the solution must be obtained through a multi objective optimisation approach.

6.3 Experimental results

A series of experiments is conducted with a Baxter robot manipulating an object with a given task using the Gazebo simulator. The set-up is shown in Fig. 6.1. The task is to pick a cuboid object and place it at different poses. The object has dimensions $0.5 \times 0.15 \times 0.2 m^3$ and uniform mass distribution with a mass value of $0.4 kg$. 10 different grasping poses are considered on the object surface. The contact locations of the grasping poses are uniformly distributed on the top edge of the cuboid. Three of the generated grasp poses are shown in Fig. 6.2. The first grasp is located at $-0.22 cm$ and the last one is at $0.22 cm$ along the y-axis. The robot approaches the contact points of each grasping pose on the top edge of the object from predefined approach points located $15 cm$ above each grasping pose.

In total, the robot performs three Pick-and-Place tasks: the robot lifts up the object $10 cm$ from its initial position along the z axis and:

1. translates it in a combined motion $-20 cm$ along the y axis and $10 cm$ along the x axis and finally puts it down $-10 cm$ along the z axis;

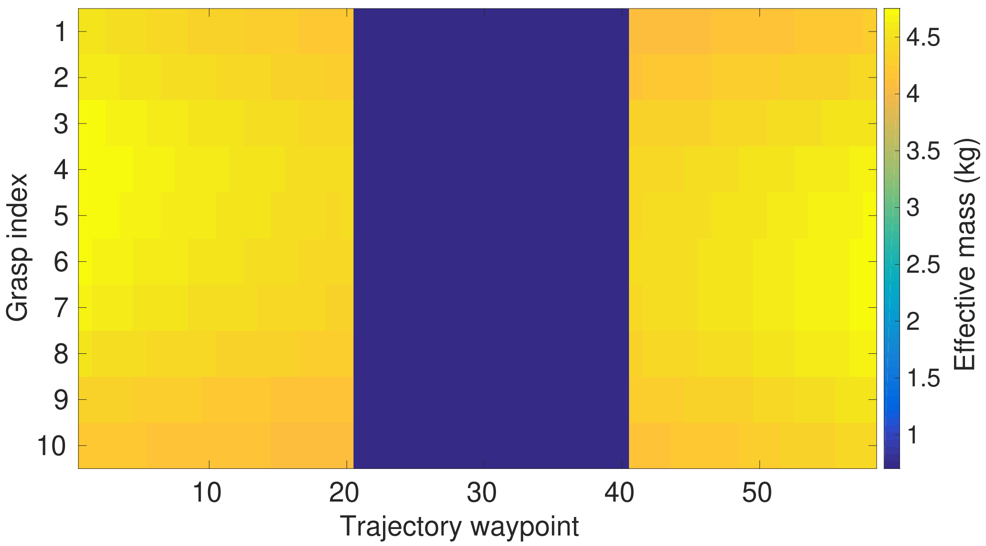


Figure 6.3: Heat map of the computed effective mass for the third task. The horizontal axis represents the waypoints along the task trajectory and the vertical axis shows the grasp poses considered on the top edge of the object. This figure shows that the metric value of effective mass correlates with the waypoint of the pick-and-place trajectory and the selected grasp pose.

2. translates $-0.35cm$ along the y axis and finally puts it down on the table;

3. translates $-35cm$ along the y axis and $-10cm$ along the x axis and finally places it on the table.

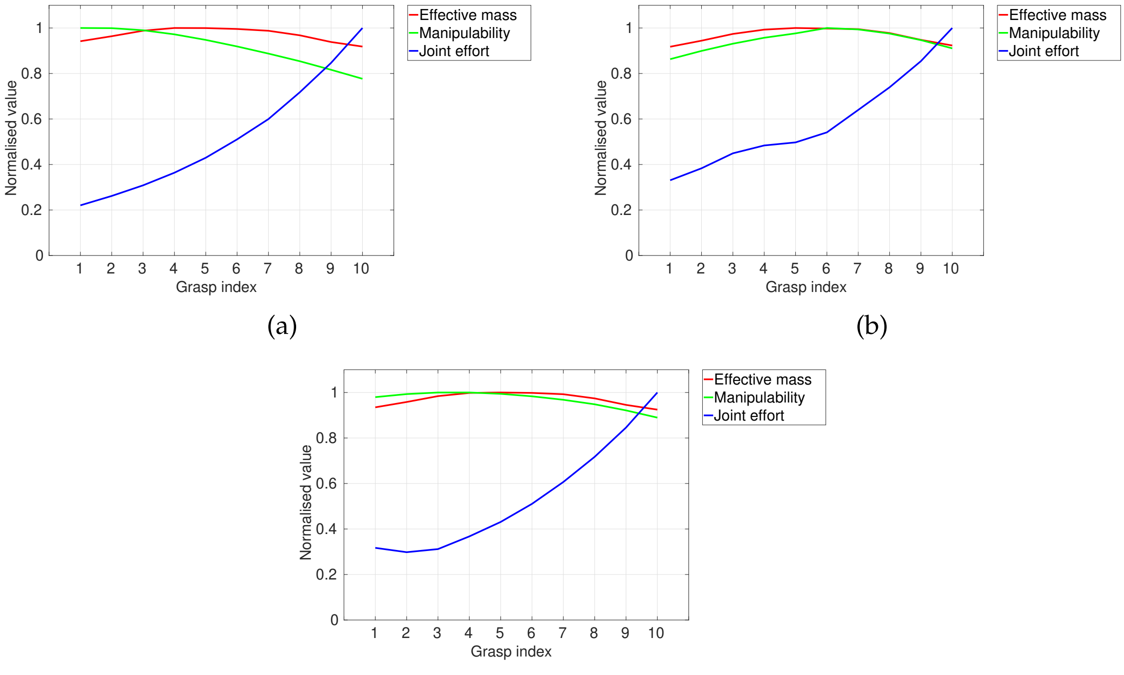
By using the Baxter PyKDL library, the Jacobian and the dynamic model for each point of a trajectory are computed, allowing for the calculation of the metrics as per Eq. 6.4, 6.6, 6.8. An example of the effective mass for every initial grasping pose versus sample points of the third task trajectory is presented in Fig. 6.3.

The integrals presented in Eq. 6.3, 6.6 and 6.8 are computed for every task. For better visualisation, the values of $\frac{1}{H_{\text{TOV}}}$, H_{TEM} and H_{TME} are normalised against their corresponding maximum; that is,

$$\mathrm{H}_{\text{TOV}} = \frac{H_{\text{TOV}}}{\max\left(H_{\text{TOV}}\right)},$$
$$\mathrm{H}_{\text{TEM}} = \frac{H_{\text{TEM}}}{\max\left(H_{\text{TEM}}\right)},$$
$$\mathrm{H}_{\text{TME}} = \frac{H_{\text{TME}}}{\max\left(H_{\text{TME}}\right)}.$$

These normalised metrics are shown in Fig. 6.4a, 6.4b and 6.4c for the first, second and third task.

A characteristic example is the first task (Fig. 6.4a), where it is clear that the grasp No. 1 is the optimal yielding minimum effective mass, minimum joint effort and maximum manipulability. Furthermore, Fig. 6.4a shows the manipulability, effort and effective mass significantly changing with the choice of grasp poses. This enables



(a)

(b)

(c)

Figure 6.4: The final scalar metric values (namely TOV shown in green line, joint effort shown in blue line and effective mass shown in red line) for (a) task 1 (b) task 2, and (c) task 3. The L2 norm of a metric along post-grasp trajectory yields a scalar value for each grasp pose. These values represent the manipulation quality of the grasp and are directly related to the task to be executed. As a result, the robot can choose a grasp that has low effective mass, low effort and higher manipulability. For instance, grasp number 1 in the first task, top figure, has maximum manipulability and minimum effective mass and effort.

the use of the presented methodology in choosing the grasp that is safe, yields the least effort and provides high manipulability for executing the task.

In contrast, the results yielded for the second task (Fig. 6.4b) show that the objectives do not agree on the optimal grasping pose, i.e. while the effective mass and joint effort are implying that grasp number 1 is optimal, TOV manipulability suggests that grasp number 6 is optimal. Likewise, the indices obtained for the task number 3 (Fig. 6.4c) show that they conflict, i.e. grasping pose number 2 yields minimum joint effort, whereas grasping pose number 1 is the best in terms of both TOV manipulability and effective mass.

6.4 Discussion

The results presented in this Chapter illustrate that the grasping pose selection for predefined manipulative actions is a complex multi-objective optimisation problem, where the manipulability metric needs to be maximised, and the effective mass and joint effort metrics need to be minimised. Although one may consider an affine combination of these objectives for grasp selection, a more clever approach of multi-objective optimisation for grasp selection would be necessary in any case-study to be solved. The optimisation problem is is tightly related to the specific user-case and the performance requirements and limits of the robot during the executed task. The definition of the cost function, as well as the optimisation algorithm is also case-dependent. In the case of highly non-linear or multi-modal problems, metaheuristic optimisation algorithms such as Genetic Algorithms could perform well. When solving a multi-objective optimisation problem the optimal solution, or Pareto solution, is the solution where further optimisation of an objective leads to degradation of any other objective. The main goal of the Chapter is to showcase how different solutions (grasps) would affect the performance of each objective individually. To examine this, the presented setup required only a small number of grasps, which is sufficient for "brute-force" examination of each grasp individually. As a result, applying an optimisation technique was left out of the scope of the Chapter. A natural next step for future work is to use the criteria defined in Chapters 4,5,6 along with a grasp synthesis method capable of extracting a large number of grasps and an optimisation method, fine-tuned in a realistic user-case.

Chapter 7

A Different Case Study: Using a Captured Object's Inertial Parameters in Space Robotics

7.1 Introduction

All the previous Chapters demonstrated various fundamental elements of combining an object's inertial parameters with the robot dynamics to generate post-grasp manipulation criteria. The presented methodology and results demonstrated the importance of using the augmented dynamics, and the variety of post-grasp criteria to be generated. The methodology was mathematically simple and fundamental, and for that reason it can be extended for use in different robot environments.

In this Chapter a more specialised case study is presented. The application examined is a robotic spacecraft in orbit that manipulates a captured object with known inertial parameters. Contrary to the previous Chapters where the inertial parameters were exploited for task-informed grasp selection, this Chapter focuses on utilising the inertial parameters for generating motion trajectories to execute the task *after* having it grasped. The aims of this chapter are to demonstrate that the exploitation of the inertial parameters can be additionally applied in the post-grasp-selection process, as well as to provide a more targeted-use case scenario than the previous Chapters as an example. What further connects it to the theme of the previous Chapters is the usage of the combined object-robot dynamics to minimise a newly-defined criterion. The case of an in-orbit space robot is examined. The usage of the inertial parameters of a captured space object for orbital operations has also not been studied extensively, and such a study fits the main theme of this thesis. While they are not used here, the criteria that were developed in the previous sections can help make the manipulation more power-efficient, and this is crucial in this type of environment.

Modern space robot applications are more and more related to operations where a robot needs to handle objects while on-orbit. Such operations include satellite servicing, where the robot must handle tools and satellite parts, space debris removal, where the robot needs to carefully grasp debris pieces, and asteroid interaction missions. Examples of missions where a robotic spacecraft would need to interact with objects are Reed et al., 2016, Castronuovo, 2011 and Mazanek et al., 2015.

A common characteristic of such missions is the need to plan for joint or end-effector trajectories to be followed by the space manipulator in order to complete the given task. The trajectory generation problem for a robotic spacecraft is a complex problem, due to the coupling between the movement of the manipulator and the movement of the base, as well as the dependence of the spacecraft's kinematic equations from the mass distribution of the whole system. There is a large amount of works providing solutions to the trajectory generation problem for a space robot using various kinemo-dynamic models. Some prominent approaches are briefly mentioned. In the work by Umetani et al., 1989, the authors introduced the concept of *Generalised Jacobian*, allowing for a mapping between the arm joint values and end-effector coordinates that incorporates the reaction effects between the arm and base movements. They demonstrated a resolved motion control scheme that uses the Generalised Jacobian. Nakamura et al., 1991 demonstrated the non-holonomic nature of a 6-DOF space robot, and used a bi-directional method to ensure motion planning away from the system's null-space. Papadopoulos et al., 2005 used polynomial functions for path planning on a free-flying space robot, subject to non-holonomic constraints. They were able to plan for both a manipulator trajectory and a spacecraft base attitude change, by only actuating the arm joints. Finally, Dubowsky et al., 1991 introduced the *Enhanced Disturbance Map*, a map of the robot's joint space that enables finding joint trajectories by following the direction on the map that minimises the base disturbance.

The problem becomes increasingly difficult when the robot is holding a captured object, as the addition of a payload at the end-effector alters the spacecraft kinemo-dynamics. The estimation of an object's inertial parameters in orbital environments is different than the methods presented in Chapter 2, due to the alterations in the motion equations. As a result, it was deemed out of the scope of this thesis. Nevertheless, there exist a number of works that provide methods to estimate the inertial parameters of the captured object (Chu et al., 2017; Murotsu et al., 1994; Pesce et al., 2017). As it is the case with terrestrial robots, these inertial parameters can then be incorporated into the spacecraft dynamics to yield a new dynamic model, and a trajectory can be planned using these combined dynamics.

It is clear that planning a trajectory for a space robot is not a trivial task due to the mathematical complexity of the problem with all the existing constraints. Moreover, a trajectory needs to be planned in a way that many more than one objectives have to met, increasing the complexity even more. For example a trajectory needs to avoid singular configurations of the manipulator, while at the same time restrict the coupled movement of the base and minimise consumed power. To further echo the comments of the previous Chapter, it is obvious that optimisation methods need to be used for satisfaction of the criteria. However, a multi-objective solution under all these constraints may be difficult and time-demanding to find with model-based optimisation techniques. As a result, multi-objective meta-heuristic optimisation methods have been used in literature to find a best solution in finite time. Some methods are presented here. Xu et al., 2008 used Particle Swarm Optimisation (PSO) in order to plan a Cartesian path for a free-floating space robot. They generated a joint trajectory by differential integration of the end-effector velocity, and used the accuracy in both end-effector and joint spaces as objectives to be minimised. Wang et al., 2015 used PSO to generate a joint trajectory of a redundant single-arm free-floating robot, that satisfies manipulability, base disturbance, and goal reaching constraints. This work was extended by Wang et al., 2018 for a dual-arm case. Finally, Liu et al., 2015

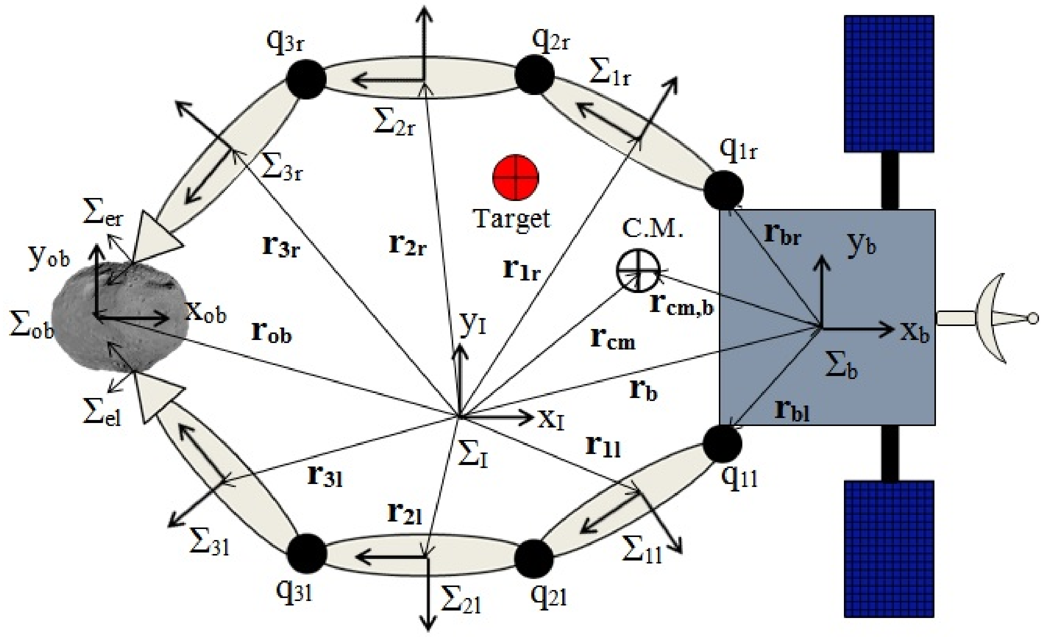


Figure 7.1: A 2D free-floating dual-arm space robot grasping an object. Σ_I, Σ_b, Σ_{ob} note the inertial , base and object coordinate frames. Σ_{1r}, Σ_{2r}, Σ_{3r}, Σ_{1l}, Σ_{2l}, Σ_{3l}, Σ_{el}, Σ_{er} note the frames of the links and end-effector of each arm. r_{cm}, r_b, $r_{cm,b}$ note the vectors from the inertial frame to the system barycenter C.M., the inertial frame to the base frame, and from the base frame to the C.M. frame. $\boldsymbol{r}_{1r}$, $\boldsymbol{r}_{2r}$, $\boldsymbol{r}_{3r}$, $\boldsymbol{r}_{1l}$, $\boldsymbol{r}_{2l}$, $\boldsymbol{r}_{3l}$ note the vectors from the inertial frame to each link, and $\boldsymbol{r}_{ob}$ is the vector from the inertial frame to the object. $\boldsymbol{r}_{br}$ and $\boldsymbol{r}_{bl}$ are the vectors from the base to the first joint of each arm. The arm joints are noted with $q_{1r}, q_{2r}, q_{3r}, q_{1l}, q_{2l}, q_{3l}$. The object needs to be transferred to a target location, shown in red.

used PSO to minimise dynamics and efficiency related objectives, for the case of a single-arm free-floating robot that handles a heavy payload.

The aforementioned literature solves the problem primarily by generating optimisation criteria that are related to the efficiency of the conducted operations. In the same time, an on-orbit robot needs to ensure the safety of the handled object. For example, a generated trajectory may induce increased stress on a handled piece of debris, increasing the risk of breaking it. The core idea of this chapter is a multi-objective trajectory generation methodology for a free-floating dual-arm space robot, that aims to primarily reduce the induced stress on the captured object during handling. For that matter, the combined dynamics of the object and spacecraft are employed, to calculate the internal stress forces created in the object while being handled, and minimise them with a Genetic Algorithm along other classical criteria. Minimising the internal forces of a captured object in a multi-objective way is a novel key aspect of this Chapter that has not been studied.

7.2 Methodology

The followed methodology is described in this section.

7.2.1 Spacecraft and object dynamics

The joint-space dynamics of a planar, dual-arm Free Floating Space Manipulator (FFSM) shown in Fig. 7.1 are found by expressing the kinetic energy of the spacecraft and arms as function of the joint angles, and using the Lagrange-d'Alembert equation. They are given by:

$$\begin{pmatrix} H_b & H^r_{bm} & H^l_{bm} \\ H^{rT}_{bm} & H^r_m & 0 \\ H^{lT}_{bm} & 0 & H^l_m \end{pmatrix} = \begin{pmatrix} \ddot{\boldsymbol{x}}_b \\ \ddot{\boldsymbol{q}}_r \\ \ddot{\boldsymbol{q}}_l \end{pmatrix} + \begin{pmatrix} \boldsymbol{c}_b \\ \boldsymbol{c}^r_m \\ \boldsymbol{c}^l_m \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_b \\ \boldsymbol{\tau}_r \\ \boldsymbol{\tau}_l \end{pmatrix} \tag{7.1}$$

where $\boldsymbol{x}_b$ is the vector of linear and angular base position and orientation w.r.t. the inertial frame, $\boldsymbol{q}_r$,$\boldsymbol{q}_l$ are the vectors of joint coordinates for the right and left arm, H_b,H^r_m,H^l_m the inertia matrices of the base and manipulators, H^r_{bm} and H^l_{bm} the dynamic coupling matrices between base and arms, $\boldsymbol{c}_b$,$\boldsymbol{c}^l_m$,$\boldsymbol{c}^r_m$ the Coriolis vectors for the base and arms, $\boldsymbol{f}_b$ the actuating force on the base, and $\boldsymbol{\tau}_r$ and $\boldsymbol{\tau}_l$ the actuating torques on the arms.

Since the system is free-floating, it is assumed that $\boldsymbol{f}_b = 0$, and thus the motion of the base is only affected by the coupling of the manipulators' motions. The conservation law of linear and angular momentum $\boldsymbol{P}_o$ and $\boldsymbol{L}_o$ for the system is given by:

$$\begin{pmatrix} \boldsymbol{P}_o \\ \boldsymbol{L}_o \end{pmatrix} = \begin{pmatrix} H_b\dot{\boldsymbol{x}}_b + H^r_{bm}\dot{\boldsymbol{q}}_r + H^l_{bm}\dot{\boldsymbol{q}}_l \end{pmatrix} \tag{7.2}$$

It is assumed that the robot stabilises after grasping the object, through thruster actuation. In this case $\boldsymbol{P}_o = \boldsymbol{L}_o$ = 0. The spacecraft base motion is given by:

$$\dot{\boldsymbol{x}}_b = \left(-H_b^{-1}H^r_{bm} - H_b^{-1}H^l_{bm}\right)\begin{pmatrix} \dot{\boldsymbol{q}}_r \\ \dot{\boldsymbol{q}}_l \end{pmatrix} = J_a\dot{\boldsymbol{q}} \tag{7.3}$$

The differential kinematics of the end-effectors are a function of both base and joint motion:

$$\begin{pmatrix} \dot{\boldsymbol{x}}^r_e \\ \dot{\boldsymbol{x}}^l_e \end{pmatrix} = \begin{pmatrix} J^r_b\dot{\boldsymbol{x}}_b + J^r_e\dot{\boldsymbol{q}}_r \\ J^l_b\dot{\boldsymbol{x}}_b + J^l_e\dot{\boldsymbol{q}}_l \end{pmatrix} \tag{7.4}$$

where J^r_b, J^l_b are the Jacobians of the base towards each arm, and J^r_e, J^l_e the Jacobians from the start of each arm to its end-effector. Substituting Eq. (7.3) to Eq. (7.4), the result is a mapping from the arm joints to the end-effector velocities:

$$\begin{pmatrix} \dot{\boldsymbol{x}}^r_e \\ \dot{\boldsymbol{x}}^l_e \end{pmatrix} = \begin{pmatrix} [J^r_e - J^r_bH_b^{-1}H^r_{bm}][-J^r_bH_b^{-1}H^l_{bm}] \\ [-J^l_bH_b^{-1}H^r_{bm}J^l_e][-J^l_bH_b^{-1}H^l_{bm}] \end{pmatrix} = J_g\dot{\boldsymbol{q}} \tag{7.5}$$

The matrix J_g is known as the Generalized Jacobian Matrix of the system (Umetani et al., 1989).

The next step is to eliminate x_b from Eq. (7.1) to get:

$$H_{el}\ddot{\boldsymbol{q}} + \boldsymbol{c}_{el} = \boldsymbol{\tau} \tag{7.6}$$

with

$$H_{el} = \begin{pmatrix} [-H_{bm}^{rT} H_b^{-1} H_{bm}^{r} + H_m^{r}] & [-H_b^{-1} H_{bm}^{l}] \\ [-H_b^{-1} H_{bm}^{r}] & [-H_{bm}^{lT} H_b^{-1} H_{bm}^{l} + H_m^{l}] \end{pmatrix} \tag{7.7}$$

and

$$\boldsymbol{c}_{el} = \begin{pmatrix} \boldsymbol{c}_m^{r} - H_b^{-1} \boldsymbol{b}_b \\ \boldsymbol{c}_m^{l} - H_b^{-1} \boldsymbol{b}_b \end{pmatrix} \tag{7.8}$$

The dynamics of the system are then projected to the captured object frame. This provides an expression for the applied wrench $\boldsymbol{F}_{tot}$ in the object coordinates. To do so, one can use use the Generalized Jacobian J_g to project from joint coordinates to end-effector frame as well as the grasp map G to project from end-effector frame to object frame. Moreover, the inertia and Coriolis matrices of the captured object Λ_{ob} and μ_{ob} are added. As mentioned, the dynamics of the object are assumed known, and this could be achieved by one of the methods in the existing literature (Chu et al., 2017; Murotsu et al., 1994; Pesce et al., 2017). More details about the projection and the use of Generalized Jacobian can be found by Murray et al., 1994,Russakow et al., 1995,and Abiko et al., 2008.

The total dynamic model expressed in the object frames is (Murray et al., 1994):

$$\Lambda_{tot}\ddot{\boldsymbol{x}} + \mu_{tot} = \boldsymbol{F}_{tot} \tag{7.9}$$

with

$$\Lambda_{tot} = GJ_g^{-T} H_{el} J_g^{-1} G^T + \Lambda_{ob} \tag{7.10}$$

$$\mu_{tot} = GJ_g^{-T}(\boldsymbol{c}_{el} J_g^{-1} G^T + \Lambda_{tot}\frac{d}{dt}(J_g)^{-1} G^T) + \mu_{ob} \tag{7.11}$$

To express the object coordinates $\ddot{x}$ as function of joints, one can use the grasp constraint $J_g\dot{q} = G^T\dot{\boldsymbol{x}}$ (ibid.). It is clear that the result is a formula that maps the joint configuration to the total applied wrench. The next step is to find the joint trajectory that minimises the applied wrench.

7.2.2 Trajectory formulation

The joint trajectory to be executed by the robot needs to be smooth in order to avoid sudden jumps in motion. A way to design smooth robot joint trajectories is to formulate them as n-degree polynomials. As the polynomial degree increases, the smoothness of higher derivatives (velocity, acceleration etc) of the joint position is ensured. In order to ensure smoothness up to the joint acceleration, a 7-degree polynomial is used. The joint trajectories, velocities and accelerations in time t are defined as follows:

$$q(t) = L_1 sin(\sum_{i=0}^{7} \lambda_i t^i) + L_2 \tag{7.12}$$

$$\dot{q}(t) = L_1 cos(\sum_{i=0}^{7} \lambda_i t^i)(\sum_{i=1}^{7} i\lambda_i t^{i-1})^2 \tag{7.13}$$

$$\ddot{q}(t) = -L_1 sin(\sum_{i=0}^{7} \lambda_i t^i)(\sum_{i=1}^{7} i\lambda_i t^{i-1})^2 + \tag{7.14}$$

$$L_1 cos(\sum_{i=0}^{7} \lambda_i t^i)(\sum_{i=2}^{7} i(i-1)\lambda_i t^{i-2})$$

The variables $L_1 = (q_{max} - q_{min})/2$ and $L_2 = (q_{max} + q_{min})/2$ are used along with the sinusoid to constrain the joint values in their allowed limits, q_{max} and q_{min}. The λ_i coefficients need to be determined for each joint.

In this case, it is assumed that each joint starts from an initial value q_{init}. As the captured object needs to be transferred to a target location (e.g. a storing compartment on the spacecraft), this will lead to a final value q_{fin}. The q_{fin} configuration for each joint will be the result of the inverse kinematics planner.

It is also assumed that the arm starts the trajectory from rest after grasping the object, and that the arm stops as soon as it reaches the desired configuration. Thus, for each joint $\dot{q}_{init} = \dot{q}_{fin} = \ddot{q}_{init} = \ddot{q}_{fin} = 0$. The total motion is executed from $t_0 = 0$ sec, to t_f sec. By substituting these assumptions in Eqs. (7.12-7.14), it follows for each joint:

$$\begin{aligned}
\lambda_0 &= arcsin(\frac{q_{init} - L_2}{L_1}) \\
\lambda_1 &= \lambda_2 = 0 \\
\lambda_3 &= -\frac{(3\lambda_7 t_f^7 + \lambda_6 t_f^6 - 10L_3 - 10L_4)}{t_f^3} \\
\lambda_4 &= \frac{(8\lambda_7 t_f^7 + 3\lambda_6 t_f^6 - 15L_3 - 15L_4)}{t_f^4} \\
\lambda_5 &= -\frac{(6\lambda_7 t_f^7 + 3\lambda_6 t_f^6 - 6L_3 - 6L_4)}{t_f^5} \\
L_3 &= arcsin(\frac{q_{fin} - L_2}{L_1}) \\
L_4 &= \lambda_0
\end{aligned} \tag{7.15}$$

This leaves each joint's λ_6 and λ_7 coefficients to be determined by minimising the objectives. Thus, the outcome of the optimization process is a 12x1 coefficient vector λ .

7.2.3 Definition of objective functions

A space robot operation typically requires a lot of objectives to be considered, increasing the problem complexity. In this case, the selected objectives are based on the careful handling of the object, as well as the accuracy of the motion. For simplicity, 3 objectives were selected for minimisation, but more can be included.

Minimisation of object stress

The basic property to be minimised is the stress applied on the object during motion. This reduces the risk of the object breaking apart during the motion. The forces applied on the object by the grasping arms, can be decomposed in motion-generating and internal forces. This decomposition happens by using the pseudoinverse $G^{\dagger}$ and projecting the forces in the null-space of the grasp map G (Murray et al., 1994).

$$\boldsymbol{F} = G^{\dagger}_{tot} + (I - G^{\dagger}G)\boldsymbol{k} \tag{7.16}$$

In this case F_{tot} is the wrench that results in object motion, and is equal to the projected dynamics of the spacecraft and arms in the object frame. $\boldsymbol{k}$ can be any real vector. To generate a minimum-internal force objective, the analysis by Nakamura et al., 1991 is followed. Let the captured object to be rigid, with a massless elastic layer on top. The elastic layer has spring-like behaviour. Starting from these assumptions, and assuming rigid object grasping from the arms, ibid. concluded that the optimal applied wrench that minimises the induced stress on the object, while under a fixed grasp, is proportional to $\boldsymbol{F}_{tot}$. Consequently, it is proportional to the projected dynamics on the object frame. As a result, the first three objective functions are selected as the norm of each component of the calculated $\boldsymbol{F}_{tot}$ signal during motion:

$$\begin{aligned} C_{f_1} &= ||f_{tot_x}|| \\ C_{f_2} &= ||f_{tot_y}|| \\ C_{f_3} &= ||n_{tot}|| \end{aligned} \tag{7.17}$$

The norm along the total trajectory was selected to obtain an overall indication of the applied wrench over the whole motion. Another interesting minimisation criterion would be to restrict the applied wrench under some threshold values for each motion instance.

Minimisation of base disturbance

The base disturbance along the motion can be minimised by integration of Eq. (7.3) along the trajectory time (Wang et al., 2018). Starting from a given spacecraft base pose $\boldsymbol{x}_{b_{init}}$, and resulting in a final pose $\boldsymbol{x}_{b_{fin}}$, the base disturbance is given by :

$$\boldsymbol{x}_{b_{fin}} - \boldsymbol{x}_{b_{init}} = \boldsymbol{x}_{dist_b} = \int_{t_0}^{t_f} J_a \dot{\boldsymbol{q}} dt \tag{7.18}$$

For limited disturbance, the final pose needs to be the same as the initial. As a result, one can generate the objective functions for the base disturbance as the 3 components of the above integral $\boldsymbol{x}_{dist_b}$:

$$\begin{aligned} C_{f_4} &= ||x_{dist_{bx}}|| \\ C_{f_5} &= ||x_{dist_{by}}|| \\ C_{f_6} &= ||x_{dist_{b\theta}}|| \end{aligned} \tag{7.19}$$

Minimisation of end-effector disturbance

The accuracy of the task models how close the space robot has brought the captured object to a desired pose. It is he difference of the initial to the final pose of the end-effectors (Wang et al., 2018):

$$\boldsymbol{x}_{e_{fin}} = \boldsymbol{x}_{e_{init}} + \int_{t_0}^{t_f} J_g \dot{\boldsymbol{q}} dt \tag{7.20}$$

This means that for a desired target end-effector pose $\boldsymbol{x}_{e_{des}}$, one needs to to minimise the difference between the final and desired poses. The task accuracy objective functions are defined as follows:

$$\begin{aligned} C_{f_7} &= ||(x_{e_{fin}} - x_{e_{des}})_x|| \\ C_{f_8} &= ||(x_{e_{fin}} - x_{e_{des}})_y|| \\ C_{f_9} &= ||(x_{e_{fin}} - x_{e_{des}})_\theta|| \end{aligned} \tag{7.21}$$

7.2.4 Genetic algorithm for multi-objective optimisation

The definition of the multi-objective optimisation problem is:

$$\begin{aligned} \underset{\lambda}{\text{Minimize}} \quad & C_f(\lambda) = (C_{f_1}(\lambda), C_{f_2}(\lambda), ..., C_{f_9}(\lambda)) \\ \text{subject to} \quad & \lambda \in [\lambda_{min}, \lambda_{max}] \end{aligned} \tag{7.22}$$

Many approaches have been developed for solving such type of problems. Analytical methods can be very complex and time consuming, and as such a meta-heuristic method was chosen, to achieve an optimal solution for all objectives and constraints in finite time. Specifically, a variation of a Genetic Algorithm is used, the *Multi-Objective Evolutionary Algorithm with Decomposition* (*MOEAD*). The followed approach is briefly described below. A comprehensive description of the method can be found by Zhang et al., 2007.

In an evolutionary algorithm, an initial *population* of randomly generated solutions, is examined according to the objective (or *fitness*) function. The best solutions are then selected as *parents* that produce an offspring according to a set of *genetic operations*, such as *crossover* or *mutation*. These operators result in new offspring populations with better fitness, and the process repeats. The algorithm termination criteria include achieving a specified fitness value, or stopping after a finite execution time among others.

In a MOEAD framework, the problem is decomposed in different optimisation sub-problems, each one solved simultaneously. The solutions of the sub-problems can be combined, increasing efficiency and allowing better convergence to a global

Table 7.1: Properties of the simulated spacecraft and objects

Link	Mass (kg)	Inertia ($kg \times m^2$)
Base	300	200
1r	5	0.1
2r	4	0.08
3r	3	0.06
1l	5	0.1
2l	4	0.08
3l	3	0.06
Object	10	0.64

solution. By splitting into sub-problems, one is able to keep a solution that satisfies a sub-problem, and check the solution's neighbourhood. If for example, the algorithm manages to minimise the applied wrench, the other solutions for the other sub-problems are checked around the wrench-minimising solution. The solutions eventually form a *Pareto frontier*, namely a frontier in which it is impossible to decrease the value of an objective, without increasing the value of another. After all sub-problems have terminated, a the solution in the Pareto frontier that minimises all the objectives is returned as a global solution.

7.3 Numerical evaluation

To test the method, a numerical simulation for a dual-arm free-flying space robot was set up. For simplicity, each arm has 3DOF and executes a 2D planar motion, as shown in Fig. 7.1. The spacecraft is assumed to hold a captured object in a fixed grasp G. The object needs to be moved to a given target position. The inertial properties of the spacecraft are shown in Table 1. The base is modelled as a cube with side length of $2m$, and the links are modelled as cylinders with radius of $0.2m$, and lengths of $1.2m$, $1.1m$ and $1m$ for the first, second and third link of each arm. The object is modelled as a solid sphere with radius of $0.4m$, with uniform density.

The spacecraft is assumed to have grasped the object from joint configuration q_{init}, and after the motion, it reaches a predefined configuration q_{fin}. The corresponding $x_{e_{des}}$ is calculated from the inverse kinematics of the system, to be used in the end-effector criterion.Zero initial and final velocities and accelerations are assumed, as well as motion time of $t_f = 50$ sec. The joint trajectories are sampled with a timestep $\delta t = 0.01$ sec.

The aim is to formulate the necessary trajectories, as per Eqs. (7.12-7.14). As a result, the population vector is random variations of the 12x1 vector
$\lambda = (\lambda_{6_1 r}, \lambda_{7_1 r}, ..., \lambda_{6_3 l}, \lambda_{7_3 l})$. Each element of the population vector is constrained in the $[-10, 10]$ value range.

For each population element λ_i, the following steps are executed:

1. Substitute the element λ_i and t_f in Eqs. (7.12-7.14) to get trajectories for the joint positions, velocities and accelerations for every timestep

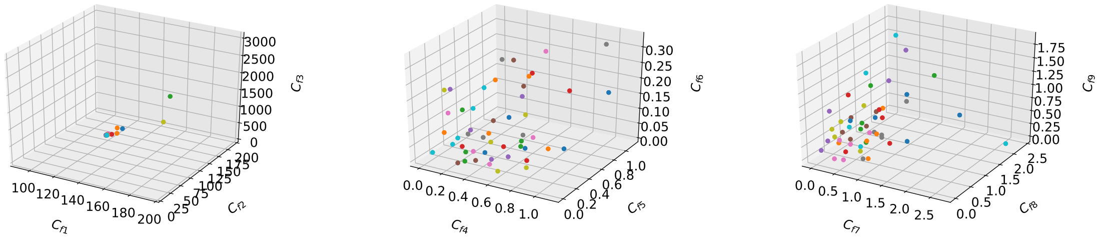


Figure 7.2: Pareto frontiers for each of the 3 optimisation sub-problems. The first sub-problem (left) consists of minimising the applied wrench objectives C_{f1} to C_{f3}. The second sub-problem (middle) consists of minimising the base disturbance objectives C_{f3} to C_{f6}, and the third (right) of minimising the end-effector disturbance objectives C_{f7} to C_{f9}. The results show that in all sub-problems, the algorithm is able to gradually minimise the given objectives, especially in the disturbances sub-problems, where the objective approaches zero. It is also evident from the results that indeed, the joint trajectory affects the applied wrench on the object, and that a minimum-force trajectory can be found.

2. Calculate the applied wrench, base disturbance, and end-effector disturbance using the joint trajectories

3. Calculate the fitness functions in Eqs. (7.17),(7.19),(7.21)

4. Select the best population individuals as parents and apply the mutation operator

After 5000 iterations, the algorithm terminates.

In total, 3 decomposed problems are chosen for the MOEAD framework, namely minimisation of applied wrenches, base disturbance and end-effector disturbance. The Pareto results are shown in Fig. 7.2, for each sub-problem. The results indicate that indeed the algorithm has found a set of polynomial coefficients that minimise all the objectives. As it can be seen, the selected coefficients produce polynomials with minimum applied wrenches on the object. In addition, both the base translation and rotation disturbances are very low, and the end-effectors reach the target with high accuracy.

Fig. 7.3 shows the trajectories for the optimal individual. It is clear that the trajectories are indeed smooth, minimising the risk for jumps during the motion.

It is evident that the algorithm succeeds in finding a solution that minimises all the objectives, with reasonable accuracy. The selected trajectory is a zero-jerk trajectory which has also intuitive meaning, as jerky motions on the joints would induce jerk on the end-effectors and vibrations on the object. The algorithm can be extended to include different starting and final joint values. This enables the evolutionary algorithm to find minimum objectives given any initial grasping configuration and any target position for the captured object. Additionally to the selected objectives, more fitness criteria can be generated, such as minimum-power and collision-avoiding criteria, and the Pareto-optimal solution can be found.

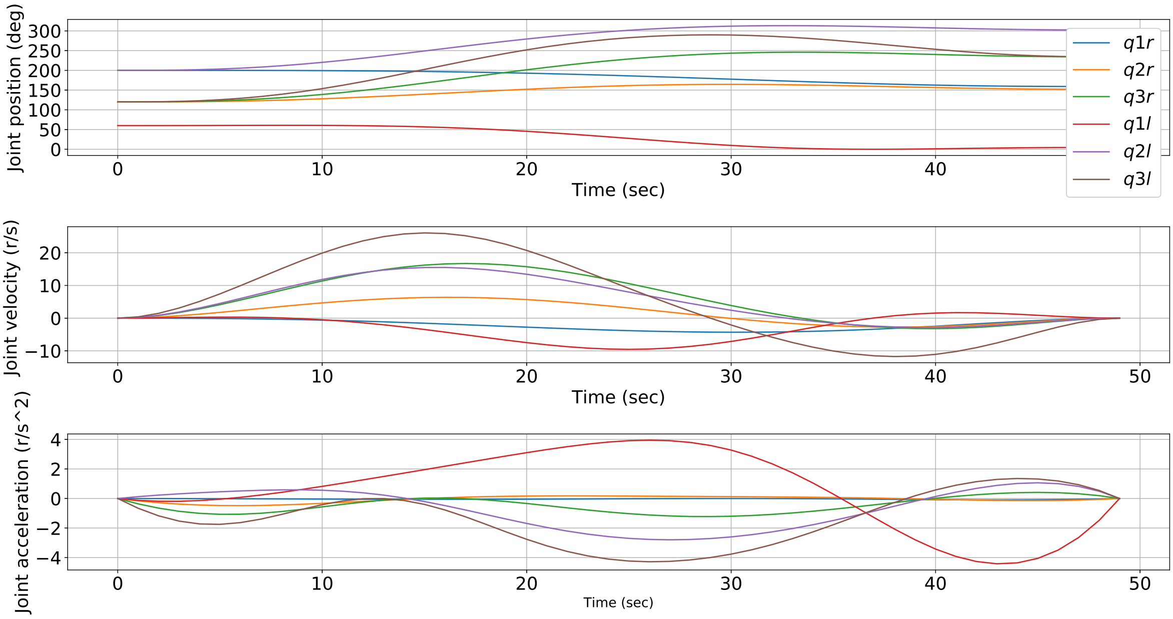


Figure 7.3: Joint trajectories for the optimal solution. The trajectories for the joint positions (up), velocities (middle) and accelerations (down) are smooth, avoiding possible jumps and jitters during the motion that would induce extra applied stress to the captured object.

7.4 Discussion

In this chapter, a multi-objective minimisation method was provided for an on-orbit dual-arm free-floating space robot, that seeks to find the trajectory to bring a captured object to the target point under minimum object strain and maximum task accuracy. The method was tested using a numerical simulation in a 2D robot, and the results showed that indeed, a minimum-force, accurate trajectory can be found. The analysis can be extended in a more complex a 3D case. Nevertheless, projecting the dynamics of a 3D redundant space robot in the object coordinate frame, may be a computationally intensive task. As a result, a possible extension for future work, would be to formulate the spacecraft dynamics and applied wrenches on the object using the less intensive Newton-Euler formulation. For simplicity, objectives related to motion efficiency (manipulability maximisation, power conservation etc) or collision avoidance for the generated trajectories were not examined.

In future work more objectives can be added, related to the oned developped in the previous Chapters. In addition, the method can be tested in a robot simulator, to further validate and compare different trajectories according to the applied forces. Finally, it would be interesting to verify the method using real manipulators, objects, and force sensors, ideally on an emulating dual-arm space robot testbed.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

This thesis addressed the two related problems of: firstly, estimating the inertial parameters of objects through exploratory and learning methods; and, secondly, using these parameters in manipulation tasks, to make robot actions safer, and more efficient. Additionally, the thesis provided a structured literature survey of classic works and current trends in both domains.

For the first part, the aim was to introduce a paradigm shift in the estimation procedure, that is data-driven using machine learning approaches. Initially, a large simulation dataset of 3D models along with their inertial parameters was created, in order to collect a relatively small amount of pushing data and evaluate the estimating method's performance. A random forest was trained to predict the mass and inertia of simulated cubes, and it was tested in other dataset objects. The performance was mixed, with the algorithm showing larger errors for objects that were not similar to those in the training data-set. To improve the performance and further investigate the problem by also estimating the CoM, a large and more diverse dataset of pushes was generated in simulation, and a learning algorithm was trained with richer features. This new algorithm was tested on both a part of the simulated data, as well as real data taken from the MCube pushing dataset. The results were greatly improved, demonstrating high accuracy, even with a single push. In general, the experiments conducted, demonstrated that the inertial parameters can be estimated using data-driven methods, achieving satisfactory performance with a variety of different objects.

For the second part, the aim was to demonstrate novel applications of manipulation tasks that exploit knowledge of the inertial parameters. For focus was the promising domain of task-informed grasping, i.e. selecting from a number of possible stable grasps, in order to optimise the post-grasp manipulative task of moving the object. The thesis provided several new metrics for choosing grasps, based on the requirements of the post-grasp manipulation task. The metrics were calculated from the inertial parameters of the grasped objects. The first metric introduced was the manipulation effort during the task execution. The effort was defined as the norm of the manipulation torque of all the joints. As the torque is connected to the consumed energy, this metric enables the robot to become power-efficient. The second metric was related to manipulation safety. The object's inertial parameters were used to create a metric that minimises the total effective mass of the combined manipulator and object. As the effective mass is related to the impact force, this metric is able to minimise the force in case the end effector collides with its surroundings, making manipulation safer for the robot and its environment or nearby humans. These metrics were then combined, along with manipulability metrics that show how far the robot is from singular configurations, to showcase the need of an multi-objective optimisation framework that depends on the task to be executed. Finally, a post-grasp manipulation metric was proposed, to minimise the internal forces of the object during manipulation and ensure the safety of the manipulated object. To demonstrate the usefulness of the research in different fields, an application in space robotics was examined.

8.2 Future work

The work presented in this thesis comprises a variety of novel methods, metrics and paradigms, supported by simple but demonstrative experiments. As a result, it can be extended in numerous ways. Some interesting topics for future work are described here.

- The inertial dataset described could be modified to include 3D models and inertial parameters of real objects. The 3D models can be extracted using techniques similar to those mentioned by Pokorny et al., 2017, and the inertial parameters with one of the fixed-object methods mentioned in Chapter 2. The dataset could then be used as a real-object benchmark for estimation and usage algorithms that offers additional credibility to the tested methods, as they would have been tested in real objects.

- The estimation methods presented in this thesis can be further augmented to include visual data, for the calculation of all 3D parameters. Under the assumption of uniform density, the problem of estimating the 3D inertial parameters, reduces to estimation of mass and volumetric distribution, as shown by Mirtich, 1996. In this case, A robot could calculate a 3D model from visual information (e.g. several point-cloud views of the object), use them to generate the volume distribution integrals, and calculate the mass from a single push as described in this thesis. In this case, the robot would have generated the full 3D inertial parameters from only a single push and visual info. This would be a breakthrough result that would have applications in different fields of autonomous robotics, and would enable the direct calculation of the manipulation metrics described in the thesis, for executing tasks. The inertial dataset could be used for testing and evaluation on different objects.

- The volumetric and mass distribution of an object is defined by using continuous integrals. An approximation is usually made by discretising the continuous 3D space into small segments, resulting in estimates of the volume and mass distributions which are defined as sums. In computer vision, some object representations that take advantage of this segmentation of 3D space are octrees and voxel grids. By expressing the object as a voxel grid or octree, the volume distribution is easier to calculate. Now, if a small mass could be assigned to every voxel, this would result in a "mass distribution voxel grid", related to those presented by Chien et al., 1986. This mass-weighted voxel grid would be a novel object representation that could be used for inertial estimation and object recognition.

- The manipulation criteria presented in this thesis, require perfect knowledge of the robot's dynamic model as well as the grasped object's inertial parameters. An interesting continuation would be to show how these criteria perform under the existence of uncertainties, either in the robot dynamics, the object's parameters or both. A novel criterion that takes uncertainties into consideration, could compensate for estimation errors, and could have extra applications in more realistic scenarios, where noise usually comes from multiple sources.
- Finally, additional manipulation criteria can be generated, related to grasp stability and force closure. As these two properties are a function of the grasp location relative to the CoM, and the object's mass, the inertial properties can be used to generate and select grasps that are maximally stable with respect to force-closure.

Bibliography

Abiko, Satoko and Gerd Hirzinger (2008). "Computational efficient algorithms for operational space formulation of branching arms on a space robot". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3312–3317.

Alambeigi, Farshid et al. (2011). "Simulation and control of a multi-dof laparoscopic tele-surgry system in virtual reality". In: *IEEE International Conference on Control, Instrumentation and Automation*, pp. 1060–1066.

Amazeen, Eric L and MT Turvey (1996). "Weight perception and the haptic size–weight illusion are functions of the inertia tensor". In: *Journal of Experimental Psychology: Human Perception and Performance* 22.1, p. 213.

Amraei, Somaye, Saman Abdanan Mehdizadeh, and Somayeh Sallary (2017). "Application of computer vision and support vector regression for weight prediction of live broiler chicken". In: *Engineering in Agriculture, Environment and Food* 10.4, pp. 266–271.

Angel, L. and J. Viola (2016). "Payload estimation for a robotic system using unsupervised classification". In: *Symposium on Signal Processing, Images and Artificial Vision*, pp. 2–6.

Artashes, Mkhitaryan and Darius Burschka (2013). "Visual estimation of object density distribution through observation of its impulse response". In: *International Conference on Computer Vision Theory and Applications*.

Atkeson, Chirstopher G., Chae H. An, and John M. Hollerbach (1985). "Rigid body load identification for manipulators". In: *IEEE Conference on Decision and Control* 24.

Bailey, DG et al. (2004). "High speed weight estimation by image analysis". In: *New Zealand National Conference on Non Destructive Testing*, pp. 27–39.

Barcio, B.T. and I.D. Walker (1994). "Impact ellipsoids and measures for robot manipulators". In: *IEEE International Conference on Robotics and Automation*, pp. 1588–1594.

Beetz, M. et al. (2015). "Robotic agents capable of natural and safe physical interaction with human co-workers". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 6528–6535.

Bingham, G P and M M Muchisky (1993). "Center-of-mass perception and inertial frames of reference". In: *Perception & Psychophysics* 54.5, pp. 617–632.

Borchani, Hanen et al. (2015). "A survey on multi-output regression". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5.5, pp. 216–233.

Borst, Ch., M. Fischer, and G. Hirzinger (2004). "Grasp planning: how to choose a suitable task wrench space". In: *IEEE International Conference on Robotics and Automation* 1, pp. 319–325.

Breiman, Leo (2001). "Random forests". In: *Machine learning* 45.1, pp. 5–32.

Bruyninckx, H., S. Demey, and V. Kumar (1998). “Generalized stability of compliant grasps”. In: *IEEE International Conference on Robotics and Automation* 3, pp. 2396–2402.

Buchholz, Dirk et al. (2014). “Combining visual and inertial features for efficient grasping and bin-picking”. In: *IEEE International Conference on Robotics and Automation*, pp. 875–882.

Calli, Berk et al. (2015). “The YCB object and model set: Towards common benchmarks for manipulation research”. In: *IEEE International Conference on Advanced Robotics*, pp. 510–517.

Castiello, Umberto (2005). “The neuroscience of grasping”. In: *Nature Reviews Neuroscience* 6.9, pp. 726–736.

Castronuovo, Marco (2011). “Active space debris removal—A preliminary mission analysis and design”. In: *Acta Astronautica* 69.9-10, pp. 848–859.

Cehajic, Denis, Pablo Budde Gen Dohmann, and Sandra Hirche (2017). “Estimating unknown object dynamics in human-robot manipulation tasks”. In: *IEEE International Conference on Robotics and Automation*, pp. 1730–1737.

Chang, K S, R Holmberg, and O Khatib (2000). “The augmented object model: cooperative manipulation and parallel mechanism dynamics”. In: *IEEE International Conference on Robotics and Automation* 1, pp. 470–475.

Chang, Michael B. et al. (2016). “A Compositional Object-Based Approach to Learning Physical Dynamics”. In: *arXiv:1612.00341 (2016).*

Chavan-Dafle, N. and A. Rodriguez (2015). “Prehensile pushing: In-hand manipulation with push-primitives”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 6215–6222.

Chien, C. H. and J. K. Aggarwal (1986). “Identification of 30 Objects from multiple silhouettes using quadtrees / octrees ”. In: *Computer Vision, Graphics, and Image Processing* 36, pp. 256–273.

Chinthaka, M K C Dinesh and Tomoyuki Shimono (2017). “Inertia estimation of robot end effector for dextrous object grasping”. In: *Annual Conference of the IEEE Industrial Electronics Society*, pp. 5209–5214.

Chu, Zhongyi et al. (2017). “Inertial parameter identification using contact force information for an unknown object captured by a space manipulator”. In: *Acta Astronautica* 131, pp. 69–82.

Cole, Arlene, John Hauser, and Shankar Sastry (1988). “Kinematics and control of multifingered hands with rolling contact”. In: *IEEE International Conference on Robotics and Automation*, pp. 228–233.

Cole, Arlene A, Ping Hsu, and S Shankar Sastry (1992). “Dynamic control of sliding by robot hands for regrasping”. In: *IEEE Transactions on robotics and automation* 8, pp. 42–52.

Corah, Micah and Nathan Michael (2017). “Active estimation of mass properties for safe cooperative lifting”. In: *IEEE International Conference on Robotics and Automation*, pp. 4582–4587.

Delponte, Elisabetta et al. (2006). “Analysis on a local approach to 3d object recognition”. In: *Joint Pattern Recognition Symposium*, pp. 253–262.

Ding, Dan, Yun-Hui Lee, and Shuguo Wang (2001). “Computation of 3-D form-closure grasps”. In: *IEEE Transactions on Robotics and Automation* 17.4, pp. 515–522.

Diuk, Carlos, Andre Cohen, and Michael L Littman (2008). “An object-oriented representation for efficient reinforcement learning”. In: *ACM International Conference on Machine learning*, pp. 240–247.

Dong, Yunfei et al. (2018). “An efficient robot payload identification method for industrial application”. In: *Industrial Robot: An International Journal* 45.4, pp. 505–515.

Dubowsky, Steven and Miguel A Torres (1991). “Path planning for space manipulators to minimize spacecraft attitude disturbances”. In: *IEEE International Conference on Robotics and Automation*, pp. 2522–2528.

Duda, Richard O, Peter E Hart, and David G Stork (2012). *Pattern classification*. John Wiley & Sons.

Dutkiewicz, P., K. R. Kozlowski, and W. S. Wroblewski (1993). “Experimental identification of robot and load dynamic parameters”. In: *IEEE International Conference on Control and Applications* 2, pp. 767–776.

Ellis, Robert R and Susan J Lederman (1993). “The role of haptic versus visual volume cues in the size-weight illusion”. In: *Perception & Psychophysics* 53.3, pp. 315–324.

Farsoni, Saverio et al. (2018). “Real-time identification of robot payload using a multirate quaternion-based kalman filter and recursive total least-squares”. In: *IEEE International Conference on Robotics and Automation*, pp. 2103–2109.

Fazeli, Nima, Russ Tedrake, and Alberto Rodriguez (2017a). “Identifiability analysis of planar rigid-body frictional contact”. In: *Robotics Research*, pp. 665–682.

Fazeli, Nima et al. (2017b). “Parameter and contact force estimation of planar rigid-bodies undergooing frictional contact”. In: *The International Journal of Robotics Research* 36, pp. 1437–1454.

Ferrari, Carlo and John Canny (1992). “Planning optimal grasps”. In: *IEEE/RSJ International Conference on Robotics and Automation*, pp. 2290–2295.

Flacco, F. et al. (2012). “A depth space approach to human-robot collision avoidance”. In: *IEEE International Conference on Robotics and Automation*, pp. 338–345.

Flanagan, J Randall and Michael A Beltzner (2000). “Independence of perceptual and sensorimotor predictions in the size–weight illusion”. In: *Nature Neuroscience* 3.7, p. 737.

Franchi, Antonio, Antonio Petitti, and Alessandro Rizzo (2014). “Distributed estimation of the inertial parameters of an unknown load via multi-robot manipulation”. In: *IEEE Annual Conference on Decision and Control*, pp. 6111–6116.

— (2015). “Decentralized parameter estimation and observation for cooperative mobile manipulation of an unknown load using noisy measurements”. In: *IEEE International Conference on Robotics and Automation*, pp. 5517–5522.

Fukuda, K. and S. Tsujio (1999). “Estimation of mass and center of mass of graspless and shape-unknown object”. In: *IEEE International Conference on Robotics and Automation* 4, pp. 2893–2898.

Gautier, Maxime (1997). “Dynamic identification of robots with power model”. In: *IEEE International Conference on Robotics and Automation* 3, pp. 1922–1927.

Ghalamzan, E Amir M et al. (2017). “Human-in-the-loop optimisation: mixed initiative grasping for optimally facilitating post-grasp manipulative actions”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3386–3393.

Ghalamzan E, Amir M et al. (2016). "Task-relevant grasp selection: a joint solution to planning grasps and manipulative motion trajectories". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Goyal, Suresh, Andy Ruina, and Jim Papadopoulos (1991a). "Planar sliding with dry friction Part 1. Limit surface and moment function". In: *Wear* 143.2, pp. 331–352.

— (1991b). "Planar sliding with dry friction Part 2. Dynamics of motion". In: *Wear* 143.2, pp. 331–352.

Haddadin, S. et al. (2012). "On making robots understand safety: Embedding injury knowledge into control". In: *The International Journal of Robotics Research* 31.13, pp. 1578–1602.

Hamrick, Jessica B. et al. (2016). "Inferring mass in complex scenes by mental simulation". In: *Cognition* 157, pp. 61–76.

Hayati, Samad (1986). "Hybrid position/force control of multi-arm cooperating robots". In: *International Conference on Robotics and Automation* 3, pp. 82–89.

Heinzmann, J. and A. Zelinsky (2003). "Quantitative safety guarantees for physical human-robot interaction". In: *The International Journal of Robotics Research* 22.7-8, pp. 479–504.

Hu, Jingchen and Tianshu Wang (2016). "Pre-impact configuration designing of a robot manipulator for impact minimization". In: *Journal of Mechanisms and Robotics*.

Ikuta, Koji, Hideki Ishii, and Makoto Nokata (2003). "Safety evaluation method of design and control for human-care robots". In: *The International Journal of Robotics Research* 22.5, pp. 281–297.

Kang, Sungchul et al. (2010). "Control of impulsive contact force between mobile manipulator and environment using effective mass and damping controls". In: *International Journal of Precision Engineering and Manufacturing* 11.5, pp. 697–704.

Kanoulas, Dimitrios et al. (2018). "Center-of-mass-based grasp pose adaptation using 3D range and force/torque sensing". In: *IEEE-RAS International Journal of Humanoid Robotics*, p. 1850013.

Kasper, alexander, zhixing xue, and Rüdiger dillmann (2012). "The KIT object models database: An object model database for object recognition, localization and manipulation in service robotics". In: *The International Journal of Robotics Research* 31.8, pp. 927–934.

Khatib, Oussama (1987). "Object manipulation in a multi-effector robot system". In: *International Symposium of Robotics Research*, pp. 131–138.

— (1995). "Inertial properties in robotic manipulation: An object-level framework". In: *The International Journal of Robotics Research* 14.1, pp. 19–36.

Khosla, Pradeep K and Takeo Kanade (1985). "Parameter identification of robot dynamics". In: *IEEE Conference on Decision and Control* 24, pp. 1754–1760.

Kim, Jinhyun, Wankyun Chung, and Youngil Youm (2000). "Normalized impact geometry and performance index for redundant manipulators". In: *IEEE International Conference on Robotics and Automation* 2, 1714–1719 vol.2.

Kingma, Idsart, Peter J. Beek, and Jaap H. van Dieën (2002). "The inertia tensor versus static moment and mass in perceiving length and heaviness of hand-wielded rods". In: *Journal of Experimental Psychology: Human Perception and Performance* 28.1, pp. 180–191.

Kolbert, Roman, Nikhil Chavan-Dafle, and Alberto Rodriguez (2016). "Experimental validation of contact dynamics for in-hand manipulation". In: *International Symposium on Experimental Robotics*, pp. 633–645.

Kopicki, Marek et al. (2015). "One-shot learning and generation of dexterous grasps for novel objects". In: *The International Journal of Robotics Research*.

Krotkov, Eric (1995). "Robotic perception of material". In: *International Joint Conferences on Artificial Intelligence* 1, pp. 88–95.

Kubus, Daniel, Torsten Kröger, and Friedrich M. Wahl (2007). "On-line rigid object recognition and pose estimation based on inertial parameters". In: *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, pp. 1402–1408.

Kubus, Daniel, Torsten Kroger, and Friedrich M. Wahl (2008). "On-line estimation of inertial parameters using a recursive total least-squares approach". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3845–3852.

Lee, Hyeonbeom and H. Jin Kim (2016). "Estimation, control and planning for autonomous aerial transportation". In: *IEEE Transactions on Industrial Electronics* 64, pp. 3369–3379.

Lee, Soo Hong and MR Cutkosky (1991). "Fixture planning with friction". In: *Journal of Engineering for Industry* 113.3, pp. 320–327.

Lenz, Ian, Honglak Lee, and Ashutosh Saxena (2015). "Deep learning for detecting robotic grasps". In: *The International Journal of Robotics Research* 34.4-5, pp. 705–724.

Lin, Z. C., R. V. Patel, and C. A. Balafoutis (1995). "Impact reduction for redundant manipulators using augmented impedance control". In: *Journal of Robotic Systems* 12.5, pp. 301–313.

Lines, J. A. et al. (2001). "An automatic image-based system for estimating the mass of free-swimming fish". In: *Computers and Electronics in Agriculture* 31.2, pp. 151–168.

Lippiello, V. (2015). "Grasp the Possibilities". In: *IEEE Robotics & Automation Magazine*, pp. 69–79.

Lippiello, V., B. Siciliano, and L. Villani (2013). "Multi-fingered grasp synthesis based on the object dynamic properties". In: *Robotics and Autonomous Systems* 61, pp. 626–636.

Liu, Yong et al. (2015). "Multi-objective trajectory planning of FFSM carrying a heavy payload". In: *International Journal of Advanced Robotic Systems*, p. 118.

Lopez-Damian, Efrain, Daniel Sidobre, and Rachid Alami (2005). "A grasp planner based on inertial properties". In: *IEEE International Conference on Robotics and Automation* 2005, pp. 754–759.

Lynch, K.M., H. Maekawa, and K. Tanie (1991). "Manipulation and active sensing by pushing using tactile feedback". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems* 1, pp. 416–421.

Marino, Alessandro, Giuseppe Muscio, and Francesco Pierri (2017). "Distributed cooperative object parameter estimation and manipulation without explicit communication". In: *IEEE International Conference on Robotics and Automation*, pp. 2110–2116.

Mason, M. T. (1986). "Mechanics and planning of manipulator pushing operations". In: *The International Journal of Robotics Research* 5.3, pp. 53–71.

Masuta, Hiroyuki et al. (2015). "Direct perception and action system for unknown object grasping". In: *IEEE International Workshop on Robot and Human Interactive Communication*, pp. 313–318.

Masuta, Hiroyuki et al. (2016a). "Direct perception of easily visible information for unknown object grasping". In: *Proceedings of International Conference on Intelligent Robotics and Applications*, pp. 78–89.

Masuta, Hiroyuki et al. (2016b). "The concept of direct perception for cognitive robotics". In: *International Symposium on Micro-NanoMechatronics and Human Science*, pp. 1–7.

Mazanek, Daniel D et al. (2015). "Asteroid redirect mission concept: a bold approach for utilizing space resources". In: *Acta Astronautica* 117, pp. 163–171.

Mellinger, Daniel et al. (2011). "Design, modeling, estimation and control for aerial grasping and manipulation". In: *IEEE International Conference on Intelligent Robots and Systems*.

Methil, Nandagopal S. and Ranjan Mukherjee (2006). "Pushing and steering wheelchairs using a holonomic mobile robot with a single arm". In: *IEEE International Conference on Intelligent Robots and Systems*, pp. 5781–5785.

Miller, Andrew T and Peter K Allen (2004). "Graspit! a versatile simulator for robotic grasping". In: *IEEE Robotics & Automation Magazine* 11.4, pp. 110–122.

Mirtich, Brian (1996). "Fast and Accurate Computation of Polyhedral Mass Properties". In: *Journal of Graphics Tools* 1, pp. 31–50.

Montana, David J. (1992). "Contact Stability for Two-Fingered Grasps". In: *IEEE Transactions on Robotics and Automation* 8.4, pp. 421–430.

Murooka, Masaki et al. (2014). "Manipulation strategy decision and execution based on strategy proving operation for carrying large and heavy objects". In: *IEEE International Conference on Robotics and Automation*, pp. 3425–3432.

Murooka, Masaki et al. (2017). "Feasibility Evaluation of Object Manipulation by a Humanoid Robot Based on Recursive Estimation of the Object ′ s Physical Properties". In: *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 4082–4089.

Murotsu, Yoshisada et al. (1994). "Parameter identification of unknown object handled by free-flying space robot". In: *Journal of Guidance, Control, and Dynamics*, pp. 488–494.

Murray, David J et al. (1999). "Charpentier (1891) on the size—weight illusion". In: *Perception & Psychophysics* 61.8, pp. 1681–1685.

Murray, Richard M et al. (1994). *A mathematical introduction to robotic manipulation*. CRC press.

Nakamura, Y. and R. Mukherjee (1991). "Nonholonomic path planning of space robots via a bidirectional approach". In: *IEEE Transactions on Robotics and Automation*, pp. 500–514.

Ni, D. et al. (2018). "Translational objects dynamic modeling and correction for point cloud augmented virtual reality-–based teleoperation". In: *Advances in Mechanical Engineering* 10, pp. 1–14.

Novin, Roya Sabbagh et al. (2018). "Dynamic model learning and manipulation planning for objects in hospitals using a patient assistant mobile (PAM) robot". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1–7.

Olsen, H. and G. Bekey (1985). “Identification of parameters in models of robots with rotary joints”. In: *IEEE International Conference on Robotics and Automation* 2, pp. 1045–1049.

Omid, M., M. Khojastehnazhand, and A. Tabatabaeefar (2010). “Estimating volume and mass of citrus fruits by image processing technique”. In: *Journal of Food Engineering* 100.2, pp. 315–321.

Pagano, C C et al. (1994). “Role of the inertia tensor in haptically perceiving where an object is grasped”. In: *Journal of Experimental Psychology. Human Perception and Performance* 20.2, pp. 276–285.

Pagano, Christopher C and M T Turvey (1992). “Eigenvectors of the inertia tensor and perceiving the orientation of a hand-held object by dynamic touch”. In: *Perception & Psychophysics* 52.6, pp. 617–24.

Papadopoulos, Evangelos, Ioannis Tortopidis, and Kostas Nanos (2005). “Smooth planning for free-floating space robots using polynomials”. In: *IEEE International Conference on Robotics and Automation*, pp. 4272–4277.

Pesce, Vincenzo, Michèle Lavagna, and Riccardo Bevilacqua (2017). “Stereovision-based pose and inertia estimation of unknown and uncooperative space objects”. In: *Advances in Space Research*, pp. 236–251.

Petersen, Joshua G and Ferdinando Rodriguez y Baena (2014). “Mass and inertia optimization for natural motion in hands-on robotic surgery”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4284–4289.

Petersen, Joshua G., Stuart A. Bowyer, and Ferdinando Rodriguez Y Baena (2016). “Mass and Friction Optimization for Natural Motion in Hands-On Robotic Surgery”. In: *IEEE Transactions on Robotics* 32.1, pp. 201–213.

Petkos, Georgios and Sethu Vijayakumar (2007). “Load estimation and control using learned dynamics models”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1527–1532.

Pick Jr, Herbert L and Anne D Pick (1967). “A developmental and analytic study of the size-weight illusion”. In: *Journal of Experimental Child Psychology* 5 (3), pp. 362–371.

Plaisier, Myrthe A and Jeroen BJ Smeets (2012). “Mass is all that matters in the size–weight illusion”. In: *PLoS One* 7.8, e42518.

Pokorny, Florian T et al. (2017). “A database for reproducible manipulation research: CapriDB–Capture, Print, Innovate”. In: *Data in Brief* 11, pp. 491–498.

Ragaglia, Matteo et al. (2014). “Integration of perception, control and injury knowledge for safe human-robot interaction”. In: *IEEE International Conference on Robotics and Automation*, pp. 1196–1202.

Ramos, Francisco et al. (2013). *Time-optimal online trajectory generator for robotic manipulators*. Eidgenössische Technische Hochschule Zürich, Institute for Dynamic Systems and Control.

Reed, Benjamin B et al. (2016). “The Restore-L Servicing Mission”. In: *AIAA SPACE 2016*, p. 5478.

Ren, Yi et al. (2016). “Biomimetic object impedance control for dual-arm cooperative 7-DOF manipulators”. In: *Robotics and Autonomous Systems* 75, pp. 273–287.

Ross, Helen E and Millard F Reschke (1982). “Mass estimation and discrimination during brief periods of zero gravity”. In: *Perception & Psychophysics* 31.5, pp. 429–436.

Rossi, Roberto et al. (2015). "A pre-collision control strategy for human-robot interaction based on dissipated energy in potential inelastic impacts". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 26–31.

Russakow, Jeffrey, Oussama Khatib, and Stephen M Rock (1995). "Extended operational space formulation for serial-to-parallel chain (branching) manipulators". In: *IEEE International Conference on Robotics and Automation*. Vol. 1, pp. 1056–1061.

Saxena, Ashutosh, Justin Driemeyer, and Andrew Y Ng (2008). "Robotic grasping of novel objects using vision". In: *The International Journal of Robotics Research* 27.2, pp. 157–173.

Schedlinski, C and M Link (2001). "A survey of current inertia parameter identification methods". In: *Mechanical Systems and Signal Processing* 15.1, pp. 189–211.

Schmidtler, Jonas and Moritz Körber (2018). "Human perception of inertial mass for joint human-robot object manipulation". In: *ACM Transactions on Applied Perception* 15.3, p. 15.

Schneider, Stanley A and Robert H Cannon (1992). "Object impedance control for cooperative manipulation: Theory and experimental results". In: *IEEE Transactions on Robotics and Automation* 8, pp. 383–394.

Scholz, Jonathan et al. (2014). "A physics-based model prior for object-oriented MDPs". In: *International Conference on Machine Learning* 32, pp. 1089–1097.

Scholz, Jonathan et al. (2015). "Learning non-holonomic object models for mobile manipulation". In: *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 5531–5536.

Shi, J. et al. (2017). "Dynamic In-Hand Sliding Manipulation". In: *IEEE Transactions on Robotics* 33, pp. 778–795. ISSN: 1552-3098.

Smith, Christian et al. (2012). "Dual arm manipulation–A survey". In: *Robotics and Autonomous Systems* 60, pp. 1340–1353.

Standley, Trevor et al. (2017). "image2mass: Estimating the mass of an object from its image". In: *Annual Conference on Robot Learning* 78, pp. 324–333.

Swevers, J. et al. (2002). "An Experimental Robot Load Identification Method for Industrial Application". In: *The International Journal of Robotics Research* 21, pp. 701–712.

Swevers, Jan, Walter Verdonck, and Joris De Schutter (2007). "Dynamic model identification for industrial robots". In: *IEEE Control Systems* 27.5, pp. 58–71.

Tanaka, Hiromi T et al. (2003). "A vision-based haptic exploration". In: *IEEE International Conference on Robotics and Automation* 3, pp. 3441–3448.

Tanaka, S et al. (2004). "Active mass estimation with haptic vision". In: *International Conference on Pattern Recognition* 3, pp. 256–261.

Tiest, Wouter M Bergmann and Astrid ML Kappers (2010). "Haptic perception of gravitational and inertial mass". In: *Attention, Perception, & Psychophysics* 72.4, pp. 1144–1154.

Tscharke, M and T M Banhazi (2013). "Review of Methods to Determine Weight and Size of Livestock from Images". In: *Australian Journal of Multi-Disciplinary Engineering* 10.1, pp. 1–17.

Umetani, Yoji and Kazuya Yoshida (1989). "Resolved motion rate control of space manipulators with generalized Jacobian matrix". In: *IEEE Transactions on robotics and automation*, pp. 303–314.

Viña, Francisco et al. (2015). "In-hand manipulation using gravity and controlled slip". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5636–5641.

Viña, Francisco et al. (2016). "Adaptive control for pivoting with visual and tactile feedback". In: *IEEE International Conference on Robotics and Automation*, pp. 399–406.

Vivek Venkatesh, G. et al. (2015). "Estimation of volume and mass of axi-symmetric fruits using image processing Technique". In: *International Journal of Food Properties* 18.3, pp. 608–626.

Walker, Ian D. (1990). "The use of kinematic redundancy in reducing impact and contact effects in manipulation". In: *IEEE International Conference on Robotics and Automation*, pp. 434–439.

— (1994). "Impact Configurations and Measures for Kinematically Redundant and Multiple Armed Robot Systems". In: *IEEE Transactions on Robotics and Automation* 10.5, pp. 670–683.

Walker, Ian D, Robert A Freeman, and Steven I Marcus (1991). "Analysis of motion and internal loading of objects grasped by multiple cooperating manipulators". In: *The International Journal of Robotics Research* 10, pp. 396–409.

Wang, Mingming, Jianjun Luo, and Ulrich Walter (2015). "Trajectory planning of free-floating space robot using Particle Swarm Optimization (PSO)". In: *Acta Astronautica*, pp. 77–88.

Wang, Mingming et al. (2018). "Coordinated trajectory planning of dual-arm space robot using constrained particle swarm optimization". In: *Acta Astronautica* 146, pp. 259–272.

Wilkening, Paul et al. (2017). "Development and experimental evaluation of concurrent control of a robotic arm and continuum manipulator for osteolytic lesion treatment". In: *IEEE Robotics and Automation Letters* 2.3, pp. 1625–1631.

Wohlkinger, Walter et al. (2012). "3Dnet: Large-scale object class recognition from cad models". In: *IEEE International Conference on Robotics and Automation*, pp. 5384–5391.

Wu, Jiajun et al. (2015). "Galileo : Perceiving physical object properties by integrating a physics engine with deep learning". In: *Advances in Neural Information Processing Systems*, pp. 1–9.

Wu, Jiajun et al. (2016). "Physics 101: Learning physical object properties from unlabeled videos". In: *British Machine Vision Conference 2016*, pp. 1–12.

Wu, Jun, Jinsong Wang, and Zheng You (2010). "An overview of dynamic parameter identification of robots". In: *Robotics and Computer-Integrated Manufacturing* 26.5, pp. 414–419.

Xu, Wenfu et al. (2008). "The Cartesian path planning of free-floating space robot using particle swarm optimization". In: *International Journal of Advanced Robotic Systems*, p. 27.

Yang, Yan and Guanghui Teng (2007). "Estimating pig weight from 2D images". In: *International Conference on Computer and Computing Technologies in Agriculture*, pp. 1471–1474.

Yoshikawa, T. and M. Kurisu (1991). "Indentification of the center of friction from pushing an object by a mobile robot". In: *IEEE/RSJ International Workshop on Intelligent Robots and Systems* 2, pp. 449–454.

Yoshikawa, Tsuneo and Xin-Zhi Zheng (1993). “Coordinated dynamic hybrid position/force control for multiple robot manipulators handling one constrained object”. In: *The International Journal of Robotics Research* 12.3, pp. 219–230.

Yu, Kuan Ting, John Leonard, and Alberto Rodriguez (2015). “Shape and pose recovery from planar pushing”. In: *IEEE International Conference on Intelligent Robots and Systems*. Vol. 2015-Decem, pp. 1208–1215.

Yu, Kuan Ting et al. (2016). “More than a million ways to be pushed. A high-fidelity experimental dataset of planar pushing”. In: *IEEE International Conference on Intelligent Robots and Systems*, pp. 30–37.

Yu, Yong, Tetsu Arima, and Showzow Tsujio (2005). “Estimation of object inertia parameters on robot pushing operation”. In: *IEEE International Conference on Robotics and Automation*, pp. 1657–1662.

Yu, Yong, Takashi Kiyokawa, and Showzow Tsujio (2004). “Estimation of mass and center of mass of unknown and graspless cylinder-like object”. In: *International Journal of Information Acquisition* 1, pp. 47–55.

Zhang, Q. and H. Li (2007). “MOEA/D: A multiobjective evolutionary algorithm based on decomposition”. In: *IEEE Transactions on Evolutionary Computation*, pp. 712–731.

Zhu, Shaojun and Abdeslam Boularias (2016). “A Physically-grounded and Data-efficient Approach to Motion Prediction using Black-box Optimization”. In: *Workshop on Intuitive Physics in the Annual Conference on Neural Information Processing Systems*.

Zhu, Shaojun, Andrew Kimmel, and Abdeslam Boularias (2017). “Information-theoretic model identification and policy search using physics engines with application to robotic manipulation”. In: *arXiv:1703.07822 (2017)*.