

Reliable robotic grasping for uncertain objects through virtual model control

Auteur : Vanderheyden, Julien

Promoteur(s) : Drion, Guillaume; Sacré, Pierre

Faculté : Faculté des Sciences appliquées

Diplôme : Master : ingénieur civil électricien, à finalité spécialisée en Neuromorphic Engineering

Année académique : 2024-2025

URI/URL : <http://hdl.handle.net/2268.2/23305>

Avertissement à l'attention des usagers :

Tous les documents placés en accès ouvert sur le site le site MatheO sont protégés par le droit d'auteur. Conformément aux principes énoncés par la "Budapest Open Access Initiative"(BOAI, 2002), l'utilisateur du site peut lire, télécharger, copier, transmettre, imprimer, chercher ou faire un lien vers le texte intégral de ces documents, les disséquer pour les indexer, s'en servir de données pour un logiciel, ou s'en servir à toute autre fin légale (ou prévue par la réglementation relative au droit d'auteur). Toute utilisation du document à des fins commerciales est strictement interdite.

Par ailleurs, l'utilisateur s'engage à respecter les droits moraux de l'auteur, principalement le droit à l'intégrité de l'oeuvre et le droit de paternité et ce dans toute utilisation que l'utilisateur entreprend. Ainsi, à titre d'exemple, lorsqu'il reproduira un document par extrait ou dans son intégralité, l'utilisateur citera de manière complète les sources telles que mentionnées ci-dessus. Toute utilisation non explicitement autorisée ci-avant (telle que par exemple, la modification du document ou son résumé) nécessite l'autorisation préalable et expresse des auteurs ou de leurs ayants droit.

Reliable robotic grasping for uncertain objects through virtual model control

a thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineering

by JULIEN VANDERHEYDEN

Supervisors: GUILLAUME DRION, PIERRE SACRÉ
Academic year 2024–2025

Jury members

MARC VAN DROOGENBROECK, Professor at the University of Liège (President);

PIERRE SACRÉ, Professor at the University of Liège (Promotor) ;

GUILLAUME DRION, Professor at the University of Liège (Co-promotor);

OLIVIER BRÜLS, Professor at the University of Liège;

FULVIO FORNI, Professor at the University of Cambridge.

Declaration on the use of automatic tools for writing the manuscript

I hereby certify that I have not used any generative intelligence tool in the writing of text, graphics, images, or data reproduced in this manuscript.

The writing tools used in this document are the following:

- Reverso: All along the project.
- DeepL: All along the project.
- Synonym Dictionary: All along the project.
- Grammarly: All along the project.

Acknowledgments

First, I would like to express my sincere gratitude to my supervisors, Professors P. Sacré and G. Drion, for giving me the opportunity to work on such a fascinating topic, for their continuous support throughout the entire duration of this project, and for enabling me to undertake a research stay at the University of Cambridge.

I would also like to thank all the members of the Control Lab at the University of Cambridge for their warm welcome and for the many insightful discussions I had with several of them. I am especially grateful to Professor F. Forni for his invaluable guidance on the direction of this project, and to Daniel Larby for patiently answering my endless questions and for developing such an elegant software package.

My thanks also go to the Neuroengineering Lab at the University of Liège for providing access to a cutting-edge piece of hardware, unique in Belgium. I am particularly grateful to Antoine Debor for his repeated help in mastering the robot and for his surgical expertise.

A heartfelt thank you to Shadow Robot Company for designing such a reliable robotic hand and for their prompt and helpful support throughout the project.

I would also like to thank Alice and Robin for their help with the various pictures in this document, and for their patience and support in listening to daily updates about robotic hands throughout an entire semester.

Finally, I would like to thank all members of the jury for dedicating their time and attention to reading this report.

Summary

Dexterous robotic grasping remains a central challenge in robotics due to the high-dimensional nature of multi-fingered manipulators and the complex physical interactions involved in object manipulation. This thesis investigates the application of the Virtual Mechanisms (VM) control framework to enhance grasping capabilities in anthropomorphic robotic hands. Building on the principles of passivity-based impedance control, the VM approach introduces virtual mechanical elements such as springs, dampers, and inertial components interconnected in operational space to generate intuitive and modular control behaviors. The thesis proposes two main application approaches. The first one utilizes virtual mechanisms as simple and intuitive hand-centered trajectory planners to generate human-inspired grasping motions, employing a taxonomy-based approach to implement the following common grasp types: medium wrap, power sphere, and lateral pinch. The second one extends this approach toward object-centric behavior shaping, wherein virtual mechanisms are dynamically tailored to object geometry. Both strategies are implemented and tested on a Shadow Dexterous Hand robotic platform. Results indicate that virtual mechanisms offer a robust and versatile control paradigm, enhancing grasp robustness and adaptability while preserving stability. The thesis concludes with a proof-of-concept integration of position-based feedback into the VM framework, laying the groundwork for future developments. Overall, this work highlights the potential of virtual mechanisms as a promising alternative to traditional grasp control strategies in dexterous robotic manipulation.

Contents

Introduction	1
I Background	3
1 State-of-the-art dexterous hand grasping	4
1.1 Introduction	4
1.1.1 Background and terminology	4
1.1.2 Chapter overview	5
1.2 Grasp synthesis	6
1.2.1 Form and force closure	6
1.2.2 Grasp synthesis: analytical approach	7
1.2.3 Grasp synthesis: empirical approach	8
1.3 Control methods	9
1.3.1 Hybrid position/force control	10
1.3.2 Impedance control	10
1.4 Synergies	11
1.4.1 The concept of synergies	11
1.4.2 Synergies applications	12
1.4.3 Underactuated hands	14
1.5 Learning-based methods	15
2 Introduction to the virtual mechanisms control framework	16
2.1 General idea	16
2.2 Virtual mechanisms fundamentals	17
2.2.1 Virtual forces	17
2.2.2 Virtual elements	17
2.2.3 Virtual structures and implementation	19
2.3 Design examples and applications	19
2.3.1 Simple examples	20
2.3.2 Advanced architectures	21
2.3.3 Virtual mechanisms and dexterous manipulation	21
2.4 Optimal parameter tuning	22
2.4.1 Problem statement	22
2.4.2 Analytical approaches	23
2.4.3 Algorithmic differentiation	23
3 Control architecture	25
3.1 The Shadow Dexterous Hand	25
3.2 The VMRobotControl control package	26

TABLE OF CONTENTS

II	Applications	29
4	Virtual mechanisms as hand-centric trajectory planners	30
4.1	Grasp taxonomies	30
4.1.1	Presentation of the different existing taxonomies	30
4.1.2	Choice of the implemented grasps	32
4.2	Virtual mechanisms for grasp motions	33
4.2.1	Medium wrap	34
4.2.2	Power sphere	36
4.2.3	Lateral pinch	37
4.3	Experimental results and discussion	38
4.3.1	Practical control architecture	38
4.3.2	Experimental setup and performances evaluation	39
4.3.3	Results	42
4.3.4	Discussion	48
5	Virtual mechanisms as object-centric behavior shapers	50
5.1	General idea	50
5.2	Virtual mechanism architectures	51
5.2.1	Objects models	51
5.2.2	Medium wrap	53
5.2.3	Power sphere	58
5.2.4	Lateral pinch	59
5.3	Results and discussion	61
5.3.1	Simulation results: parameter analysis	61
5.3.2	Experimental results	67
5.3.3	Discussion and perspectives	71
6	Feedback integration	73
6.1	General idea	73
6.2	Mismatch evaluation	73
6.3	Implementation and discussion	76
III	Conclusion	78
7	Conclusion and perspectives	79
7.1	Conclusion	79
7.2	Perspectives	80
IV	Appendices	81
A	Human taxonomies	82
A.1	Cutkosky's taxonomy	82
A.2	Feix's taxonomy	83
B	Steady state grasping configurations	84
B.1	Trajectory planners	84
B.2	Object-centric grasping	86
C	Final parameters values	88

TABLE OF CONTENTS

D Additional Materials	90
Bibliography	95

Introduction

Motivation

In recent decades, the field of robotics has undergone a noteworthy transformation. Robots are increasingly present in our daily lives, while the industrial sector is shifting toward greater flexibility, enabling product customization and multi-product assembly lines. Although both domains benefit from advances in mobile robots, robotic arms, computer vision, and other technologies, one crucial component remains underdeveloped: endowing robots with advanced manipulation capabilities [1, 2, 3]. In industry, this limitation is typically addressed by equipping robots with multiple task-specific end-effectors, often requiring time-consuming tool changes and resulting in a limited range of achievable tasks [2]. In domestic settings, the lack of dexterous manipulation is still a major barrier to the integration of more capable robots into everyday life.

Various designs have been proposed to achieve more universal manipulation tools. However, the vast majority of objects in our environment have been designed to be grasped and manipulated by human hands. As a result, mimicking the human end-effector through anthropomorphic hand designs stands out as a particularly promising approach. In addition to their functional relevance, human-like robotic hands have applications in prosthetics and humanoid robotics, where familiarity of form also serves an aesthetic purpose. Following this reasoning, numerous anthropomorphic robotic hand designs have been developed over the years, ranging from highly simplified to highly biomimetic models. Some examples are illustrated in Fig. 1.

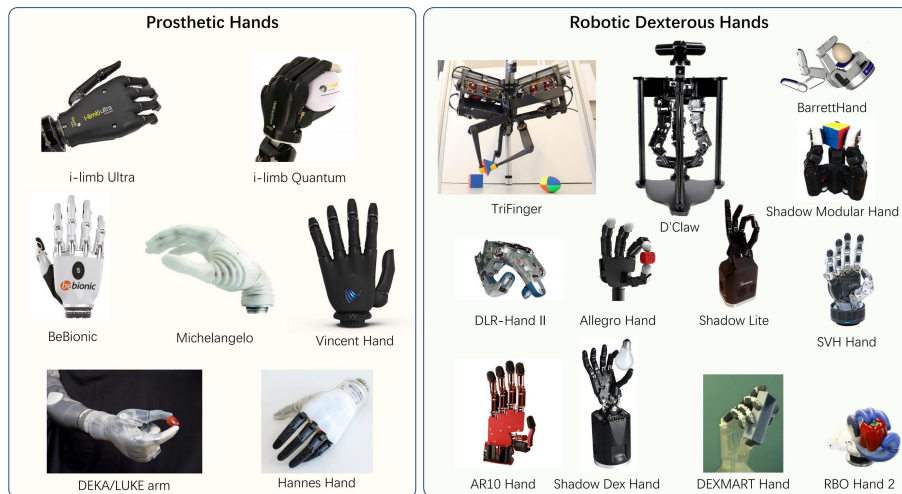


Figure 1: Examples of robotic hand designs [4]

While mechanical and electrical engineers continue to improve hardware design, control engineers face the challenge of enabling these devices to autonomously perform a wide range of tasks. Among the numerous tasks that a robotic hand should be able to accomplish, **grasping** is widely recognized as a stepping stone [2]. Indeed, grasping is a prerequisite for subsequent object manipulation and meaningful interaction with the physical world. Despite its apparent simplicity, grasping is a complex problem that requires the integration of

INTRODUCTION

multiple subsystems: tactile sensing, motion planning, visual perception, object and contact modeling, and the control of a highly redundant, multi-degree-of-freedom device.

Over the years, numerous control strategies have been proposed to address this challenge, each with varying levels of success. The principal approaches include analytical and empirical grasp synthesis [5], combined with hybrid position/force control [6] or impedance control [7], the use of synergies [8, 9], and more recently, learning-based methods [10, 1, 4]. Nonetheless, the robotics community has yet to converge on a universal, robust, and reliable control method for grasping [3]. Recently, the Control Laboratory at the University of Cambridge introduced a control framework known as **virtual mechanisms** [11], which integrates several key principles that have previously been employed in the design of grasping controllers. This framework has already shown promising results in applications ranging from surgical robotics [12] to daily-life robotics [13].

In this context, the objective of this thesis is to investigate the potential of the virtual mechanisms control framework in the domain of dexterous robotic grasping, and to evaluate the performance through experiments conducted on the anthropomorphic robotic hand available at the University of Liège.

Structure

This thesis is organized into four main parts, comprising a total of seven chapters:

Part I - Background lays the theoretical and conceptual foundation of the work.

Chapter 1 reviews the relevant literature and presents the different approaches proposed by the scientific community to achieve dexterous grasping. This provides both a comprehensive understanding of the current **state of the art** and an introduction to fundamental principles that will underpin the rest of the work.

Chapter 2 is dedicated to the **virtual mechanisms control framework**, introducing its core concept, theoretical foundations, and practical illustrations. This chapter equips the reader with the necessary background and intuition to understand the contributions in the subsequent chapters.

Chapter 3 introduces the practical **system architecture** from both software and hardware perspectives.

Part II - Applications focuses on implementations and experimental validation of grasping policy, using all the notions introduced in Part I.

Chapter 4 presents a first, straightforward application of virtual mechanisms, namely their use as **hand-centered trajectory planners**. It describes the different grasping taxonomies considered and details the experimental setup along with the performance evaluation criteria.

In **Chapter 5**, an original grasping strategy referred to as **object-centric behavior shaping** is proposed. Building upon the previous chapter, it reuses the same taxonomy, control architecture, and experimental setup. With those elements already established, the focus shifts to the description of the complex controller architecture and the practical evaluation of its performance.

Chapter 6 explores the integration of **position-based feedback** into the grasping strategies, acting on the discrepancy between the commanded and the actual position of the manipulator. This offers a proof-of-concept and opens the door to future investigations.

Part III – Conclusion concludes the thesis.

Chapter 7 summarizes the main contributions of the work and discusses several promising directions for future improvements and research.

Part IV – Appendices provides supplementary material that complements the main text. It includes a detailed presentation of the grasping taxonomies, photographs of the final grasp configurations, tables listing the final values of the parameters, and links to both the GitHub repository and a YouTube demonstration video.

Part I

Background

Chapter 1

State-of-the-art dexterous hand grasping

Since the development of the first robotic hands [14], the scientific community has devoted significant effort to addressing the problem of grasping. This section provides a concise overview of the diverse approaches that have been explored, ranging from the earliest analytical models of grasping to the most recent learning-based techniques. To facilitate the reader's understanding, this chapter begins with an introductory section that clarifies terminology and outlines the overall structure of the content presented.

1.1 Introduction

1.1.1 Background and terminology

In [15], a grasp is defined as "every static hand posture with which an object can be held securely with one hand, irrespective of the hand orientation." In the next pages, the word "grasp" will thus refer to the **hand configuration** (rather than the whole grasping motion) used by either the human or the robot to hold the considered object. With such a definition, the task of grasping may appear broad and relatively straightforward. However, given this generic definition, one might naturally ask what constitutes a good grasp.

A first and very intuitive property expected from a grasp is that it should be **stable**. Extensive work has been conducted in the literature to mathematically define what constitutes a stable grasp [16, 17], leading to several important concepts such as force closure and form closure, which will be discussed later in this chapter. A more general perspective on grasp stability is provided in [5], where it is stated that "a grasp is stable if a small disturbance, either in the object position or in the finger forces, generates a restoring wrench that tends to return the system to its original configuration."

Although stability is a crucial aspect, grasping is rarely the final goal: an object is typically grasped in order to perform a specific task, and the way it is grasped often depends on the nature of that task. For example, the hand configuration used to pick up a wine glass will differ depending on whether the goal is to drink from it, clean it, hand it to someone, and so on. Therefore, a second essential property for a grasp to be considered "good" is its **task specificity**. As with stability, several attempts have been made to mathematically quantify how well a grasp aligns with the requirements of a given task [18, 19, 20]. Despite its importance, this aspect of grasping will be given less emphasis in this project, in favor of the next and final grasp property.

Finally, while maintaining stability and task specificity, human beings typically do not adopt a unique grasp for every possible combination of object and task. As will be discussed later in this work (see Sec. 4.1), they instead tend to select grasps from a limited set of common patterns, often referred to as a taxonomy. Once again, the scientific community has devoted considerable effort to classifying these grasp types [15, 21, 22], and these taxonomies will be reviewed in the corresponding section. Based on this observation, it is expected that a grasp should generalize across a wide variety of objects, including those the robotic system has never encountered before. Therefore, the final desirable property of a "good" grasp is its **robustness**.

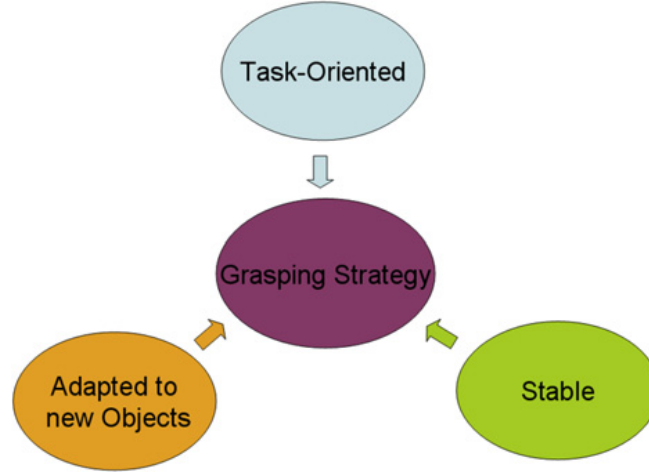


Figure 1.1: Desired grasp properties [5]

The three aforementioned properties are summarized in Fig. 1.1. In this project, the primary focus will be on maximizing grasp robustness while ensuring a satisfactory level of stability. Although task specificity is an important consideration, it will receive less attention here, as the scope of the project is limited to the act of grasping itself and does not (yet) encompass the execution of tasks with the grasped object, beyond potentially relocating it to another position.

1.1.2 Chapter overview

Now that the grasping problem has been introduced in general terms, this chapter proceeds with an exploration of the main strategies that have been proposed to address it. The standard approach is typically divided into three stages, as illustrated in Fig. 1.2.

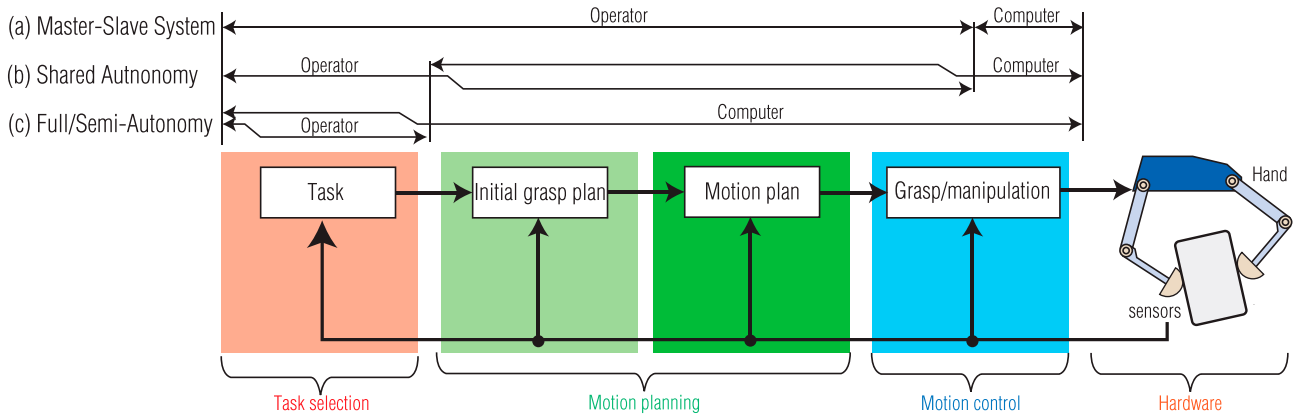


Figure 1.2: The typical grasping pipeline: grasp synthesis, motion planning, and motion control [23].

The first stage, known as **Grasp Synthesis** (covered in Section 1.2), involves determining a grasp configuration that satisfies (or ideally optimizes) one or more of the grasp quality criteria introduced earlier. The second stage, **Motion Planning**, computes the trajectory of the fingers required to achieve the synthesized grasp. While motion planning is a critical component in the broader field of manipulation, in the specific context of grasping it is often addressed using relatively simple kinematic computations. For a comprehensive review of motion planning algorithms, the reader is referred to [23]. The final stage pertains to **Control Methods**, which are responsible for executing the planned trajectory while compensating for external disturbances and ensuring the stability of the grasp. This topic will be covered in Section 1.3. More recently, insights from neuroscience have introduced the key concept of **synergies**, which can be applied at various stages of the grasping pipeline.

This concept, along with its practical implications, is discussed in Section 1.4. Finally, advances in artificial intelligence have given rise to new **learning-based approaches** that deviate from this traditional pipeline. These alternative methods will be briefly introduced in Section 1.5.

1.2 Grasp synthesis

1.2.1 Form and force closure

A critical aspect that distinguishes grasping from other standard robotics problems is that, in addition to controlling the motion of the robotic body, it is also necessary to manage the internal forces generated within the object. These forces play a crucial role in ensuring the stability of the grasp. It is relatively intuitive that not all hand configurations around an object allow for effective control of internal forces in all directions. This observation has led to the classification of grasps as either "force-closed" or not. A grasp is said to be force-closed when the fingers are capable of applying suitable forces on the object to generate wrenches in any direction [24]. Force-closed grasps have the desirable property of being stable, although not all stable grasps are necessarily force-closed. It is important to note that force closure ensures that a grasp can resist arbitrary external wrenches under suitable control, but actual stability still depends on the dynamic response of the system and the control strategy used.

To determine whether a given grasp is force-closed, one must first understand the types of forces that each finger can apply to the object. For this purpose, several **finger models** were introduced in [25], based on the nature of the forces that the finger is capable of producing:

- **Frictionless point contact:** This model accounts only for the normal force exerted by the finger on the object.
- **Frictional point contact:** In addition to the normal force, this model includes a tangential frictional force.
- **Soft finger:** Along with the normal and tangential forces, this model also considers a torsional moment around the normal axis.

These different models are illustrated in Fig. 1.3

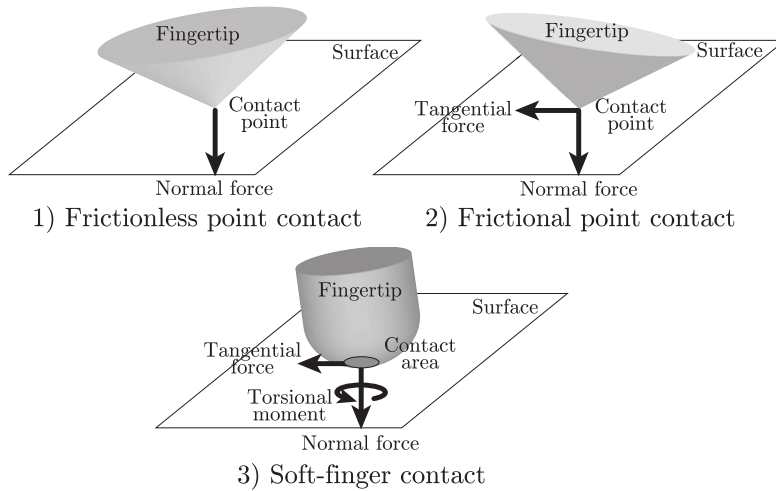


Figure 1.3: The three fingertip contact models: progressive addition of forces and torques at the contact interface [23].

In a manner similar to force closure, another classification known as **form closure** has been introduced. Form closure is a stronger condition than force closure, as it is purely geometrical: a grasp is said to achieve form closure if and only if it satisfies the conditions of force closure using frictionless point contacts. These two concepts are illustrated in Fig. 1.4, where it can be seen in (b) that any force along the vertical axis can be counteracted

by frictional forces, provided that the force applied through the prismatic joint on the right is sufficiently large. Although force and form closure are not necessary conditions for a grasp to be stable, analytical procedures have been developed to determine whether a given grasp satisfies force or form closure [25], thereby facilitating the identification of a subset of stable grasps from the set of all possible grasps for a given object.

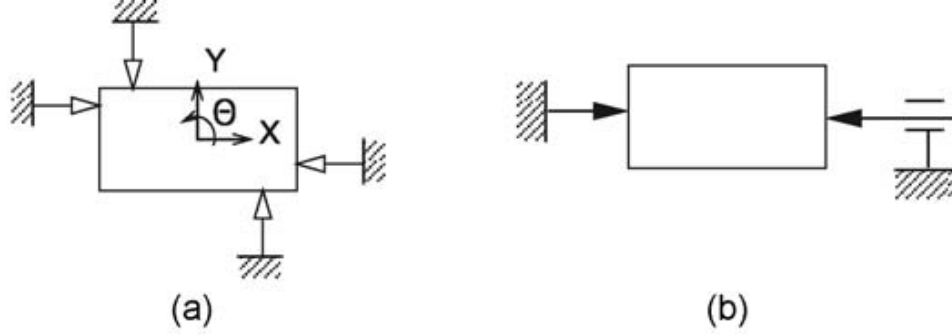


Figure 1.4: (a) Form closure, the object is geometrically constrained. (b) Force closure, the friction forces constrain the object [26].

1.2.2 Grasp synthesis: analytical approach

A wide range of algorithms have been proposed for grasp synthesis (see [5] for a comprehensive review, and [3] for a more recent meta-survey encompassing numerous surveys on the topic). Listing all of them is beyond the scope of this project. Instead, the objective here is to highlight the main research directions in order to provide a clearer understanding of the current state of the field.

Grasp generation methods can generally be categorized into two main approaches: analytical and empirical. The oldest approach is the analytical one, which aims to quantify as rigorously as possible the different properties of a grasp discussed in Sec. 1.1.1, and then use these metrics to identify a "good" grasp. A generic schematic representation of the analytical approach is illustrated in Fig. 1.5. Numerous variations of this method have been proposed, differing in their focus (e.g., force-closure, task compatibility), the object model they assume (e.g., 2D or 3D, polyhedral or not), the fingertip model, and the quality criteria employed. A particularly interesting subset of these algorithms concerns the search for **optimal grasps**: what initially appears to be an intractable problem can often be discretized, allowing the application of heuristics or gradient-based optimization techniques to identify the grasp that maximizes a given objective function.

A crucial aspect of analytical grasp synthesis methods is the choice of the quality criterion. An extensive overview of existing metrics can be found in [27]. These criteria can broadly be categorized into two groups: those that assess the **stability** of the grasp, and those that evaluate its **task compatibility**.

A widely used and well-known quality measure for evaluating grasp stability is the **Largest-minimum resisted wrench**. This metric accounts for limitations in fingertip forces and computes the largest perturbation wrench that the grasp can resist in any direction. Although the computation of this measure may appear complex, efficient algorithms have been proposed to evaluate it [28]. Simpler alternatives include methods based on properties of the grasp matrix G , which maps fingertip forces to the resultant wrench on the object, such as its singular values [29]. Other approaches rely on geometric considerations, such as the alignment between the hand principal axis and the object main axis of inertia [30]. Beyond the contact points themselves, some quality measures also incorporate the overall hand configuration, taking into account factors such as proximity to kinematic singularities.

Evaluating task-oriented performance is more challenging. The general idea is to model the task as an ellipsoid in the 6-dimensional wrench space (three force components and three torque components), where the ellipsoid is elongated in the directions that are most critical for the task. A task compatibility index can then be computed by

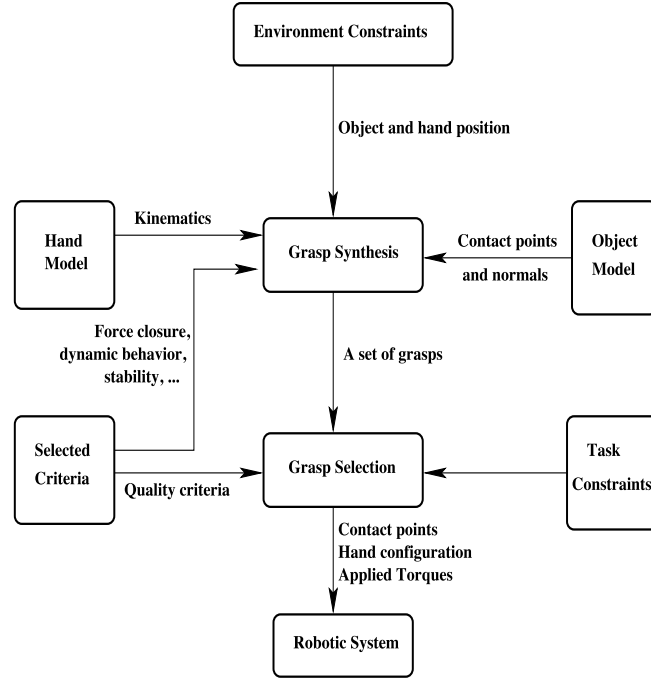


Figure 1.5: Analytical approach for grasp synthesis: quality criteria are used to select a grasp [5].

comparing this task wrench ellipsoid to the grasp wrench ellipsoid, which characterizes the directions in which the grasp can exert the largest wrenches. See [6] for a very practical example of such a quality measure.

Finally, all these different metrics can of course be combined to yield more comprehensive and general-purpose criteria. An important observation that emphasizes the value of using multiple metrics is that a grasp considered optimal according to one criterion may turn out to be highly suboptimal with respect to another. This point is clearly illustrated in [27] through a simple example involving a 2D rectangular object.

Although analytical approaches can yield promising results in certain scenarios, such as pick-and-place tasks, they still present several limitations. First, the search space they need to explore is extremely large, often resulting in substantial computational costs. Moreover, task-oriented grasping requires an accurate model of the task, which is often difficult to obtain. Additionally, these methods rely on a precise model of the object to be grasped, which may not always be available. This specific limitation will be addressed in the following chapters of this project.

1.2.3 Grasp synthesis: empirical approach

The high computational demands and the challenges associated with task modeling, combined with the limited amount of scientific work on task-oriented analytical grasp synthesis, significantly weaken the effectiveness of analytical methods when dealing with task-specific grasping. This observation has led to the emergence of a new paradigm: the empirical approach (also referred to as the data-driven approach). The underlying idea is straightforward: since mathematical models and equations often fall short in producing task-appropriate grasps, robotic systems should instead take inspiration from human behavior. The strategy, therefore, is to collect data from humans performing grasping on real objects using appropriate sensors, storing them and using them to derive a policy, as illustrated in Fig. 1.6.

As analytical approaches can be divided into stability-focused and task-focused criteria, empirical methods can similarly be subdivided into **hand-centered** and **object-centered** approaches. In the hand-centered approach, the emphasis is placed on the motion and configuration of the hand: data is acquired using magnetic data gloves or vision-based sensors, and the evolution of the hand joints is recorded. Learning-based methods are then applied to this data, for instance to classify different grasps into a taxonomy and to learn the parameters

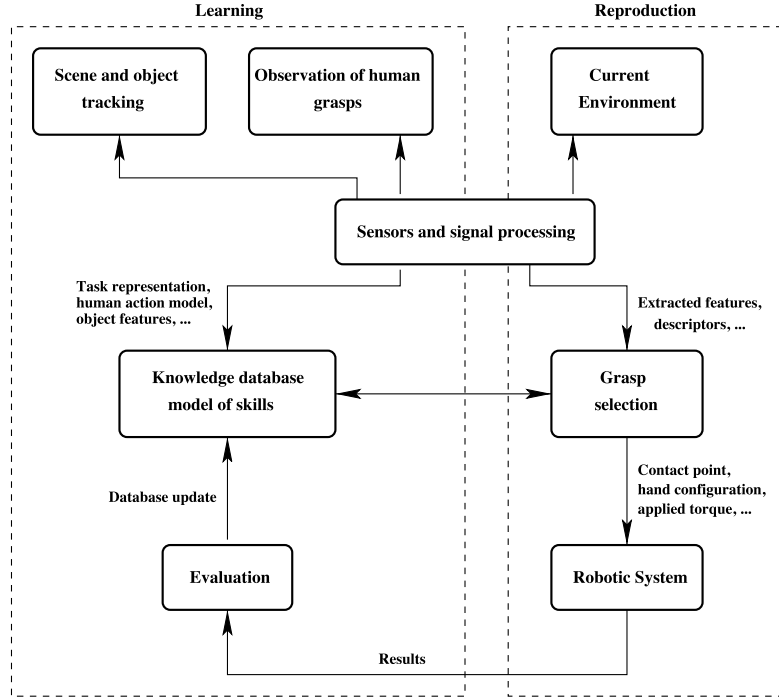


Figure 1.6: Empirical grasp synthesis approach. The policy is learned through data acquisition [5].

associated with each grasp [31].

The object-centered approach differs in the type of data being recorded: in this paradigm, the shape of the grasped object is stored, along with the corresponding grasp points. Based on this data, models can be trained to learn a relationship between object shape and the grasp used to manipulate it, or to identify suitable grasping regions on the object. This hand-centered/object-centered distinction is a recurring topic within the grasping community and will be further discussed later in this project, as inspiration will be drawn from both approaches.

These empirical approaches help to overcome the difficulty of task modeling and generally perform better than analytical methods when it comes to task-oriented grasping. However, while analytical methods can guarantee the generation of a "good" and stable grasp for any given object (provided an accurate model is available), empirical methods lack such guarantees. In particular, there is no assurance that a model trained on a given dataset will generalize well to unseen objects, which remains one of the main limitations of current data-driven approaches.

1.3 Control methods

Once the grasp has been determined, appropriate control methods must be employed to execute it and respond to potential disturbances that may occur during manipulation. As discussed in Sec. 1.2.1, relying solely on position control is insufficient for achieving a stable and effective grasp. Instead, the control strategy should also take into account the forces applied by the fingertips, as well as the internal forces generated within the object. As outlined in [26], control solutions addressing this combined force/position problem can be broadly categorized into two main approaches: hybrid position/force control and impedance control. For both of these approaches, numerous control strategies have been proposed, which primarily differ in the assumptions made regarding the finger contact model. A comprehensive chronological review of these methods can be found in [23].

1.3.1 Hybrid position/force control

The fundamental principle behind hybrid position/force control [6] is to control the position along certain directions while controlling the force along others, as illustrated in Fig. 1.7. This approach remains relevant even in the absence of contact between the grasped object and other elements in the environment, since internal forces within the object must still be appropriately managed.

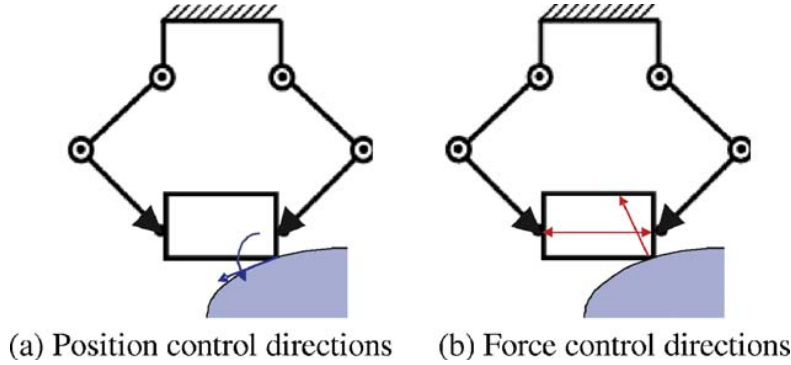


Figure 1.7: Hybrid position/force control [26]

In practice, this control strategy is typically implemented using a two-stage procedure. First, the dynamic equations governing the motion of the hand and the object are decoupled to enable simultaneous position and force control along the directions of interest. Once this decoupling is achieved, conventional control laws such as PID controllers can be applied to regulate both variables independently.

A major limitation of this framework lies in its requirement for accurate knowledge of the dynamic parameters of both the hand and the object. While it is reasonable to assume that the hand parameters are known, assuming full knowledge of the object dynamics notably restricts the generality of the approach. To address this limitation, [32] proposed a strategy in which the robot learns the dynamic parameters of both the object and itself during the manipulation process. This approach is referred to as adaptive control.

Another challenge that remains unsolved stems from the inherent nature of the hybrid controller. When the grasped object comes into contact with external elements in the environment, the directions of force and position control must be redefined. This transition typically introduces delays and may cause overshoots in the positional response due to the time required for the controller to adapt.

1.3.2 Impedance control

Whereas the strategy discussed in the previous section involves combining force and position control along different directions, impedance control adopts a fundamentally different perspective. It considers force and position control not as separate entities, but rather as two extremes of a continuous spectrum. When the grasped object is in "free motion" (interacting only with the robot), position control is generally more appropriate. Conversely, when the object enters into rigid contact with its environment, force control becomes crucial. However, most real-world interactions fall between these two extremes, involving complex dynamics where force and position are inherently coupled. This viewpoint is extensively developed in [7].

Rather than specifying a desired trajectory or target force directly, impedance control aims to shape the dynamic relationship between position and force. This is achieved by designing control laws that enforce a desired mechanical impedance on the system, effectively making parts of the robotic system behave like virtual spring-damper systems. The general idea is to define how the system should respond to external disturbances, rather than strictly prescribing motion or exerted force.

Several variants of impedance control have been proposed, as illustrated in Fig. 1.8. In configuration (a), the impedance of the joints is directly regulated, causing the fingers to behave like elastic elements [33]. In (b),

the impedance is specified at the fingertip level [34], whereas in (c), the control aims to regulate the impedance of the object itself [35]. Internal forces can also be controlled, as in configuration (g), where virtual springs connect the fingertips directly. More advanced architectures have been proposed, such as the "inherently passive controller" shown in (d) [36], in which the fingertips are linked to a virtual object whose position can be adjusted via virtual springs.

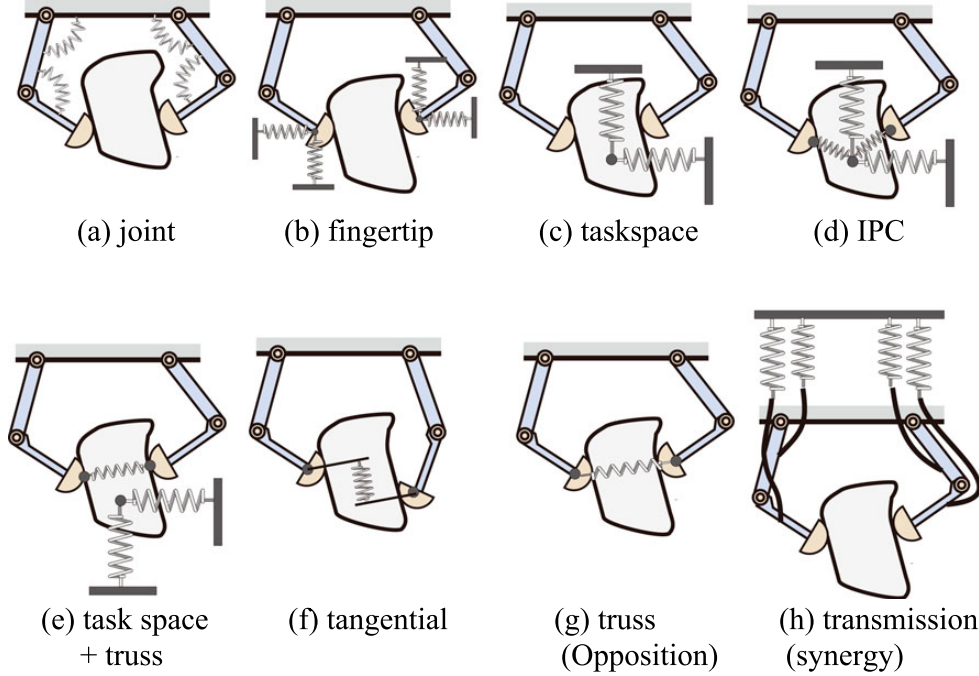


Figure 1.8: Impedance control can be implemented at multiple different levels [23].

Although impedance controllers generally offer lower performance than hybrid force/position control in terms of accurately regulating force or position, they provide greater flexibility in controller design. In particular, some configurations (such as the opposition control in (g) or the joint impedance control in (a)) do not require knowledge of the object dynamic parameters. Others, like the task-space control in (c), do rely on such information, once again demonstrating the adaptability of impedance-based strategies to diverse contexts. Moreover, the inherently passive nature of these controllers contributes to ensuring system stability. Finally, the ability to explicitly specify stiffness and damping properties with respect to external forces proves especially beneficial for advanced tasks, such as tool manipulation or interaction with unstructured environments.

1.4 Synergies

1.4.1 The concept of synergies

Some neuroscience studies [8] have investigated human grasping by collecting hand motion data and applying Principal Component Analysis (PCA) to it. The goal was to determine whether certain motion patterns consistently emerge. A major finding of this work is that **more than 80% of everyday grasping motions can be explained by only two principal components**. These components, commonly referred to as "synergies", are illustrated in Fig. 1.9. This intriguing result suggests that, although the human hand is fully actuated, its control is primarily organized around a few coordinated motion patterns.

These two principal components (also called eigengrasps [37]) can be interpreted as a preferable basis for the grasp configuration space, i.e., the minimal number of linearly independent vectors that, under specific operations, can span the entire set of grasp postures [38]. Consequently, complex hand motions, which would typically require high-dimensional parameterization due to the many degrees of freedom of the hand, can now

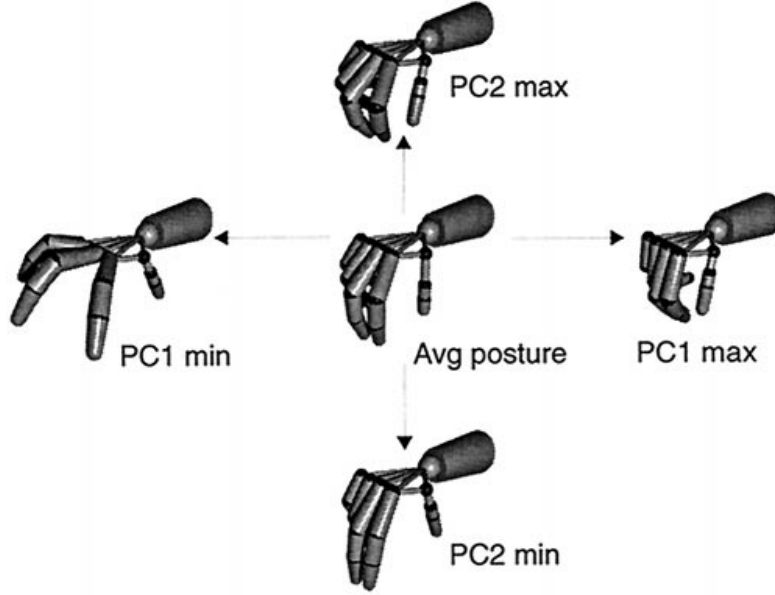


Figure 1.9: Two principal components of postural synergies [8]

be represented using low-dimensional vectors (only two in this case). This substantial reduction in dimensionality considerably simplifies the grasp synthesis problem. The recent discovery of such synergies has thus sparked considerable interest in the scientific community, leading to numerous practical applications that will be discussed in the following sections.

An additional point of interest concerns the relevance of the remaining degrees of freedom. In [8], it is shown that the principal components from the third to the sixth, despite accounting for less than 20% of the total variance, correspond to specific grasping behaviors rather than noise. It is also important to note that the study was conducted without physical objects, with participants reproducing grasp shapes from memory. This suggests that while the early planning stage of a grasp can be described using a low-dimensional representation, the final phase of the grasp adapts more finely to the object through contact interactions. Moreover, these principal components were derived from a dataset of everyday tasks: different components might better represent more specific or complex manipulation scenarios.

In summary, an important portion of the variance in human hand motion is governed by a few preferred coordination patterns, referred to as synergies. While these synergies alone are generally insufficient to fully control a multifingered robotic hand, the dimensionality reduction they offer holds strong potential for robotic applications. When applied to fully actuated hands via dedicated control algorithms, this concept is referred to as **software synergies**. In contrast, **hardware synergies** embed these coordination patterns directly into the mechanical design of the hand, leading to **underactuated hands**. Since the robotic hand used in the remainder of this project is (almost) fully actuated (see Sec. 3.1 for a detailed description), the focus will be on software implementations of synergies. Nevertheless, for the sake of completeness, a brief section will be devoted to hardware synergies and underactuated hand designs at the end of this chapter.

1.4.2 Synergies applications

Synergies find applications at various stages of the grasping task. One of their primary uses lies in simplifying and accelerating the grasp synthesis phase [37]. By taking advantage of the important dimensionality reduction offered by synergies, computationally intensive search algorithms originally operating in the high-dimensional joint space can instead be applied in the much smaller synergy space, thereby making real-time grasp planning more tractable.

In addition to planning, synergies also play an important role in control. At first glance, and as noted in [8],

synergies may appear to provide a purely geometric and kinematic framework, mainly suitable for modeling pre-grasp postures, and seem ill-equipped to deal with real-world interactions. As illustrated in Fig. 1.10e)-h), using the synergystic motions shown in Fig. 1.10a)-d) with an object leads to unrealistic behaviors such as finger-object interpenetration. The desired outcome is rather illustrated in Fig. 1.10i)-l), where the hand conforms naturally to the object shape, achieving a stable grasp without penetration.

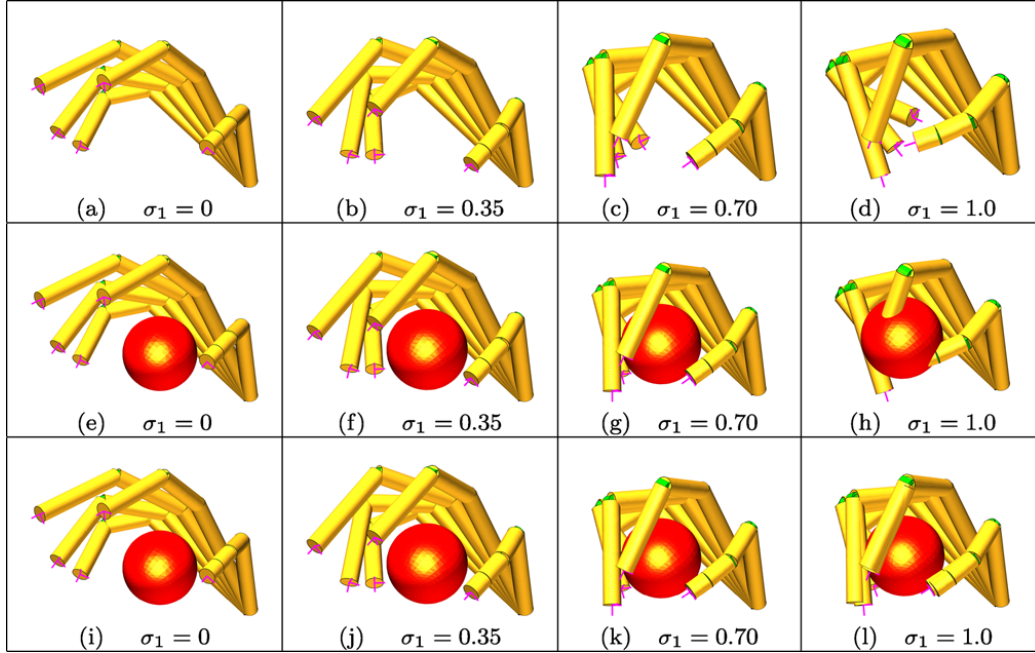


Figure 1.10: Illustration of the synergy-object interaction problem. Top row (a-d): synergy-based hand motion in free space. Middle row (e-h): direct application of the same synergy motion with an object present, resulting in interpenetration. Bottom row (i-l): desired behavior where the hand conforms to the object [39].

Nevertheless, the work presented in [39] explores the role of synergies in the distribution of grasp forces and demonstrates that, when properly adapted, the original synergy model can also contribute to grasp stability and force closure. To account for the interaction with the object, the synergy control framework must be modified to handle contact constraints and object properties. These modified approaches are typically categorized into two main families [9]: soft synergies [40] and adaptive synergies [41]. While soft synergies can be implemented both in software and hardware, adaptive synergies are, at present, only meaningful in a hardware context. They rely on mechanical structures that passively adapt the finger configuration to the object being grasped, and as such, are inherently tied to the physical design of the hand. For this reason, adaptive synergies will be briefly reviewed in the final section of this chapter, which focuses on underactuated robotic hands.

Soft synergies

The concept of soft synergies introduces the notion of a **virtual hand**, which serves as a reference configuration for the real robotic hand. This virtual hand follows a predefined synergy trajectory, and the real hand is attracted toward it through a stiffness model, typically represented as a set of virtual springs. In the absence of contact with objects or the environment, the real hand aligns perfectly with the virtual one, as the equilibrium position of the system corresponds to the synergy-defined configuration.

However, once contact with an object occurs, the final configuration of the hand results from a balance between two forces: the attracting forces exerted by the stiffness model that pull the hand toward the virtual reference, and the repulsive contact forces arising from physical interaction with the object. This results in a compliant behavior that allows the fingers to naturally adapt to the object shape while preserving the overall synergy coordination. This mechanism is illustrated in Fig. 1.11, where the virtual hand is shown in wireframe, partially embedded in

the object to be grasped.

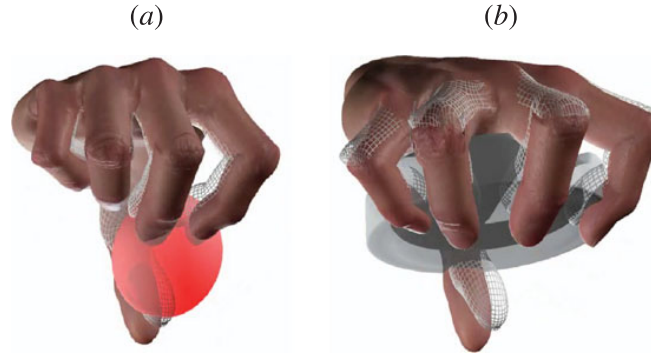


Figure 1.11: Soft synergies: the wireframe shows the virtual hand moving according to a synergy trajectory, serving as a reference for the real hand [40].

Soft synergies thus offer a way to incorporate compliance into synergy-based control strategies, an aspect that was missing from purely kinematic synergy models, and represent a promising approach for enabling robust and adaptive grasping in uncertain environments.

1.4.3 Underactuated hands

The first design principles behind underactuated hands were introduced in [42]. A conceptual illustration is shown in Fig. 1.12a). The main limitation of such position-controlled hands lies in their insufficient number of Degrees of Actuation (DoA), which prevents full control over each joint force independently. To address this, the same authors proposed in [43] to enhance actuation capabilities using Shape Memory Alloys (SMA) embedded in the fingertips, thus compensating for the lack of actuators.

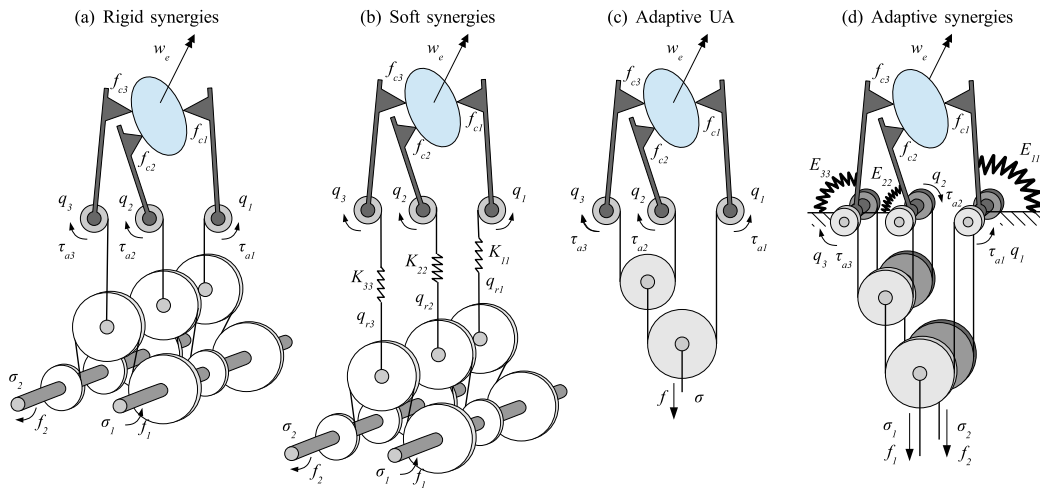


Figure 1.12: Different underactuated hand principles [41]

To further address the absence of compliance in traditional synergies frameworks, soft synergies were introduced. A mechanical implementation of soft synergies was proposed in [44], as schematically represented in Fig. 1.12b). However, this design resulted in a mechanical structure nearly twice as complex as that of a fully actuated hand, limiting its practical applicability.

Another feature that has attracted substantial interest from roboticists is the introduction of **adaptivity**. In this approach, the mechanical design incorporates differential transmission systems, enabling the fingers to passively conform to the geometry of the object being grasped. This passive adaptation enhances the robustness of the

grasp without requiring precise control of each joint. The concept is schematically illustrated in Fig. 1.12c), and has been successfully implemented in several robotic hands, such as the one described in [45].

Finally, the concepts of soft synergies and adaptive hands have been unified in the so-called adaptive synergies framework [41], mentioned earlier. These hands combine passive elastic elements with differential transmission mechanisms, as shown in Fig. 1.12d), thereby enabling both adaptability and compliant interaction through a mechanically efficient design.

1.5 Learning-based methods

In recent years, learning-based approaches have gained growing popularity in the field of dexterous robotic grasping and manipulation, offering promising alternatives to classical control and planning methods. These methods utilize large datasets and powerful function approximators, such as deep neural networks, to cope with the high-dimensional state and action spaces characteristic of multi-fingered robotic hands. Learning-based techniques can be broadly categorized into learning from observation, imitation learning and reinforcement learning [4]. These different approaches will be briefly discussed in this section:

- **Imitation learning (IL):** Imitation learning involves training a policy by directly mimicking expert demonstrations, assuming access to both state and action trajectories. It is particularly useful in tasks with sparse or poorly defined rewards. However, its practicality is limited by the need for accurate and extensive data collection pipelines involving full state-action supervision, which are often costly and difficult to scale. Despite this, imitation learning remains effective in simulation environments or controlled setups where such data can be reliably obtained.
- **Learning from Observation (LfO):** In contrast to imitation learning, LfO assumes access only to observational data such as visual recordings of human or robot behavior, without corresponding action labels. This makes it more scalable and suitable for obtaining large datasets, including videos from the internet or in-the-wild human demonstrations. However, LfO is inherently more challenging, as it requires solving auxiliary perception problems such as object pose estimation, human hand pose tracking, and retargeting of motions to the robot hand, often through inference or auxiliary models.
- **Reinforcement learning (RL):** Reinforcement learning enables an agent to learn optimal behaviors through interaction with the environment, guided by reward signals. In the context of dexterous grasping, RL can be used to develop robust control policies without relying on demonstrations. RL methods are typically divided into model-free and model-based approaches. Model-free methods, such as policy gradients or Q-learning, learn behavior directly from experience but often require extensive training and may suffer from sample inefficiency. Model-based methods, on the other hand, learn a model of the environment dynamics, allowing planning and faster learning, but are often less robust to modeling inaccuracies. Given the physical risks and high cost of training on real robots, sim-to-real transfer techniques such as domain randomization or fine-tuning with real-world data are critical for applying RL policies learned in simulation to real robotic systems. A comprehensive review of reinforcement learning for multi-fingered manipulation can be found in [1].

Chapter 2

Introduction to the virtual mechanisms control framework

The goal of this project is to apply the Virtual Mechanisms (VM) control framework to the specific task of grasping with a dexterous robotic hand. This chapter is dedicated to introducing this control framework. Its structure and content are strongly inspired by the work of Daniel Larby and Fulvio Forni [11], whose paper served as a reference throughout the entire development phase of the project. I would like to extend my sincere thanks to both of them for their continuous support and valuable contributions.

The chapter is organized as follows. Section 2.1 presents the general concept of virtual mechanisms. Section 2.2 explores the modeling aspects and explains how the VM framework operates. Section 2.3 reviews several applications of virtual mechanisms within the robotics community and introduces a few VM controller architectures to provide the reader with concrete insights into their design. This section also emphasizes the relevance of the VM approach to the problem of grasping. Finally, Section 2.4 briefly discusses the optimal tuning of the VM parameters, a central contribution of [11] though this topic lies slightly outside the scope of the present work.

2.1 General idea

The virtual mechanisms control framework has its roots in **passivity-based control**. In this approach, both the robot (or more generally, the plant to be controlled) and the controller are viewed as interconnected systems that exchange energy. The design objective is to ensure that the controller behaves as a passive system, meaning that the total energy supplied to it is always greater than or equal to the sum of the energy stored and the energy dissipated. When two passive systems are interconnected via power-preserving interfaces, the resulting system is guaranteed to be Lyapunov stable and robust to modeling uncertainties, two highly desirable properties in control theory. Achieving asymptotic convergence, however, needs to include additional mechanisms for energy dissipation.

The practical implementation of the controller is done through **energy shaping**, which refers to modifying the total energy function of the system (typically by shaping the potential energy) so that the desired equilibrium becomes a minimum of the new energy landscape [46]. To ensure convergence toward this equilibrium without persistent oscillations, a second step called **damping injection** is applied, which introduces artificial dissipation into the system.

Despite the inherent robustness and stability of these controllers, they suffer from a major drawback: the design process is notoriously difficult. First, deriving an appropriate energy function that aligns with the task objectives is non-trivial. Then, finding suitable controller parameters that yield desirable performance adds another layer of complexity.

The framework proposed in [11] seeks to address both of these challenges. The derivation of the energy function is made more intuitive through the introduction of **virtual mechanisms**, which is the main focus of this chapter. In addition, recent advances in **machine learning** are exploited to automate the parameter tuning process, as will be briefly discussed in the final section.

The idea of virtual mechanisms was first introduced in [47]. The core concept is to virtually attach mechanical components, typically springs and dampers, to the robot. The controller is then implemented in such a way that it emulates the forces that would be exerted by these virtual elements. This strategy introduces mechanical intuition familiar to most engineers into the controller design process. Moreover, it facilitates modularity: multiple task-specific virtual mechanisms can operate concurrently without causing undesirable coupling effects, thanks to the structured energy-based formulation. Finally, while traditional control methods typically aim to avoid or linearize nonlinearities, the virtual mechanisms framework incorporates them as a feature rather than a drawback. Nonlinear elements such as saturating springs or rectified dampers can be deliberately introduced into the controller design to produce a rich variety of complex behaviors. This flexibility broadens the range of achievable dynamics while maintaining the passivity and stability guarantees inherent to the framework.

2.2 Virtual mechanisms fundamentals

2.2.1 Virtual forces

As previously stated, the objective of the controller is to emulate the forces that would be exerted by the virtual mechanisms if they were physically present, by appropriately commanding the robot actuators. These emulated forces are referred to as virtual forces.

From a mathematical perspective, consider a robot whose configuration is described by the vector \mathbf{q} , representing all joint angles. The goal is to compute the actuator input \mathbf{u} that emulates a virtual force \mathbf{F} applied to an arbitrary point \mathbf{z} rigidly attached to the robot. The position of this point can be expressed as a configuration-dependent function: $\mathbf{z} = h_z(\mathbf{q})$. The velocity of point \mathbf{z} is then given by:

$$\dot{\mathbf{z}} = J_z(\mathbf{q}) \cdot \dot{\mathbf{q}},$$

where $J_z(\mathbf{q}) = \frac{\partial h_z(\mathbf{q})}{\partial \mathbf{q}}$ is the Jacobian of point \mathbf{z} . By equating the virtual work performed by the force \mathbf{F} and the actual work done by the actuators through \mathbf{u} , we obtain the following expression for the input torque:

$$\mathbf{u} = J_z(\mathbf{q})^\top \cdot \mathbf{F}.$$

This formulation naturally generalizes to the case where multiple virtual forces are applied at different points on the robot. By the principle of superposition, the total input torque becomes:

$$\mathbf{u} = \sum_{i=1}^n J_{z_i}(\mathbf{q})^\top \cdot \mathbf{F}_i.$$

2.2.2 Virtual elements

Now that the transformation of virtual forces into actual actuator torques has been explained, a natural question arises: How are these virtual forces generated? They result from the interaction between the robot and a conceptual structure referred to as the **virtual mechanism system**. This system is composed of basic mechanical elements, as illustrated in Fig. 2.1, including springs, dampers, inerters, as well as prismatic and revolute joints.

Springs, dampers, and inerters are force-generating components and will be discussed in detail in this section. In contrast, revolute and prismatic joints serve primarily as force-transmission elements and will be addressed in the following section.

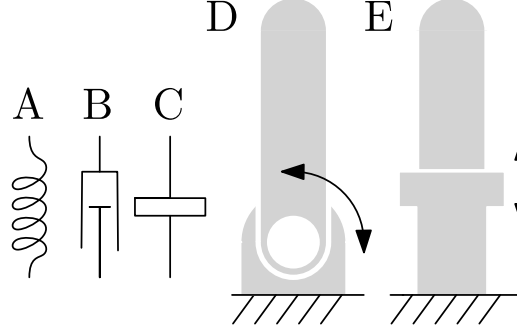


Figure 2.1: Virtual mechanisms symbols: a) Spring b) Damper c) Inerter d) Revolute joint e) Prismatic joint [11]

Springs: Springs are fundamental to the control framework, as they generate motion by defining the equilibrium positions of the system. The spring is characterized by its potential energy function $V(\mathbf{z})$. When connected between the points \mathbf{z}_1 and \mathbf{z}_2 , the extension of the spring is given by $\mathbf{z} = \mathbf{z}_1 - \mathbf{z}_2$. The forces exerted by the spring are expressed as $\mathbf{F}_1^\top = -\frac{\partial V(\mathbf{z})}{\partial \mathbf{z}_1}$ and $\mathbf{F}_2^\top = -\frac{\partial V(\mathbf{z})}{\partial \mathbf{z}_2}$.

For linear springs, the potential energy function is given by $V(\mathbf{z}) = \frac{1}{2}k\mathbf{z}^\top\mathbf{z}$, resulting in the force $\mathbf{F} = k\mathbf{z}$. However, as mentioned previously, non-linear springs can also be employed for numerous applications, such as the saturating spring or the ReLU spring.

The saturating spring uses $V(\mathbf{z}) = \frac{\sigma^2}{k} \ln(\cosh(k\mathbf{z}/\sigma))$ as its energy function, which leads to a force $\mathbf{F} = \sigma \tanh(k|\mathbf{z}|/\sigma) \frac{\mathbf{z}}{|\mathbf{z}|}$. The behavior of this spring is illustrated in Fig. 2.2a, where it is evident that the force the spring can exert is upper-bounded, making it useful for safety checks in human-robot interaction scenarios.

ReLU springs exhibit a ReLU-shaped force profile. They can be easily shifted or combined to generate interesting features. The **deadzone spring** has a dead band around 0, where it does not exert any force, which can be useful, for example, in modeling the limits of a prismatic or revolute joint. The **reverse ReLU spring** applies a repulsive force only when compression exceeds a certain threshold, and will be used later for modeling virtual contacts. Both types of springs are illustrated in Fig. 2.2b.

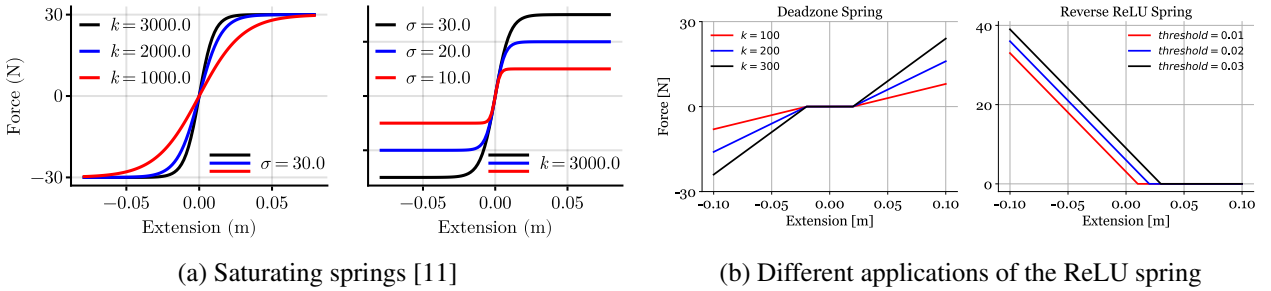


Figure 2.2: Nonlinear springs offer a high level of flexibility.

Dampers: Dampers also play a crucial role in the control of the robot, as they are responsible for energy dissipation, allowing the system to converge to its equilibrium positions. The dissipation is ensured by the condition that, at any instant, $\mathbf{F}^\top \dot{\mathbf{z}} \geq 0$, where \mathbf{F} is the force exerted by the damper at its terminals, as a function of the relative velocity $\dot{\mathbf{z}}$ and the relative displacement \mathbf{z} .

The standard case involves the linear damper, which exerts a force $\mathbf{F} = c\dot{\mathbf{z}}$, with $c \geq 0$. Similarly to springs, a range of non-linear dampers can be considered, such as saturating dampers with $\mathbf{F} = c \tanh(\dot{\mathbf{z}})$, or localized dampers with $\mathbf{F} = c\mu(\mathbf{z})\dot{\mathbf{z}}$, where $0 \leq \mu(\mathbf{z}) \leq 1$, which only act within a specific region.

Inerters: The inerter [48] is an ideal mechanical element that generalizes the concept of mass or inertia to a more generic two-terminal configuration, by exerting a force $\mathbf{F} = m\ddot{\mathbf{z}}$, where m is referred to as the inertance.

When one terminal is connected to the ground, the inerter behaves like a point mass that is unaffected by gravity.

Inerters are typically not connected directly to the robot, since accurately sensing the robot acceleration is often challenging. This design choice is common in impedance controllers, which usually focus on shaping stiffness and damping properties while leaving inertia unaltered [11, 26]. However, inerters can still serve a valuable function when integrated within virtual mechanisms such as prismatic or revolute joints, for which acceleration can be computed analytically. A virtual joint connected to a massless link would theoretically experience infinite acceleration, while a joint connected to a point-mass link would still require knowledge of the robot acceleration. The inerter provides an ideal compromise, allowing mass to be added to the virtual structure without the need to sense the robot actual acceleration.

2.2.3 Virtual structures and implementation

These different virtual elements are interconnected with the robot through prismatic/revolute joints that link rigid bodies. These structures are assigned mass, inertia, or inertance so that they can be simulated as second-order dynamic systems. This approach goes beyond the "simple" virtual springs discussed in the previous chapter, as the introduction of these virtual structures enables more complex features such as virtual tools or digital twins, as will be demonstrated in the next section.

Another aspect that makes this approach more general is the concept of "operation-space" [49]. Indeed, the interconnection between the virtual mechanisms and the robot is accomplished through "operation-space," a highly generic and flexible term. In the case of the spring mentioned earlier, it was connected between two fixed points of the robot and exerted a force in the direction that would drive the difference between these points to zero. However, springs (and dampers/inerters) can be connected to more sophisticated "coordinates." For instance, one could consider only the difference in the x-coordinates of these points, resulting in a spring acting solely along the x-axis. Alternatively, a spring could act on the difference between the value of a joint and a given reference value, functioning like a classic joint-space proportional controller. Geometric properties of the system can be added, stacked, or rotated, creating new coordinates to which springs or dampers can be connected, offering a highly flexible range of possibilities. This behavior is illustrated in Fig. 2.3, where it is shown that the virtual mechanisms act on **interface coordinates**, which can be defined in a very generic manner. For further explanation on the broad vision of operational space control adopted by the VM framework, refer to [50].

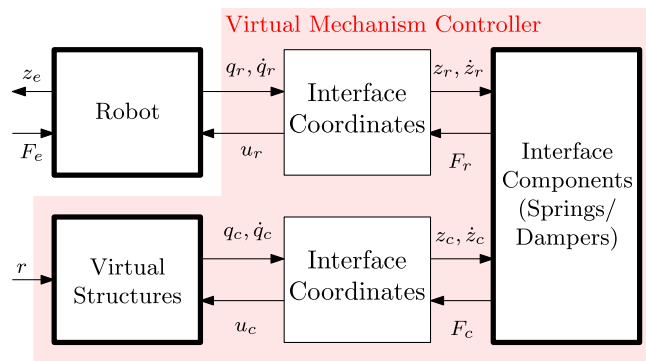


Figure 2.3: Interface coordinates fill the gap between the structures and the virtual components in the controller implementation [11].

2.3 Design examples and applications

Virtual model control originally found its roots in quadruped robots and, more generally, legged robots [47, 51]. However, more recently, after being updated to reflect contemporary advancements by [11], its range

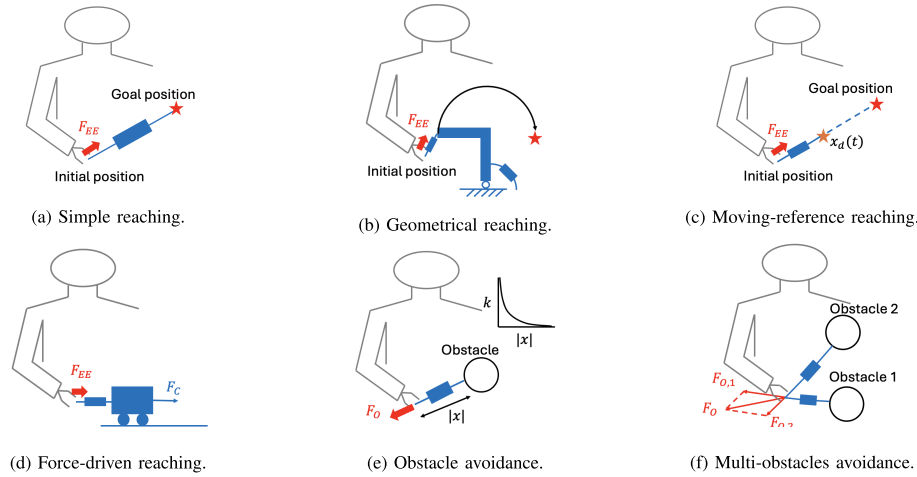


Figure 2.4: Virtual mechanisms performing different reaching tasks [13].

of applications has expanded remarkably. This includes fields such as surgical robotics [50, 12], robotic-arm reaching [13], and even vegetable cutting.

2.3.1 Simple examples

This section aims to present a few virtual mechanism architectures derived from these applications, allowing the reader to gain more intuition about the design of virtual mechanism controllers. These simple examples are taken from [13], which illustrates the multiple ways in which reaching tasks can be performed using virtual mechanism control.

Impedance control reaching: A simple method to achieve reaching is by connecting the end effector to the goal position through a spring/damper pair, as shown in Fig. 2.4 a). This configuration makes the end effector converge towards the target point. This task is referred to as "impedance control" because the robot is not fixed at the end position: applying a force to the robot still allows it to move. During the movement phase, the correct selection of spring stiffness and damper damping coefficients can also enable intuitive human-robot interaction through compliant behavior. As shown in Fig. 2.4 b) and c), a geometric trajectory can be defined using the appropriate structure, and the reference position can move over time.

Compliant path following: A more complex path-following architecture is presented in Fig. 2.4 d), where the end effector is not directly connected to the reference position, but rather to a moving cart. This cart is driven by a constant force source to follow a pre-defined path. If the arm encounters an obstacle during the task, it will stop, and the cart will come to rest once equilibrium is reached. The different aspects of the task are separated: the spring/damper pair connecting the end effector to the cart ensures the robot stays on the path and can be made very stiff if the constraint is critical. Simultaneously, the constant force source on the cart drives the arm along the path, and its force can be adjusted to achieve desired compliance in the direction of motion. The tuning of the friction and mass properties of the cart, as well as the force source, notably influences the final behavior which can be easily understood through mechanical intuition.

Obstacle avoidance: Fig. 2.4 e) and f) illustrate an example of obstacle and multi-obstacles avoidance. The positions of the obstacles are assumed to be known, and the end effector is connected to them via repulsive springs. This is another example of using non-linear springs: in this case, Gaussian springs are employed such that the repulsive forces exerted on the end effector increase exponentially as the robot approaches the obstacles. Although the need for prior knowledge of obstacle positions might seem like a limitation, this controller can actually be integrated with a higher-level controller responsible for providing the real-time positions of obstacles, such as through vision systems.

2.3.2 Advanced architectures

Now that these simple examples have been introduced, two more advanced architectures are presented, originating from the surgical robotics application described in [50]. The considered scenario is a minimally invasive surgical procedure in which the robot end-effector must pass through a specific anatomical point referred to as the Remote Center of Motion (RCM) to access the patient's body. All subsequent motions of the tool must preserve this constraint. The two proposed solutions to this problem are named **Prismatic extension** and **Virtual instrument**, both illustrated in Fig. 2.5.

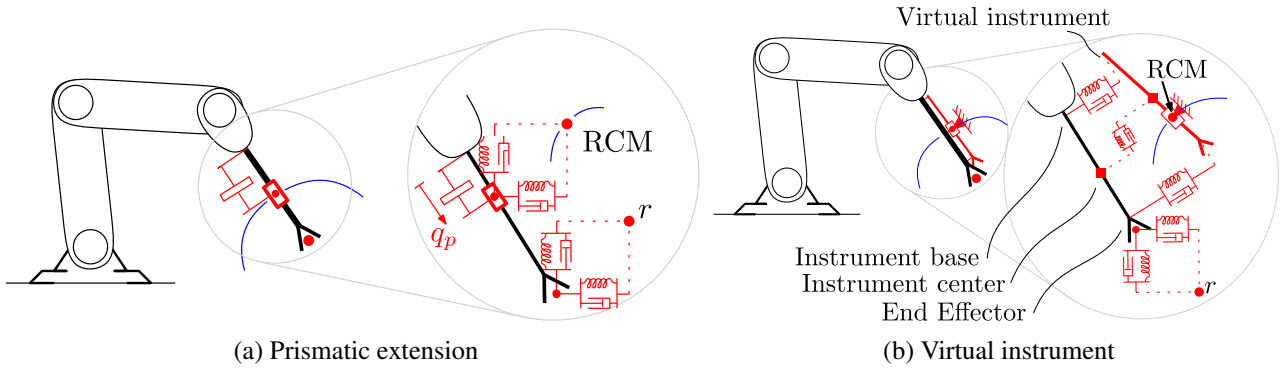


Figure 2.5: Complex virtual mechanism architectures [50]

Prismatic extension: As with the previous examples, the architecture is systematically derived by decomposing the task requirements. The first objective is to guide the tool end-effector towards a reference point r , which is achieved by connecting it to r via spring/damper elements. The second constraint mandates that the instrument must pass through the RCM while retaining the ability to slide along its axis. To enable this motion, a virtual prismatic joint is defined along the tool axis. This joint is connected to the robot structure through an inerter, thereby allowing it to experience well-defined accelerations. The prismatic joint is then connected to the RCM via an additional spring/damper pair. This architecture exemplifies the modular nature of virtual mechanism control and highlights how mechanical intuition facilitates the mapping between task requirements and controller design.

Virtual instrument: In this second architecture, the convergence to the reference point is achieved similarly via a spring-damper connection. However, the RCM constraint is handled differently. Here, a virtual instrument is introduced, as illustrated in red in Fig. 2.5b. This virtual mechanism consists of a revolute joint centered at the RCM, followed by a prismatic joint aligned with the instrument axis. By construction, this ensures that the virtual tool passes through the RCM at all times. As before, mass is assigned to the virtual instrument to allow for acceleration computation. The actual tool is then constrained to remain attached to this virtual instrument via three spring/damper pairs. This architecture offers a refined example of operation space design: among the three spring/damper pairs, the two located at the extremities constrain motion orthogonal to the instrument axis, while the central one constrains motion along the axis. This allows independent control of the three degrees of freedom of the tool, with the potential to assign different stiffness values to each spring depending on the control requirements.

2.3.3 Virtual mechanisms and dexterous manipulation

Although this control framework has not yet been applied explicitly to dexterous grasping, its integration appears highly relevant when considered in light of the various grasping strategies discussed in the previous chapter. In particular, the different scenarios of impedance control illustrated in Fig. 1.8 can be interpreted as particular instances of the more general virtual mechanism framework.

Furthermore, several studies have already introduced other concepts that align with this generalized approach. For instance, the notion of a virtual object was introduced in [36] to model the object being grasped, enabling a

compliant interaction between the robotic hand and the object. Similarly, in [52], the internal forces generated during grasping are captured using virtual linkages, which effectively model the internal constraints and force distribution within the grasp.

Ultimately, these disparate elements can be unified within the virtual mechanism paradigm presented in this chapter. This convergence of ideas provides strong justification for adopting such a framework in the context of this project. The ability of virtual mechanisms to intuitively capture and modularly represent complex mechanical interactions makes them a compelling candidate for developing novel control strategies for dexterous grasping.

2.4 Optimal parameter tuning

As outlined in the introduction to this chapter, the methodology proposed in [11] follows a two-stage strategy. First, the controller is designed in a modular and task-specific manner using the concept of virtual mechanisms. Second, its performance is subsequently optimized through the use of machine learning techniques.

Although the second stage is not implemented in the scope of this work, it represents a natural and promising continuation. For this reason, it is briefly discussed in the present section. Quantitative examples and numerical results illustrating the effectiveness of the different optimization approaches are provided in detail in [11].

2.4.1 Problem statement

The robotic system is modeled as an open system, denoted by R in Fig. 2.6, while the controller is represented by the block C . External disturbances affecting the system are introduced through the input vector \mathbf{w} , whereas the output vector \mathbf{y} represents the variables of interest: typically displacements and velocities relevant to the task.

The controller is parameterized by the vector $\boldsymbol{\theta}$, which encapsulates physical and structural parameters such as stiffness, damping, inertance, and relevant geometric configurations. The objective is to optimize the closed-loop behavior of the overall system by determining the parameter vector $\boldsymbol{\theta}$ that minimizes a performance criterion combining both \mathcal{L}_2 and \mathcal{L}_∞ norms from the input \mathbf{w} to the output \mathbf{y} .

More precisely, both the disturbance and output vectors are decomposed into subcomponents: $(\mathbf{w}_1, \mathbf{w}_2)$ and $(\mathbf{y}_1, \mathbf{y}_2)$. This allows for a differentiated treatment of each performance aspect. The optimization problem can then be formally stated as follows:

$$\min_{\boldsymbol{\theta}} \max_{\mathbf{w}} L(\boldsymbol{\theta}, \mathbf{w}),$$

where

$$L(\boldsymbol{\theta}, \mathbf{w}) = \frac{\|\mathbf{y}_1\|_2}{\|\mathbf{w}_1\|_2} + \frac{\|\mathbf{y}_2\|_\infty}{\|\mathbf{w}_2\|_\infty}.$$

The inner maximization over \mathbf{w} accounts for worst-case disturbances. The first term in the loss function $L(\boldsymbol{\theta}, \mathbf{w})$ captures the average performance (in the \mathcal{L}_2 sense), while the second term penalizes peak responses (via the \mathcal{L}_∞ norm). The particular choice of which components to assign to $(\mathbf{w}_1, \mathbf{w}_2)$ and $(\mathbf{y}_1, \mathbf{y}_2)$ depends on the specific control objectives and task-related considerations.

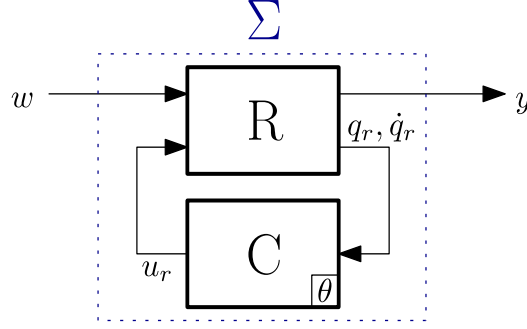


Figure 2.6: Block diagram representation of the robot + controller system [11]

2.4.2 Analytical approaches

The min-max problem presented above is generally intractable due to the infinite dimensionality of the space of potential disturbance signals \mathbf{w} . To address this, certain approximations must be introduced to make the optimization problem analytically and computationally feasible. Two strategies are proposed:

Sampling-based approach: This method involves selecting a finite subset \mathcal{W} of disturbance signals \mathbf{w} that are considered representative or particularly relevant for the task at hand. The optimization problem is then reformulated as follows:

$$\min_{\theta} \max_{\mathbf{w} \in \mathcal{W}} L(\theta, \mathbf{w}).$$

Adversarial approach: Rather than relying on a predefined set of signals, the adversarial method involves parameterizing the disturbance signals as $\mathbf{w}(\omega)$, where ω is a vector of parameters. A sequence of perturbation sets \mathcal{W}_k is then constructed iteratively. At each iteration k , the algorithm identifies the most detrimental disturbance signal, characterized by parameters $\bar{\omega}$ that maximize the performance loss. The process can be summarized as:

$$\min_{\theta_k} \max_{\mathbf{w} \in \mathcal{W}_k} L(\theta_k, \mathbf{w}),$$

with

$$\mathcal{W}_{k+1} = \mathcal{W}_k \cup \{\mathbf{w}(\bar{\omega})\}, \quad \text{for } \bar{\omega} = \arg \max_{\omega} L(\theta_k, \mathbf{w}(\omega)).$$

Although both approaches have proven effective in practice, as shown in [11], they remain inherently suboptimal. This limitation arises from the fact that they only consider a finite subset of all possible disturbance signals, potentially overlooking more critical perturbations.

2.4.3 Algorithmic differentiation

As previously mentioned, one of the main contributions of [11] is the integration of machine learning techniques for the derivation of optimal parameters. More specifically, the approach uses **automatic differentiation**, a powerful tool widely used in modern machine learning frameworks.

This technique requires the optimization problem to be formulated as a differentiable ordinary differential equation (ODE) of the form $\dot{\mathbf{x}} = f(\mathbf{x}, \theta)$, where θ represents the tunable parameters. This ODE must encapsulate the dynamics of both the robot and the virtual mechanism system, as well as the accumulation of the cost function L . While rewriting the virtual mechanism dynamics as an ODE is relatively straightforward, special care must be taken to integrate the cost components \mathcal{L}_2 and \mathcal{L}_∞ in a differentiable way.

\mathcal{L}_2 Component The \mathcal{L}_2 component normally involves computing an indefinite integral of the form $\int_0^\infty \cdot dt$. In practice, this is approximated by a finite integral over a simulation horizon $[0, T]$. To make this term compatible

with automatic differentiation, two auxiliary state variables $\mathbf{c}_{\mathbf{w}_1}$ and $\mathbf{c}_{\mathbf{y}_1}$ are introduced:

$$\begin{aligned}\dot{\mathbf{c}}_{\mathbf{w}_1} &= \mathbf{w}_1^T \mathbf{w}_1, \\ \dot{\mathbf{c}}_{\mathbf{y}_1} &= \mathbf{y}_1^T \mathbf{y}_1.\end{aligned}$$

The resulting \mathcal{L}_2 cost is then computed as:

$$\sqrt{\frac{\mathbf{c}_{\mathbf{y}_1}(T)}{\epsilon + \mathbf{c}_{\mathbf{w}_1}(T)}},$$

where ϵ is a small positive scalar added for numerical stability and to avoid division by zero.

\mathcal{L}_∞ Component For the \mathcal{L}_∞ component, the original max-norm definition is relaxed using a smooth approximation. Two new state variables, $\mathbf{c}_{\mathbf{w}_2}$ and $\mathbf{c}_{\mathbf{y}_2}$, are introduced to track the running maximum of the absolute value of the relevant signals:

$$\begin{aligned}\tau \dot{\mathbf{c}}_{\mathbf{w}_2} &= \max(\mathbf{c}_{\mathbf{w}_2}, |\mathbf{w}_2|) - \mathbf{c}_{\mathbf{w}_2}, \\ \tau \dot{\mathbf{c}}_{\mathbf{y}_2} &= \max(\mathbf{c}_{\mathbf{y}_2}, |\mathbf{y}_2|) - \mathbf{c}_{\mathbf{y}_2}.\end{aligned}$$

For sufficiently small values of the time constant τ , this system converges towards the true infinity norm. The corresponding cost is then computed as:

$$\frac{\mathbf{c}_{\mathbf{y}_2}(T)}{\epsilon + \mathbf{c}_{\mathbf{w}_2}(T)}.$$

By augmenting the system dynamics with these auxiliary states, the entire optimization pipeline becomes fully differentiable. This enables the use of gradient-based optimization techniques powered by automatic differentiation to efficiently compute optimal controller parameters $\boldsymbol{\theta}$.

Chapter 3

Control architecture

The two main tools employed in this work are the **Shadow Dexterous Hand** [53] and the Julia control package **VMRobotControl.jl** [54], developed by Daniel Larby. This chapter aims to present these hardware and software components in detail, thereby laying the foundations for the remainder of the project.

3.1 The Shadow Dexterous Hand

The Shadow Dexterous Hand is an anthropomorphic robotic hand developed by Shadow Robot Company. As illustrated in Fig. 3.1a, it comprises five fingers analogous to those of a human hand: a thumb, a first finger, a middle finger, a ring finger, and a little finger.

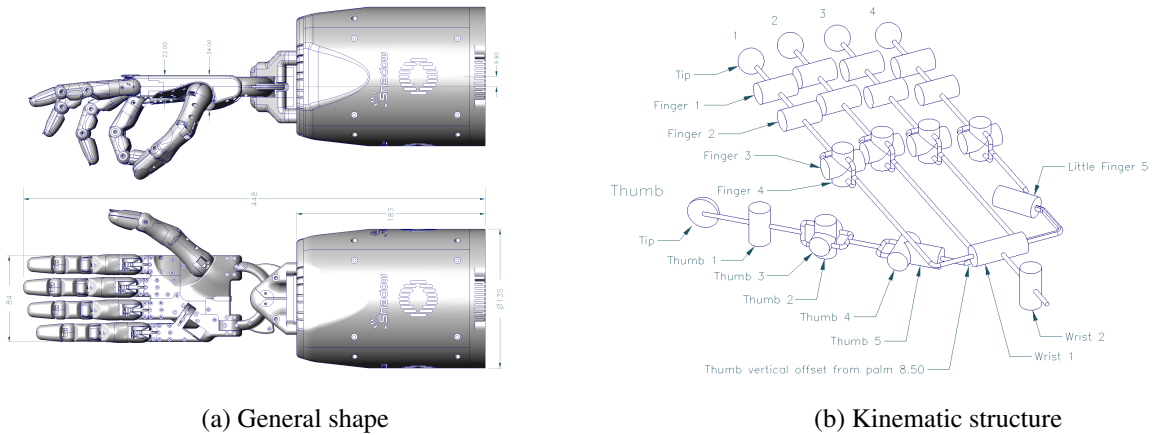


Figure 3.1: Shadow Dexterous Hand structure visualization [55]

The first, middle, and ring fingers share an identical kinematic structure. As depicted in Fig. 3.2a, each of these fingers consists of three phalanges: distal, middle, and proximal (ordered from fingertip to palm) separated by revolute joints. The base joint of each finger allows movement in two planes, specifically along the transverse and sagittal directions. The little finger follows the same kinematic configuration as the other fingers but includes an additional joint located within the palm. This extra degree of freedom enables it to oppose the thumb, thereby enhancing grasping capabilities.

In contrast, the thumb possesses a distinct structure, illustrated in Fig. 3.2b. Its phalanges differ in shape, and the joints are arranged differently. Two revolute joints with axes perpendicular to the thumb main axis separate the distal and middle phalanges. Additionally, the base of the thumb incorporates two revolute joints, one of which is aligned with the finger axis.

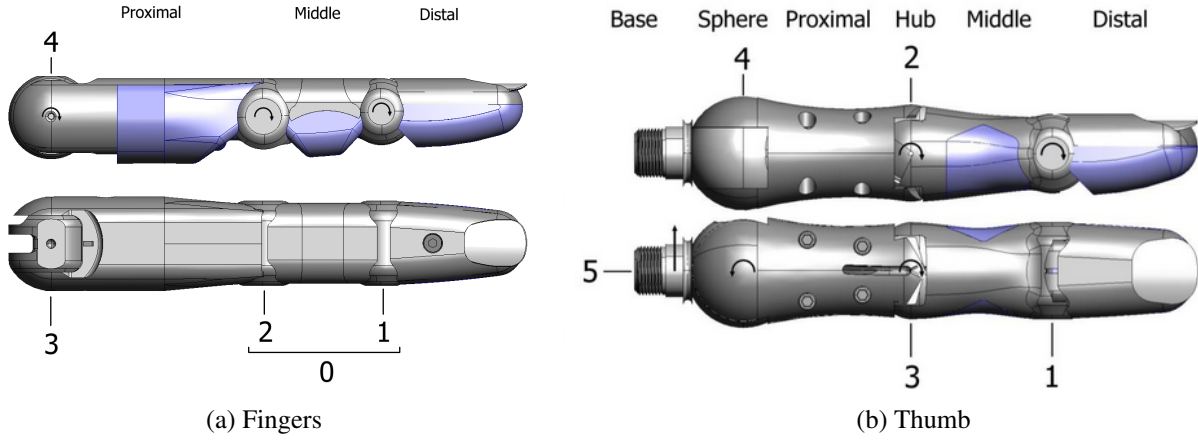


Figure 3.2: Shadow Dexterous Hand fingers and thumb structure [55]

Finally, the hand includes a wrist with two degrees of freedom, bringing the total number of degrees of freedom (DoF) to 24. The distribution of these DoFs is detailed in the kinematic diagram shown in Fig. 3.1b.

Although the hand features 24 degrees of freedom (DoF), it is equipped with only 20 degrees of actuation. Specifically, for each finger except the thumb, the two last joints (denoted J1 and J2) are mechanically coupled and actuated by a single motor (referred to as J0, see Fig. 3.2a). This results in underactuation, an arrangement that mirrors the structure of the human hand. This mechanical coupling enforces a constraint whereby the angle of the proximal joint (J2) is always greater than or equal to that of the distal joint (J1).

Each joint is equipped with a Hall-effect position sensor, enabling **position control**. However, due to the tendon-driven nature of the actuation, accurately modeling internal friction within the hand proves to be infeasible. As a result, torque control is not achievable, which poses a major challenge given that the virtual mechanisms framework is fundamentally designed for torque-controlled robots. The strategies adopted to address this limitation are presented in dedicated sections.

While most joints can be directly controlled, J1 and J2 (due to their mechanical coupling) require indirect actuation through J0. In the absence of external contact, actuating J0 initially causes J2 to move until it reaches its mechanical limit of 90° , after which J1 begins to move. In contrast, when the hand is interacting with the environment, the distribution of motion between J1 and J2 is governed by their internal coupling. This results in a form of built-in compliance, making the fingertips more adaptive to diverse object geometries.

3.2 The VMRobotControl control package

The VMRobotControl package [54] is a Julia software library developed by Daniel Larby as part of his PhD research at the University of Cambridge. It provides an integrated framework designed to simplify both the design and implementation of virtual mechanism controllers. The Julia programming language was chosen due to its combination of high-level expressiveness and high-performance execution. In particular, Julia offers a robust ecosystem for differential equations, matrix operations, and more generally scientific computing, all of which are highly relevant for the development and simulation of virtual mechanisms.

The central concept of the package is the notion of a **Mechanism**. Both the robot and any virtual structures intended to interact with it are modeled as mechanisms, which are interconnected within the control architecture. A mechanism is generically defined by three main components: a rigid-body tree, coordinates, and components.

The rigid-body tree consists of links connected through prismatic or revolute joints and defines the minimal set of configuration variables that describe the system state, namely the joint positions q and velocities \dot{q} . The second element, referred to as coordinates, encompasses points or features of interest derived from the robot

configuration. These coordinates serve as a practical realization of the broader concept of the "operational space" introduced in Chapter 2 and originally formalized in [49]. Lastly, the components include the springs, dampers, masses, and inertias that define the forces and accelerations acting within the mechanism.

Constructing the rigid-body tree of a robot with accurate dimensions, masses, and inertias can be a tedious and error-prone task. To address this issue, the VMRobotControl package includes a **URDF loading** feature, which enables the direct construction of the rigid-body tree from a Universal Robot Description File (URDF).

Once the rigid-body structure is defined, diverse coordinates can be introduced and combined to support the development of complex and expressive behaviors. The main coordinates used throughout this project are summarized in Tab. 3.1. Through appropriate combinations, these coordinates allow the definition of virtual structures that retain mechanical intuition, even if such structures would not be physically realizable in the real world. For instance, a virtual spring acting exclusively along a single spatial axis.

Finally, components, primarily springs and dampers, can be applied to these coordinates. These components may follow standard linear behavior, characterized by specified stiffness and damping parameters, or exhibit non-linear behavior as introduced earlier. The package includes a selection of predefined non-linear elements. However, users are also free to define custom components by extending the package internal data structures.

Coordinate Name	Description
FramePoint/FrameOrigin	A point (or the origin) in a given frame
JointSubspace	Value of a joint
ConstCoord	Constant coordinate
CoordDifference/CoordSum	Difference/Sum of existing coordinates
CoordNorm	Norm of a coordinate
CoordSlice	Takes only some specific indices of a given coordinate.

Table 3.1: The VMRobotControl package [54] offers a wide range of coordinates.

Now that the general concept of a mechanism has been introduced, attention can be turned to the **intuitive controller design**. The approach consists of defining an initial mechanism that represents the robot, and a second mechanism corresponding to the virtual mechanisms. These two mechanisms are then interconnected via virtual components, resulting in what is referred to as the **Virtual Mechanism System (VMS)**, which constitutes the core of the controller. The package automates the generation of the necessary Jacobian matrices for this system. As a result, the user's primary task is to construct the architecture of the virtual mechanism by specifying the desired interconnections and component configurations.

Once the controller has been constructed, VMRobotControl provides a **simulation feature**. Specifically, the motion equations defined by the virtual structure can be automatically reformulated as an ordinary differential equation (ODE) problem, which is then solved using the `DifferentialEquations.jl` library. Although this simulation does not constitute a full physics engine like Gazebo or Mujoco (lacking for instance the modeling of collisions and friction), it remains highly valuable for visualizing the free motion generated by the controller, particularly during the debugging phase.

Indeed, the solution of these simulations can then be passed to the **visualization module**, which exploits the 3D mesh data from the URDF file to produce three-dimensional animations of the robot motion. This graphical feedback provides an intuitive way to assess the behavior and correctness of the designed virtual mechanisms.

The package also includes a feature for **real-time robot control** through ROS. Communication between ROS and Julia is handled via a combination of TCP and UDP sockets: a TCP connection is used to establish the communication protocol, while the actual data transfer occurs over UDP for improved efficiency. The flow of

information between the different components is illustrated in Fig. 3.3. First, the current state of the robot (joint positions and velocities) is received. This data is then used to update the internal representation of the robot within the package. Subsequently, a one-step simulation is carried out for the virtual mechanism component of the system (note that only the virtual mechanism is simulated, not the physical robot). From this simulation, the torques acting on the robot joints are computed and sent back via ROS to be applied to the hardware.

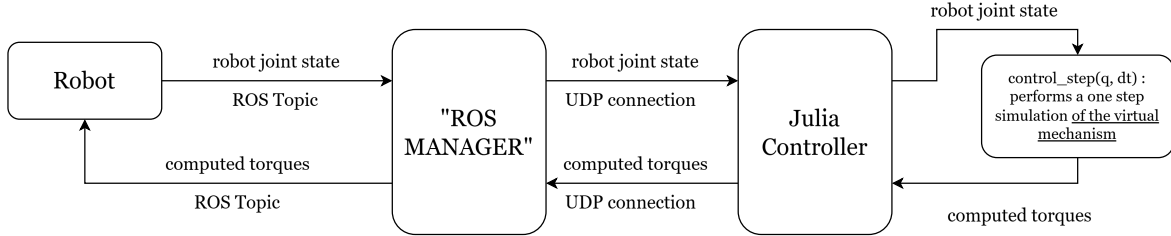


Figure 3.3: ROS control information flow

Finally, **algorithmic differentiation** is also integrated into the package through the use of the `ForwardDiff.jl` and `Enzyme.jl` libraries. These tools allow for efficient and accurate computation of derivatives. Although this functionality is not directly used in the present project, it lays the groundwork for future extensions. Thanks to this integration, the implementation of gradient-based methods could be seriously simplified using the existing infrastructure provided by the package.

Part II

Applications

Chapter 4

Virtual mechanisms as hand-centric trajectory planners

The objective of this master thesis is to explore potential applications of the virtual mechanisms framework in the context of robotic grasping. A major limitation of this approach arises from the fact that virtual mechanisms were originally designed for torque-controlled robots, whereas the robotic hand used in this project is position-controlled. At first glance, this mismatch implies a loss of the native compliance provided by the control framework. However, one natural and still valuable application remains: using the intuitive design principles of virtual mechanisms to generate grasping trajectories. This chapter focuses on this application. The following chapters will address more "advanced" uses of the framework, aimed at generating more complex behaviors and mitigating the limitations of position control.

One of the central challenges in robotic grasping is the **lack of robustness**. While advanced learning-based systems can achieve highly precise manipulation of specific objects [56], they often fail to generalize across a wide variety of objects, as stated in [57, 10]. The approach taken in this chapter adopts the opposite philosophy: rather than aiming for highly specialized manipulations, it focuses on performing the simplest grasping tasks while generalizing to as many object types as possible. The only prior knowledge assumed about the object is its position and a raw idea of its shape. This strategy is inspired by the concept of synergies, which mimics human motor behavior using a reduced number of degrees of freedom.

This chapter begins in Section 4.1 with a review of human grasping taxonomies, which serve as a foundation for designing virtual mechanism architectures capable of replicating such motions. These architectures are described in detail in Section 4.2. Finally, the generated trajectories are evaluated through experiments conducted within the setup presented in Section 4.3, accompanied by an analysis and discussion of the observed results.

4.1 Grasp taxonomies

The human hand exhibits exceptional graspability, allowing it to manipulate a vast variety of objects. However, despite the large number of possible interactions, the hand configurations used during grasping tend to converge toward a discrete set of representative patterns. In response, the scientific community has devoted substantial effort to the development of **grasp taxonomies**: systematic classifications of common human grasp types. The aim of this section is to review the most prominent taxonomies proposed in the literature, to select one that best fits the objectives of this work, and to implement a subset of the grasp types it defines.

4.1.1 Presentation of the different existing taxonomies

The first known grasp taxonomy was proposed by Schlesinger [21], and is illustrated in Fig. 4.1. It identifies six basic grasps based primarily on the geometry of both the hand and the object being manipulated. In contrast,

Napier [58] argued that grasp classification should instead be driven by the nature of the task, distinguishing between two main types: the **Power grip** and the **Precision grip**. In a power grip, the object is held rigidly with the hand, while the precision grip allows for in-hand manipulation. This distinction was later refined by Landsmeer [59], who proposed the term **Precision handling** to better describe the versatility of precision grasps. According to their classification, palmar, lateral, and tip grasps fall under precision grip, whereas spherical and cylindrical grasps are associated with power grips.

This power/precision dichotomy inspired further developments such as Cutkosky's hierarchical tree-shaped taxonomy [22], which organizes grasps according to both structural and functional similarities (see Appendix A.1). Later, an intermediate category, the **Intermediate grip**, was introduced to describe grasps that combine elements of both power and precision [60].

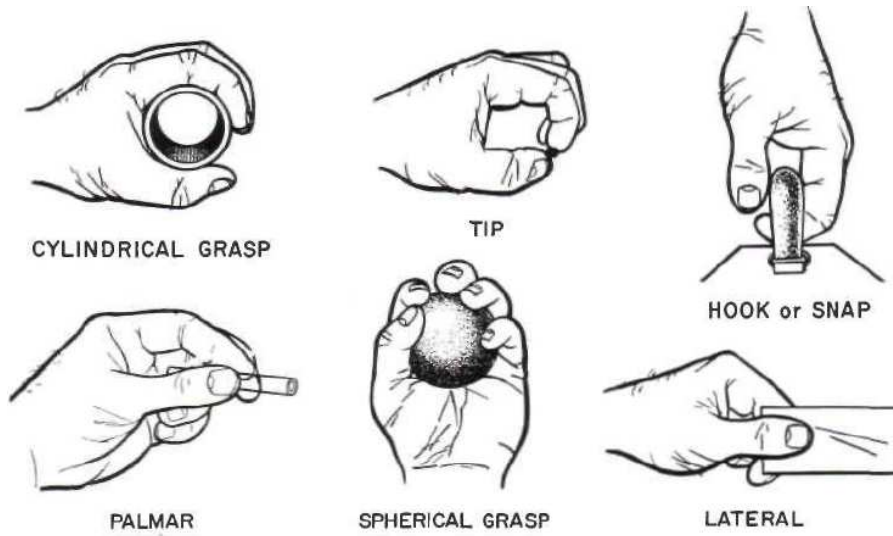


Figure 4.1: Schlesinger's taxonomy [61]

A different perspective was introduced by Iberall through the concept of the **Virtual Finger (VF)**[62]. According to this concept, fingers applying forces in the same direction and of similar nature can be grouped into a single virtual finger. Prehensile grasps are then interpreted as the result of opposition between such virtual fingers. This idea led to a new form of classification defined by the **opposition type**: the axis along which the opposition occurs within the hand structure [63]. This classification is illustrated in Fig. 4.2.

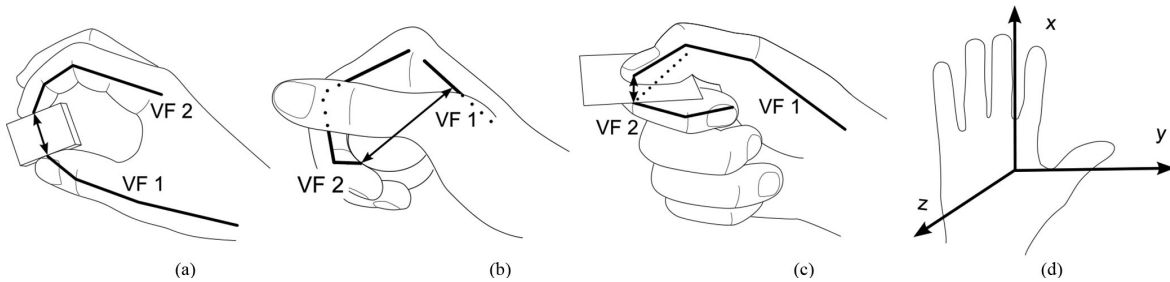


Figure 4.2: Different opposition types: a) Pad opposition b) Palm opposition c) Side opposition d) Hand coordinate system [15].

More recently, Feix [15] proposed a comprehensive taxonomy that integrates the power/precision/intermediate classification, the notion of opposition type, and an additional criterion: the position of the thumb (see Fig. 4.3). The resulting matrix-shaped taxonomy includes 33 distinct grasps, which can be reduced to 17 representative ones by merging grasps lying in the same matrix cell. This taxonomy was developed through an extensive review of prior work and has now become a widely adopted standard. It will therefore serve as the reference taxonomy

in this project. The full matrix can be found in Appendix A.2.

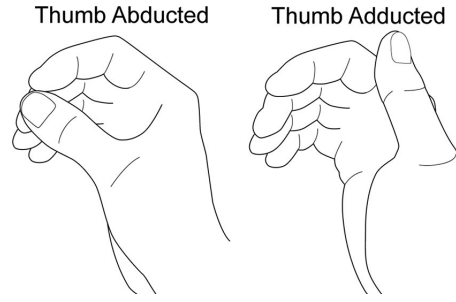


Figure 4.3: Last criteria of Feix's taxonomy: the thumb position [15].

4.1.2 Choice of the implemented grasps

Implementing all 33 grasps (or even the 17 representative ones) from the selected taxonomy is beyond the scope of this project. Instead, the focus is placed on a carefully selected subset. Several studies have attempted to "rank" grasps by considering diverse criteria. For example, some approaches rely on the frequency and/or duration of grasp usage in daily life activities [64].

In this project, the selection strategy is inspired by the work of Bullock et al. [65], who proposed an object-based evaluation method. In their study, a set of commonly used objects is considered, and each grasp is associated with the objects it can successfully manipulate. From this data, a new metric called **grasp span** is introduced, which measures the versatility of each grasp. This allows the definition of optimal subsets of grasps that collectively maximize object coverage.

The study was conducted on two professional populations: housekeepers and machinists. Since the tasks performed by housekeepers are more representative of everyday grasping situations, their results have been retained for this work. The optimal subset derived from this context is illustrated in Fig. 4.4. Drawing from these findings, three grasps have been selected for implementation: the **medium wrap**, the **power sphere**, and the **lateral pinch**.

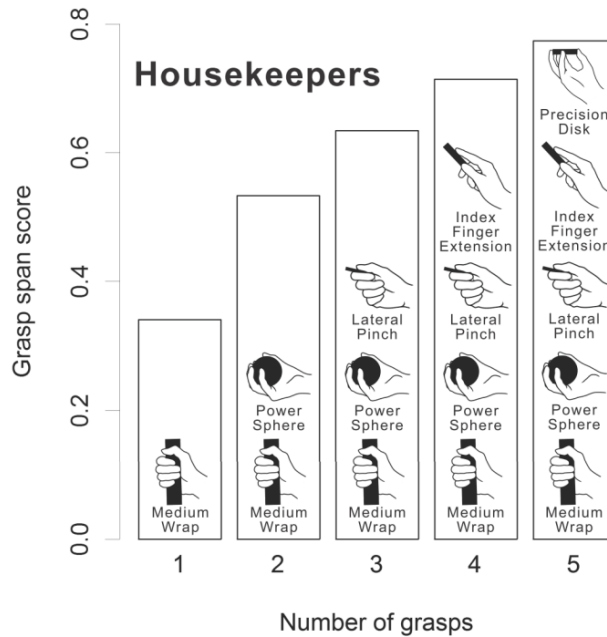


Figure 4.4: Optimal grasps subsets depending on the number of considered grasps [65].

4.2 Virtual mechanisms for grasp motions

This section presents the different virtual mechanisms that have been designed to produce the three grasps identified in the previous section. A fundamental challenge must first be addressed: how can a control framework originally designed for torque-controlled robots be adapted to work with a position-controlled robot? The approach adopted here draws inspiration from the soft synergies discussed in Sec. 1.4. Rather than directly linking the virtual mechanisms to the physical robot, they are connected to a virtual replica referred to as the **virtual hand**. This virtual hand acts as a reference: its joint positions, computed based on the motion of the virtual mechanism, are sent to the physical robot as target positions.

This architecture is illustrated in Fig. 4.5. In addition to the virtual hand and the real robot hand, a third entity appears: the so-called digital twin. This component represents the internal model of the robot maintained by the controller. It is continuously updated to reflect the actual state of the physical hand. Although not yet used at this stage, this digital twin will later play a crucial role in the feedback control loop introduced in Sec. 6.

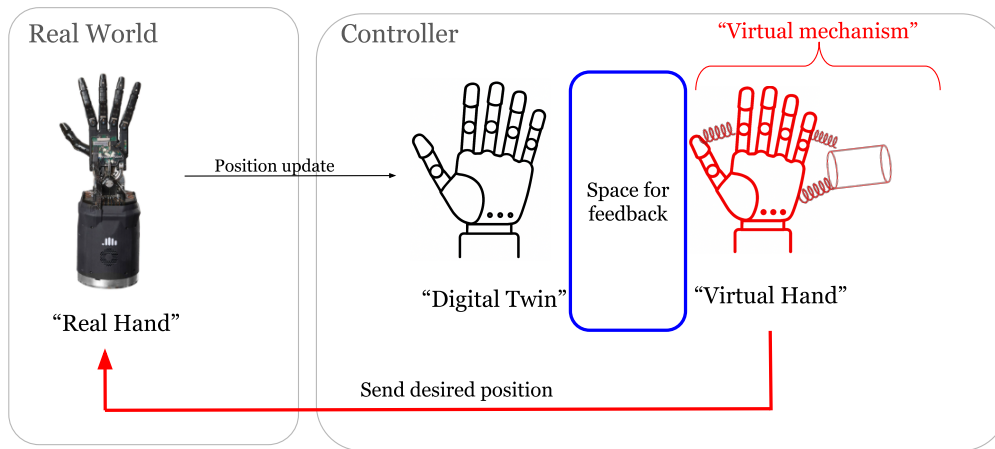


Figure 4.5: Solution for position-controlled robot: the virtual hand.

With the overall control structure now established, the following sections present in detail the motion strategy and the virtual mechanisms developed for each of the selected grasps. These mechanisms are designed to replicate the grasps identified in the previous section. However, so far, these grasps have only been characterized by their final configuration, without addressing the dynamic motion of the hand during the grasping process.

Research on human motor behavior [66] has shown that grasping typically involves two distinct phases: an initial phase during which the hand reaches toward the object while pre-shaping itself, followed by a second phase where the hand encloses the object in the act of prehension. Since this work focuses exclusively on the hand rather than the arm, the reaching phase is not considered. Consequently, the grasping motion is decomposed into two simplified stages: the **pre-shaping** phase and the **prehension** phase.

This distinction is explicitly illustrated in the case of the lateral pinch grasp in [67]. During the pre-shaping phase, the hand configuration is adapted in anticipation of the object geometry and the specific requirements of the selected grasp. Notably, [68] highlights that pre-shaping behavior is influenced by visual uncertainty: when vision is degraded or unavailable, humans tend to increase their hand aperture to account for estimation errors, thereby improving the likelihood of a successful grasp.

In the context of the present work, the uncertainty regarding the object exact dimensions is maximal, as the controller has no information about the object at all. To mitigate this and maximize the probability of a successful grasp, the proposed approach consists of opening the hand to its maximal aperture during the pre-shaping phase, and then gradually closing it. This adaptive motion strategy allows the hand to conform to a wide range of object sizes without prior knowledge of their dimensions.

The underlying principle behind these implementations is to control each grasp using a single degree of freedom. To this end, a dedicated grasp coordinate is introduced for each of the presented grasp types. This scalar parameter varies between 0 and 1, corresponding respectively to the initial (fully open) and final (fully closed) configurations of the hand. By modulating this grasp coordinate, the full grasping motion can be actuated in a smooth and unified manner, simplifying both control and integration with higher-level planning systems.

4.2.1 Medium wrap

The medium wrap grasp is primarily used for objects with a cylindrical shape. In this context, an object is considered cylinder-shaped if it has one dominant dimension significantly longer than the other two, along with a roughly constant cross-section. The pre-shaping phase of the medium wrap consists of aligning the hand with the principal axis of the object and fully extending the thumb to maximize the hand aperture. Subsequently, the four fingers begin to wrap around the object while the thumb moves in opposition to them, stabilizing the grasp. Since the object dimensions are unknown a priori, the proposed trajectory of the fingers is designed such that the hand progressively adopts a circular configuration with a decreasing diameter. This virtual circle remains tangent to the palm throughout the motion, allowing the grasp to adapt naturally to many object sizes while preserving mechanical coherence.

The motion described above has been implemented using the virtual mechanism framework. One of the major advantages of this control approach lies in its modularity: each component of the controller can be designed and tuned independently. This modular structure was exploited to implement the different aspects of the grasping motion, as detailed below:

- **Linking the four palm fingers:** The four fingers of the palm (index, middle, ring, and little fingers) are treated as a single virtual finger. To implement this behavior, each fingertip is connected to a common reference point via standard spring-damper elements, ensuring that its relative positions are maintained. Moving the reference point thus results in coordinated motion of all four fingers.
- **Wrapping motion:** The wrapping of the fingers around the object follows a circular trajectory of decreasing radius, remaining tangent to the palm. This was implemented manually by exploiting an additional feature of the control package: a custom prismatic joint referred to as a "rail joint", which follows a pre-defined curved path instead of a straight axis. By moving the reference point along this curved trajectory, the desired wrapping motion is achieved.
- **Thumb extension:** One of the joints in the robotic hand ($THJ4$) directly controls the thumb extension. The extension motion is therefore implemented straightforwardly by applying a rotational spring-damper system on this joint, allowing it to reach its maximal extension during pre-shaping.
- **Thumb flexion:** During the grasping phase, the thumb transitions from an extended to a flexed position. This behavior is implemented by connecting the thumb tip to the middle phalanx of the index finger using a virtual spring whose rest length gradually decreases over time. This induces a flexion motion of the thumb that conforms to the grasping geometry.
- **Collision avoidance:** Although the VMRobotControl package provides many powerful features, it does not currently support any collision model. As a result, undesired interpenetration between the thumb and the index finger was frequently observed during wrapping motions. To address this, a Gaussian spring was introduced between the thumb and the distal phalanx of the index finger. This virtual spring generates a repulsive force in a narrow vicinity of the target, effectively emulating a soft collision avoidance behavior.

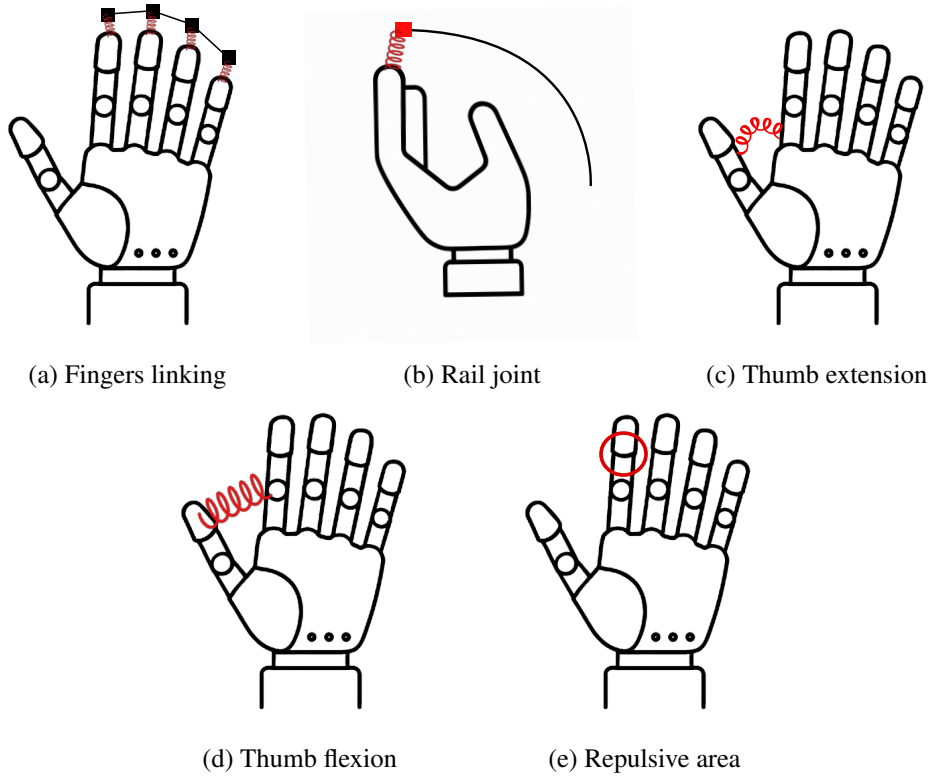


Figure 4.6: Modular design of virtual mechanisms for medium wrap grasp

Varying the grasp coordinate from 0 to 1 drives the entire grasping motion by gradually actuating the underlying virtual mechanisms. Specifically, it moves the reference point along the rail joint to produce the wrapping trajectory, decreases the extension angle of the thumb by adjusting the angular spring, and shortens the spring connecting the thumb to the first finger to induce the flexion motion. Intermediate steps of the resulting motion are illustrated in Fig. 4.7, with the pre-shape as the first image.



Figure 4.7: Final medium wrap motion

4.2.2 Power sphere

As its name suggests, the power sphere is primarily intended for grasping sphere-shaped objects. In this context, a sphere-shaped object is defined as one having approximately equal dimensions along all three axes. The pre-shape associated with this grasp consists of a fully open hand, with the four palm fingers maximally spaced to cover the largest possible area, and the thumb fully opposed to them. The grasping motion is then executed by gradually closing all fingers around the object. Contact is typically first established by the proximal phalanges, followed by the middle, and finally the distal phalanges. The virtual mechanisms designed to implement this grasp are described below:

- **Finger spacing:** To ensure the widest possible spacing between the four palm fingers, spring-damper pairs are applied at the joints connecting the fingers. The springs are configured with rest positions beyond the hand natural range of motion, thereby forcing it to open to its maximum aperture. The thumb opposition is achieved in the same manner as in the medium wrap grasp, through the application of angular spring-damper elements.
- **Sequential closing:** As the grasp coordinate increases toward 1, each joint of the fingers is sequentially actuated. This is accomplished by gradually shifting the rest position of a spring at each joint, from its initial (zero) angle to its maximum flexion. The sequence of joint activation has been carefully tuned to reflect natural human grasping behavior. The final activation timeline is presented in Fig. 4.8, where the joints are, respectively: the MCP joint (between the palm and the proximal phalanx), the PIP joint (between the proximal and middle phalanges), and the DIP joint (between the middle and distal phalanges).

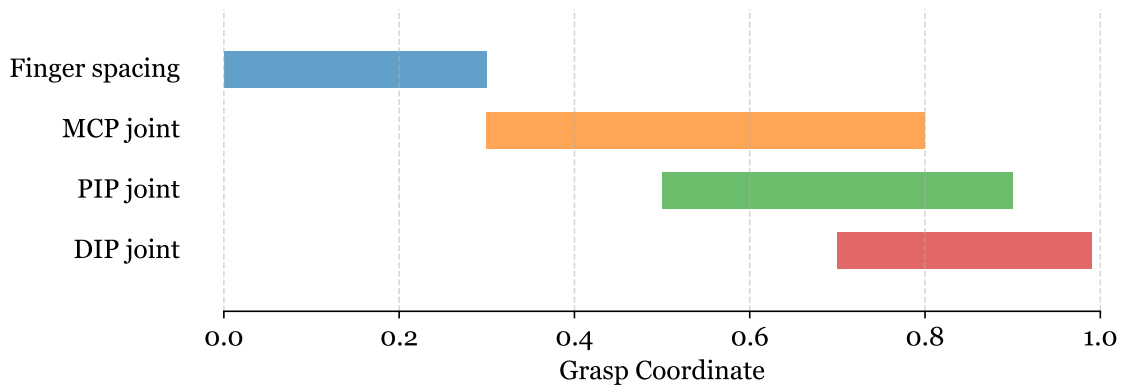


Figure 4.8: The joints are actuated sequentially in the power sphere grasp timeline.

The final grasping motion, obtained through the mechanisms described above, is illustrated in Fig. 4.9, where the sequential activation of each joint can be clearly observed.



Figure 4.9: Final power sphere motion

4.2.3 Lateral pinch

The final grasp, known as the lateral pinch, is employed to grasp flat objects: those characterized by two significantly larger dimensions compared to the third. Such objects are typically pinched between the thumb and the middle phalanx of the index finger. As the remaining three fingers do not contribute meaningfully to this grasp, the pre-shape configuration consists of these fingers being fully closed. The grasping motion is then executed in two distinct phases. In the first phase, the index finger flexes while the thumb moves into position just above the index finger middle phalanx, effectively enclosing the object. The second phase involves the application of pressure by the thumb onto the index finger, securing the object between the two.

The following virtual mechanisms were implemented to produce this motion:

- **Index finger flexion:** As in previous grasps, the flexion of the index finger is achieved by applying spring-damper pairs at the joint level, inducing a smooth closing motion.
- **Thumb positioning:** To bring the thumb to the desired location above the index finger, a positional spring-damper pair is applied between the thumb tip and a dynamically placed target point. To simulate the transition into the second phase of the grasp, this target point is gradually moved inward toward the index finger, causing the thumb to exert pressure against it.

The resulting grasping motion is illustrated in Fig. 4.10.



Figure 4.10: Final lateral pinch motion

4.3 Experimental results and discussion

4.3.1 Practical control architecture

To implement the grasping motions described above in practice, several modifications have been made to the control architecture initially presented in Fig. 3.3.

The first major change concerns the nature of the control signals: instead of sending torques as originally intended in the virtual mechanisms framework, position commands are now transmitted, in accordance with the fact that the Shadow Dexterous Hand is a position-controlled device. More precisely, it is the position of the virtual hand that is sent.

Secondly, while the Shadow Dexterous Hand is controlled by publishing joint values on independent ROS topics, the control package used in this work publishes all joint values as a single message on a unified topic. To reconcile these two formats, an additional ROS node, referred to as the "ros-bridge", has been introduced. This node acts as an interface, converting the unitary command messages into separate signals compatible with the robot. Furthermore, since this node has access to all messages in both directions (i.e., control and feedback), it has been extended with an additional feature: the ability to log data into a spreadsheet-like file for subsequent analysis.

Finally, a key modification involves the timing and synchronization of the control loop. In the original package architecture, the execution of each simulation step is triggered by the reception of new joint state values. In this setting, receiving a joint state update initiates one simulation step of the virtual mechanism, which in turn produces and sends a new command to the robot. For the Shadow Dexterous Hand, joint state updates are received at a fixed rate of 125 Hz. This implies that both the simulation and control commands are updated at the same frequency. However, this configuration proved problematic: at 125 Hz, the simulation became unstable due to the large timestep, while on the control side, the robot was overwhelmed by the overly frequent commands, which did not match the dynamics of the physical system.

To address this issue, the architecture was adapted such that the simulation is executed n times for each joint update, using a timestep divided by n , thereby improving numerical stability. Simultaneously, the ros bridge node includes a filtering mechanism that only forwards one control command out of every n to the robot. The values of n used for simulation and control can, in principle, be different. The final architecture is illustrated in

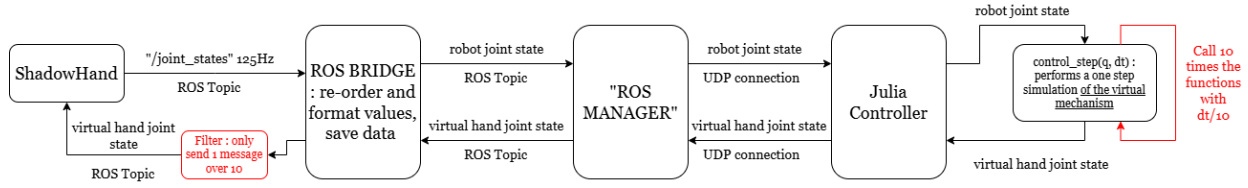
Fig. 4.11, for the case $n = 10$.

Figure 4.11: Modified control architecture

Another important aspect that had to be addressed concerns the coupling of the last two joints of each finger (except the thumb). In the control package, each joint is assumed to be independently actuated. However, on the physical robot, this assumption does not hold: a single actuator (denoted as $J0$) mechanically controls two joints (referred to as $J1$ and $J2$), referred to as the distal (DIP) and middle (PIP) joints of each finger. Due to this coupling, the actual configuration of joints $J1$ and $J2$ depends on their interaction with the environment (e.g., contact forces with grasped objects), and it is therefore not possible to determine a one-to-one mapping between the individual $J1$ and $J2$ values computed in the virtual hand and the single command value $J0$ that must be sent to the real robot. Nevertheless, the hardware guarantees that at all times:

$$J0 = J1 + J2.$$

According to this relationship, it has been decided to compute the command to be sent to actuator $J0$ as the sum of the desired angles for joints $J1$ and $J2$ in the virtual hand. This approach does not guarantee that the real robot will exactly reproduce the configuration of the virtual model, but it enables object interactions and passive compliance to potentially result in similar final poses. In practice, this method proved to be sufficiently robust for generating functional and realistic grasping motions.

Finally, a crucial point concerning the practical implementation of the control architecture must be addressed. In this setup, virtual mechanisms are used as trajectory planners, but they operate without any knowledge of the object to be grasped. As a result, once the hand makes contact with the object, no feedback mechanism is in place to indicate that the trajectory should be interrupted or adjusted. The virtual mechanism continues to generate feedforward commands, attempting to move the hand toward configurations that may lie within the physical boundaries of the object, in a completely blind manner.

Fortunately, the Shadow Dexterous Hand is equipped with both hardware and software saturation mechanisms that limit motor actuation, thereby preventing damage to the hand or the object at least when the object is not overly deformable. These saturation units constrain both the motor torque and the PWM signals sent to the actuators. The motors themselves are located in the forearm and drive the finger joints via a complex tendon routing system. Due to this intricate layout, the internal friction generated along the tendons is difficult to estimate, as it depends on numerous internal and external parameters, including hand posture, joint coupling, and interaction with the environment.

As a consequence, it is not possible to derive a reliable mapping between the torque limits at the motor level and the actual torques at the joint level. Thus, no quantitative assessment of the maximal joint torques during grasping can be provided. This is also the reason why it is not yet possible to control the hand in torque. Nevertheless, in order to offer a qualitative understanding of this phenomenon, a few experiments have been conducted using two slightly deformable objects. The degree of observed deformation serves as an indirect indicator of the forces exerted during grasping. Apart from these specific test cases, all demonstrations and evaluations presented in this section are restricted to rigid and non-deformable objects.

4.3.2 Experimental setup and performances evaluation

Now that the control architecture has been clearly defined, the experimental setup can be described. The Shadow Dexterous Hand used in this work is a right-handed model mounted on a Universal Robot arm. This collaborative

robot can be easily manipulated by hand, allowing flexible and intuitive positioning. The complete system is illustrated in Fig. 4.12. As this work focuses exclusively on grasping, the reaching phase (typically the initial part of a full manipulation sequence) is deliberately omitted. Instead, the robotic arm is manually positioned to an appropriate location, and the object is also placed by hand in a natural and ergonomic configuration that would correspond to a typical human grasp. This manual setup ensures consistent and repeatable initial conditions for each grasping trial.

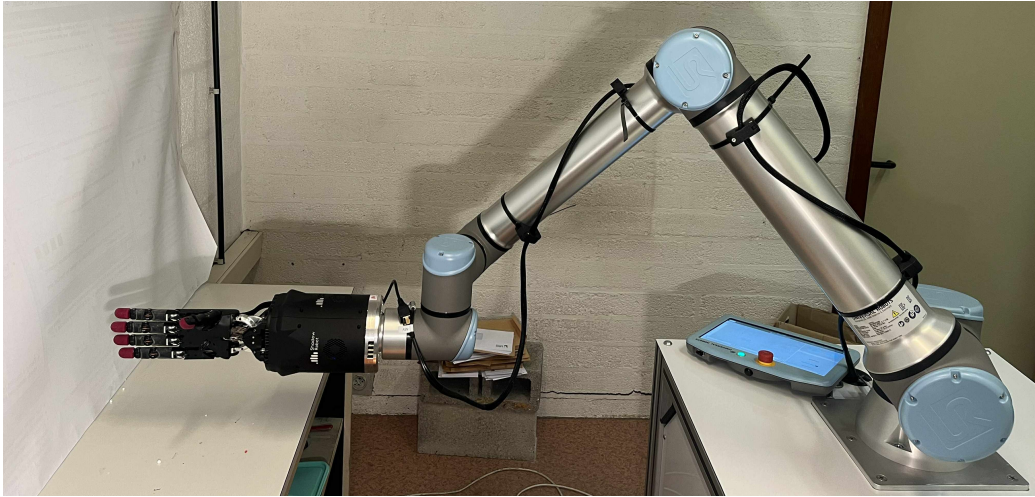


Figure 4.12: Experimental setup

The previously described motions are applied to the experimental setup with the objective of grasping various objects. A first class of results, qualitative in nature, consists of visually assessing the trajectory of the hand, its final configuration, and determining whether the object has been successfully grasped. However, as outlined in Sec. 1.1.1, several criteria define what constitutes a good grasp, and a more quantitative evaluation is desirable. As stated in the corresponding section, three main properties characterize a grasp: stability, robustness, and task-specificity. The latter is inherently addressed through the selection of motions implemented in this work, each tailored to a particular task. Consequently, the analysis will focus primarily on evaluating the **stability** and **robustness** of the grasps.

Stability:

Numerous analytical metrics have been developed to assess grasp stability in theory [27], but such approaches often fail to generalize to practical robotic setups. In particular, it has been shown that high scores on these theoretical quality measures do not necessarily correlate with higher grasp success rates in real-world experiments [30]. Instead, a more reliable approach involves the use of **performance-based measures**, as found in the literature.

For instance, [69] proposes a simple binary criterion: lifting the object and holding it for a fixed period. In [30], the object is also subjected to external perturbations (i.e., shaking) to assess its ability to resist disturbances. A more systematic methodology is introduced in [70], where the evaluation consists of several stages: the object is first lifted and held steady, then shaken progressively with three increasing intensities. The grasp is then classified according to the stage at which the object slips, if any. Finally, in [71], grasp stability is manually tested by an operator who deliberately jiggles the object to break the grasp. A five-level score is then assigned based on the outcome of this inspection.

The procedure involving shaking the object with increasing acceleration was identified as the most principled and systematic evaluation strategy, and was therefore selected as the primary testing method in this project. However, in practice, this approach occasionally fails to expose certain weaknesses inherent to specific grasp configurations. To mitigate this limitation, the evaluation protocol was refined by shaking the object along differ-

ent axes, as not all grasps exhibit their vulnerabilities in the same direction. The practical shaking configurations implemented with the robotic arm are illustrated in Fig. 4.13. In addition to this refinement, it was decided to complement the dynamic shaking evaluation with the manual inspection proposed in [71]. This dual-evaluation strategy provides a more comprehensive assessment of grasp stability, capturing failure modes that may not manifest under shaking alone.

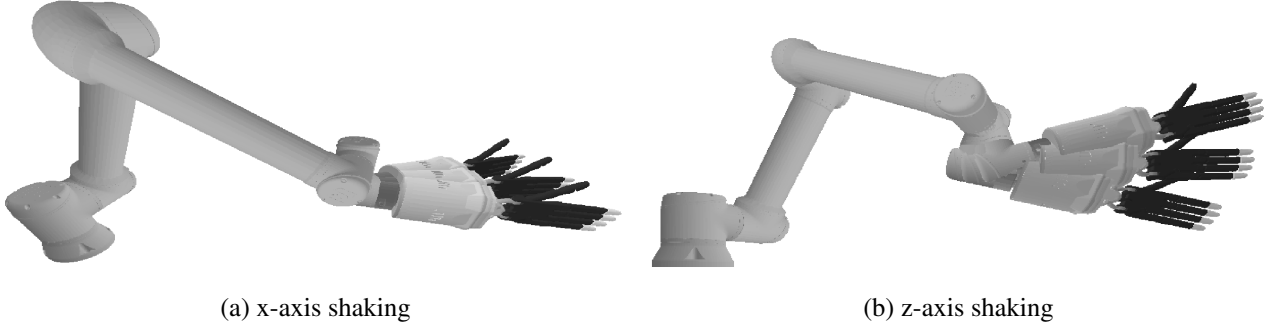


Figure 4.13: Shaking test configurations: the object is shaken along two distinct axes to improve evaluation accuracy.

Ultimately, the results of both methods were aligned on a common scoring scale for simplicity, even though equal scores do not necessarily correspond to equivalent grasp performance. The precise meaning of each score in both evaluation procedures is detailed in Table 4.1.

Score	Shaking Test	Manual Inspection
0	Grasping failed	Grasping failed
1	Object fell during lifting	Grasp fragile (could not carry the object with the grasp)
2	Object fell during first shaking stage	Object movable and droppable by the disturbance force
3	Object fell during the second shaking stage	Object moves by the disturbance force, but will not drop
4	Object fell during the third shaking stage	Stable grasp (unmovable by the small disturbance force)
5	Object did not fall at all	Stable grasp, even against high disturbance forces

Table 4.1: Score-based stability evaluation through shaking and manual inspection.

Robustness:

To evaluate robustness, the grasp generalization ability is tested across a variety of objects. A first and obvious parameter to vary is the **object dimensions**: a robust grasp should accommodate a wide range of sizes. Another critical factor is the **shape** of the object. In real-world applications, objects rarely conform to idealized geometries such as perfect cylinders, rectangles, or spheres. A robust grasp should therefore handle deviations from these canonical forms.

Additional parameters to consider include the **weight** of the object and its **surface properties**, such as friction. Since it is not always possible to isolate each of these factors individually, the approach adopted here is to select a diverse set of objects that together span a wide spectrum of these parameters. The grasps will then be evaluated on their performance across this representative object set.

4.3.3 Results

Before presenting the practical results, it is important to mention a hardware issue that arose during the project. Specifically, a mechanical failure occurred in the *RFJ0* joint (distal and proximal joints of the ring finger), rendering it uncontrollable. This incident took place after the design phase but before the systematic evaluation of grasping performance and the collection of pictures.

To address this issue, the adopted workaround consisted in disabling the affected joint by keeping it fixed at its zero position. Interestingly, despite this limitation, the grasping performance did not appear to be notably impacted. Although this mechanical defect represents a constraint, it also highlights the robustness of the proposed grasping policy. The system remains capable of executing reliable grasps even with a partially disabled hand. This circumstance explains why the ring finger appears consistently extended in nearly all the grasping configurations shown throughout this document.

Medium wrap

The medium wrap grasp is identified in [65] as the most versatile grasp type, capable of handling the largest number of everyday objects. This is primarily due to the omnipresence of cylindrical-shaped items in our daily environment. Consequently, this grasp was the easiest to collect a diverse set of test objects and will be evaluated across the widest range of items among the three considered grasp types.

In addition to its broad applicability, the medium wrap grasp was also selected to qualitatively estimate the maximum joint torques, as discussed in the previous point. For this purpose, two slightly deformable objects were chosen. The first is a **plastic bottle** which, when open, exhibits a moderate level of deformability. The second is a **paperboard tube**, which is more deformable while not being excessively soft or fragile. These two objects, used for torque estimation through deformation observation, are illustrated in Fig. 4.14.



Figure 4.14: Deformable objects for maximum torque assessment

Both objects were successfully grasped. As shown in Fig. 4.15, both the plastic bottle and the paperboard tube exert a low level of deformation. Additionally, it was observed that the bottle was held very firmly: removing it manually from the hand was nearly impossible. From these observations, one can conclude that, although no quantitative information about the joint-level torque limits is available, the saturation thresholds appear to have been appropriately calibrated. They seem to provide an optimal trade-off: sufficient to ensure a very strong and stable grasp, while remaining low enough to avoid damaging any reasonably rigid object. This supports the idea that even without precise torque feedback, the current control strategy can achieve a safe and effective grasping behavior. Consequently, this aspect of the hardware will be considered as given and will not be discussed further in the remainder of this section.



Figure 4.15: Both objects undergo a low level of deformation.

With the hardware aspects now addressed, standard grasping tests can be performed. As a first step, it is interesting to visualize the reference free-motion trajectory executed by the robot in the absence of any object. Snapshots of this medium wrap trajectory are provided in Fig. 4.16, illustrating the coordinated finger movement over time.

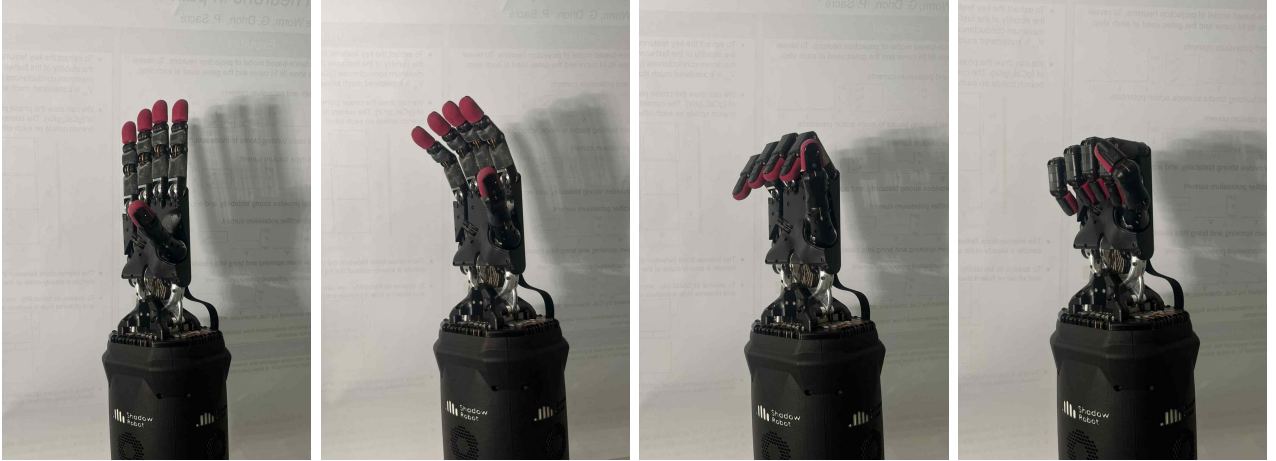


Figure 4.16: Snapshots of the medium wrap trajectory on the Shadow Dexterous Hand.

This trajectory is then used in a series of grasping tests to evaluate its effectiveness. To perform these tests, a diverse set of objects has been selected to span a wide range of characteristics, as outlined in the previous section. All selected objects are shown in Fig. 4.17 and are categorized as follows:

- **Dimensions:** (Fig. 4.17a) A medicine tube with a diameter of 2 cm, a 3D-printed cylinder with a diameter of 5 cm, a metal can with a diameter of 6 cm, and a rice bottle with a diameter of 8 cm.
- **Non-circular cross-section:** (Fig. 4.17b) A 3D-printed square prism with a side length of 5 cm and a square tea box with a side length of 7 cm.
- **Non-constant cross-section:** (Fig. 4.17b) A plastic glass, a beer bottle, and a Jungle Speed totem.
- **Weight:** To isolate the effect of object weight on grasp success, the two 3D-printed pieces have been made hollow (see Fig. 4.17c). This allows their weight to be modified by filling them with rice.

The procedure described in Sec. 4.3.2 has been applied to the selected objects, and the corresponding results are presented in Tab. 4.2. It can be observed that the grasping policy performs very well for objects with different

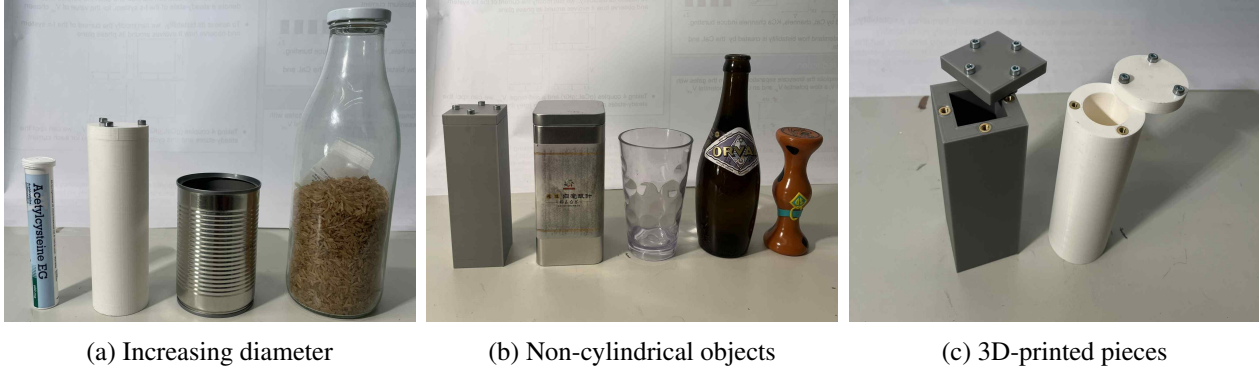


Figure 4.17: Objects used for medium wrap grasping tests.

shapes, but tends to struggle when generalizing to larger or smaller dimensions.

An additional remark can be made regarding the relevance of the shaking test. Notably, only minimum or maximum values are observed in the results. This can be attributed to the nature of the medium wrap, which is categorized as a power grasp. As such, it inherently provides high stability and strong contact forces: once an object is grasped, the grip tends to be robust enough to withstand perturbations, even if the grasp is not perfectly optimal, therefore explaining the binary outcome observed during the shaking tests.

Object	Shaking test	Manual inspection
Medicine tube	5	3
3D printed cylinder	5	5
Metal can	0	0
Rice bottle	0	0
3D printed prism	5	5
Tea box	5	2
Glass	5	4
Beer bottle	5	5
Jungle speed	5	5

Table 4.2: Medium wrap : grasping results for the different objects under test.

In Fig. 4.18, a few steady-state grasping configurations are illustrated and compared to natural human grasp postures. When the object is successfully grasped, the final hand configuration appears natural and closely resembles typical human grasping behavior. In contrast, the grasp of the tea box, which received a very low evaluation score, clearly deviates from the natural human hand posture, as can be seen in the figure. For completeness, photographs of all final grasping configurations (covering all three grasp types) are included in Appendix B.

Finally, the last set of experiments was conducted to evaluate the influence of object weight on grasping performance. The 3D-printed pieces were progressively filled with rice, covering a weight range from 120 g to 240 g. No noticeable differences were observed across the different weights. This result can once again be attributed to the power grasp nature of the medium wrap, which inherently provides a strong and stable grip. As a result, the grasping performance remains robust and unaffected by variations in object weight.

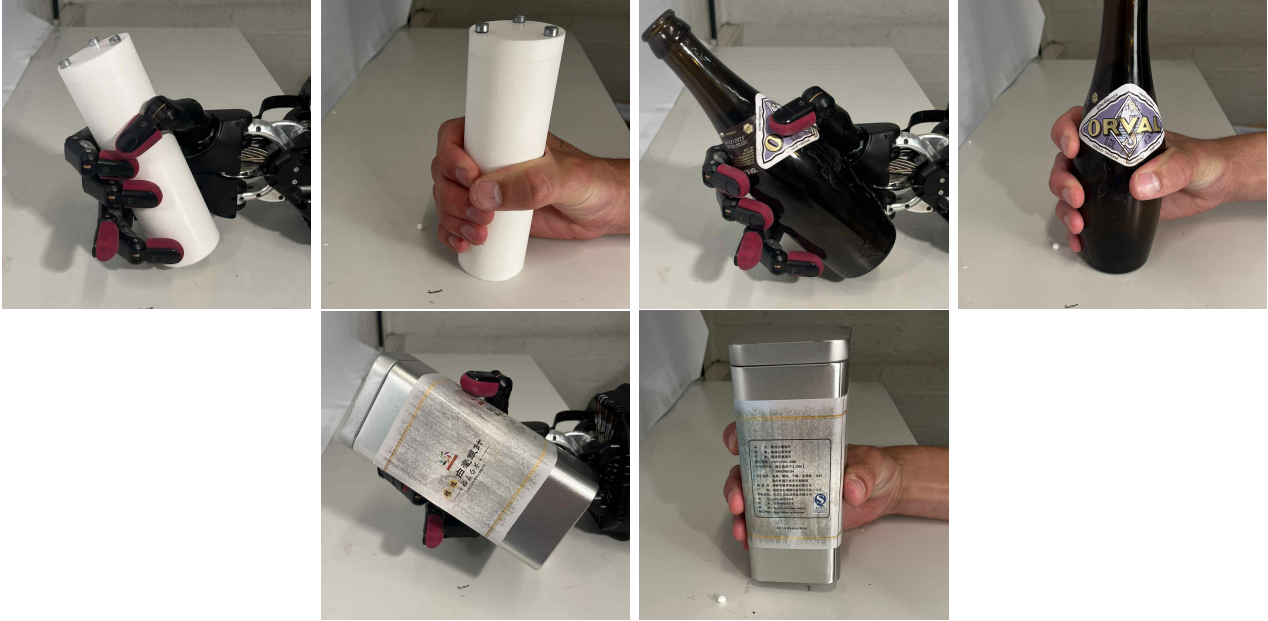


Figure 4.18: Medium wrap final configuration closely resembles human natural posture when the object is correctly grasped.

Power Sphere

Snapshots of the free-motion trajectory corresponding to the power sphere grasp are shown in Fig. 4.19. It can be observed that the final hand posture significantly deviates from a natural closed-fist configuration. This is due to the structure of the underlying virtual mechanism, which only implements the finger-closing motion without any inter-finger collision avoidance. As a result, in the absence of a grasped object, the fingers converge in a configuration that does not resemble a natural human posture.

In a similar manner to the medium wrap tests, a selection of objects has been made to evaluate the performance and robustness of this grasp type. These objects were chosen to span a wide variety of geometrical characteristics and can be seen in Fig. 4.20. They are categorized as follows:

- **Varying dimensions:** (Fig. 4.20a) A ping-pong ball of 4 cm diameter, a billiard ball of 5 cm diameter, a tennis ball of 7 cm diameter, a petanque ball of 8 cm diameter, and a almost-spherical speaker of 11 cm diameter.
- **Non-spherical geometries:** (Fig. 4.20b) A cubic plastic tupperware container, an ellipsoid-shaped egg, and the speaker.

The testing procedure has been applied to the selected objects, and the resulting performance is summarized in Tab. 4.3. Similar observations to those made for the medium wrap grasp can be drawn. In particular, the grasping policy demonstrates a good ability to generalize to non-spherical objects, successfully adapting to varying geometries. However, it struggles when dealing with objects that substantially deviate in size from the training distribution, such as the very small ping-pong ball and the relatively large speaker.

As for the shaking test, the results are again almost entirely binary: either the object is retained securely, or it is immediately lost. This behavior is consistent with the nature of the power sphere, which, as a power grasp, inherently provides strong and stable contact forces when successful.

Representative snapshots of the final grasping configurations are presented in Fig. 4.21, and are compared with natural human grasping postures. Once again, it can be observed that the two hand postures are pretty similar.

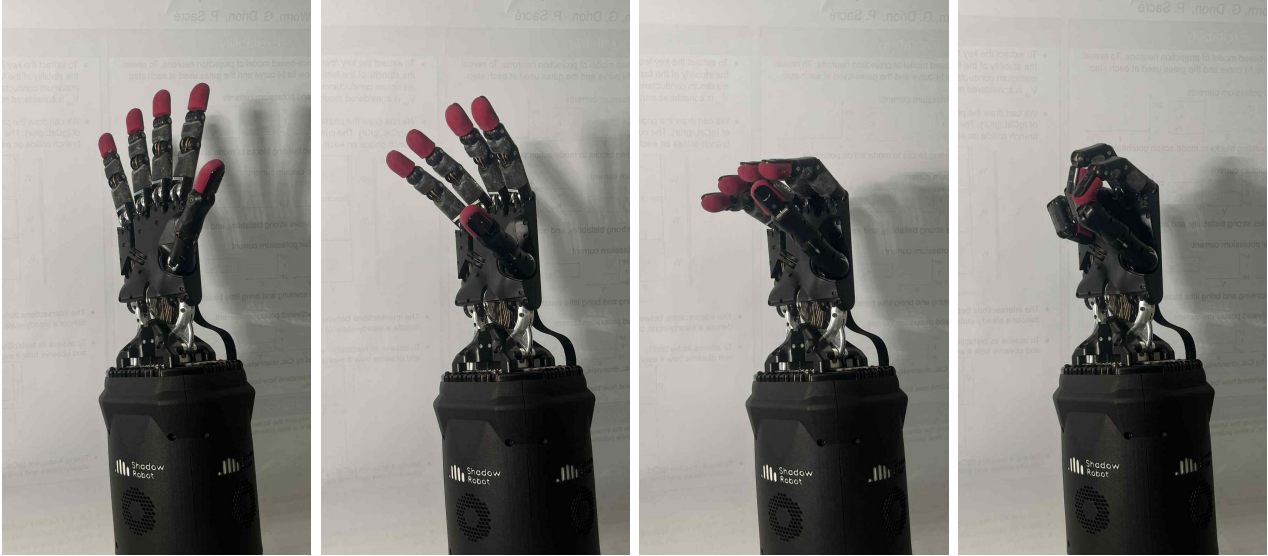
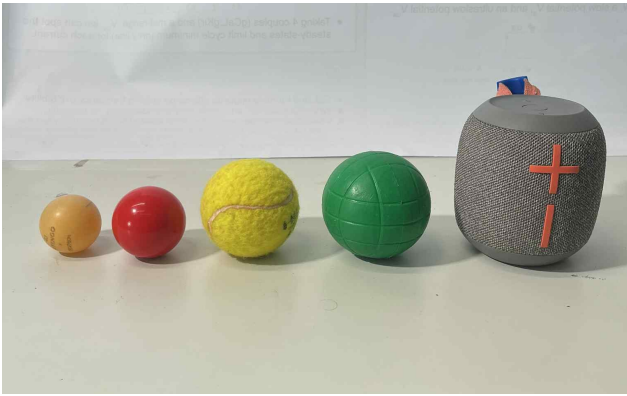


Figure 4.19: Snapshots of the power sphere trajectory on the Shadow Dexterous Hand.



(a) Increasing dimensions



(b) Non-spherical objects

Figure 4.20: Objects used for power sphere grasping tests.

Object	Shaking test	Manual inspection
Ping pong ball	0	0
Billiard ball	5	2
Tennis ball	5	5
Petanque ball	5	5
Speaker	1	1
Egg	5	5
Tupperware	5	5

Table 4.3: Power sphere: grasping results for the different objects under test.

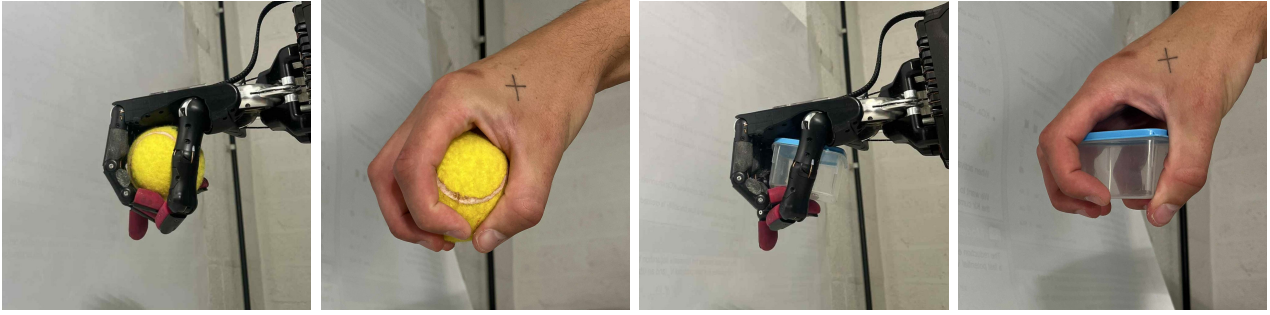


Figure 4.21: Power sphere: comparison with natural human configurations.

Lateral pinch

The structure of this section follows the same format as the previous ones. First, the free-motion trajectory associated with the lateral pinch is illustrated in Fig. 4.22. This trajectory is then evaluated on a diverse set of objects, selected to span a range of relevant characteristics. All tested objects are displayed in Fig. 4.23, and are categorized according to the following criteria:

- **Dimensions:** A flat sheet pad, a thin book (Dragon Ball, 1.5 cm), a medium book (Hunger Games, 3 cm), and a bigger book (Sapiens, 4 cm). Additionally, a small wooden piece with the same thickness as the Dragon Ball book but of smaller width has been included to evaluate the influence of the object width on grasping performance.
- **Non-perfectly flat objects:** A wooden piece containing several holes, and a plastic Tupperware lid.



Figure 4.22: Snapshots of the lateral pinch trajectory on the Shadow Dexterous Hand.

Quantitative score-based results are then depicted in Tab. 4.4, while qualitative comparison with human natural grasps is shown in Fig. 4.24. In this case, the results vary more. Indeed, the precision nature of this grasp makes it more sensitive to shaking and deformation forces: the object is only grasped from one side instead of being fully constrained, and can therefore experience consequential in-hand motion.

Although the influence of weight could not be evaluated as rigorously as in the medium wrap case (flat objects cannot easily be filled with rice), weight nonetheless appeared to have an impact on the grasping results. Indeed, heavier books exhibit higher inertia during the shaking test and are therefore more prone to slipping from the grasp. However, it is worth noting that, in realistic scenarios, if one were to grasp a heavy book with the intent of resisting such accelerations, a different strategy would naturally be adopted, such as a clamp-like grasp.



Figure 4.23: Objects used for lateral pinch grasping tests.

Object	Shaking test	Manual inspection
Sheet pad	4	3
Dragon ball	5	3
Hunger games	4	2
Sapiens	4	2
Small wood piece	4	2
Perforated wood piece	5	3
Tupperware lid	5	3

Table 4.4: Lateral pinch: grasping results for the different objects under test.



Figure 4.24: Lateral pinch: comparison with natural human configurations.

4.3.4 Discussion

In light of the presented results, it can be concluded that the proposed method demonstrates very satisfying performance. The major advantage of this approach lies in its **simplicity**: by merely replicating natural human behavior from a motion perspective, the robotic hand is still able to reliably grasp a variety of everyday objects. This highlights how these human-like motions have been optimized over millions of years of manipulation by the human hand. A particularly noteworthy feature is the **robustness** of the motions, which generalize effectively to a wide range of shapes, even those deviating remarkably from the ones they were originally designed for.

Another crucial advantage is the complete absence of any need for an **object model**. The only required informa-

tion is the choice among the three proposed primitive motions. This selection could, for instance, be automated using simple computer vision techniques that classify objects and assign a suitable grasp. This feature is especially valuable when compared to the methods discussed in Chapter 1, where both traditional grasp synthesis algorithms and learning-based approaches rely on precise 3D models of the objects. Furthermore, several of these techniques also require substantial amounts of training data, which increases the burden in terms of prior knowledge and data preparation.

It has been observed that the stability performance is superior for the medium wrap and power sphere grasps compared to the lateral pinch. This difference does not arise from the implementation itself, but rather from the inherent characteristics of these grasps: both medium wrap and power sphere belong to the "power grasp" category, designed to provide high stability and resistance to external disturbances. In contrast, the lateral pinch is classified as a "precision grasp", making it more suitable for fine manipulation tasks. As discussed previously, this work does not focus on the evaluation of the task-specificity of each grasp. However, one can intuitively understand that the lateral pinch enables a wider range of high-value, precision-oriented tasks, thus compensating for its lower stability.

Despite achieving promising results, the proposed approach still presents several limitations. The most apparent is its difficulty in generalizing to objects with extreme dimensions. This issue stems from the fact that, in such cases, humans naturally adapt their grasping motions to better match the object dimensions, deviating from a generic template motion. While humans may adopt completely different grasps for these cases, they are still able to perform variations of the three considered grasps by adjusting finger positions and contact areas. This adaptive behavior is illustrated in Fig. 4.25, where natural human grasps are shown on the left, and the corresponding adapted versions of the three grasps are shown on the right. For instance, the ping-pong ball is held in a more "pinchy" configuration without palm contact, and the medicine tube is grasped without thumb involvement, using only the fingers and palm.



Figure 4.25: Natural grasping positions vs adapted versions of the three primitives. When the object dimensions decrease, the grasps need to be adapted.

Another limitation of the strategy is that the motions are **entirely blind**. While this contributes to the simplicity of the approach, it also constitutes a major drawback. The hand executes a predefined motion without any awareness of the object being grasped, its movement within the hand, or the actual progression of the grasp. Consequently, no feedback is available to adapt or correct the motion. Moreover, it is not possible to modulate the grasp: the motion is always executed in the same way regardless of the context, and the motors always reach saturation. There is no way to control the "firmness" of the grasp. This limitation also hinders the applicability of the approach to deformable or fragile objects, as previously discussed.

Chapter 5

Virtual mechanisms as object-centric behavior shapers

5.1 General idea

In the previous chapter, the focus was placed on the hand motion, with grasping treated as a consequence of a predefined trajectory. This approach could be viewed as a re-implementation of synergy-based grasping motions, where complex coordination patterns are reduced to a small set of representative movements. While virtual mechanisms were used to facilitate the design of these motions, their full potential was not exploited. Indeed, in this context, the virtual mechanisms could have been replaced by hard-coded trajectories, since the control strategy was entirely motion-driven and independent of the object being grasped.

In contrast, we propose in this chapter an original approach, which we refer to as **object-centric behavior shaping**, a distinction previously introduced in Sec. 1.2.3. Here, the paradigm is reversed: the grasp is no longer the by-product of a prescribed motion but instead, **the motion emerges as a consequence of the object and its interaction with a virtual mechanism**. This object-centric strategy aims to more fully exploit the mechanical intelligence embedded in the virtual mechanisms by encoding primary grasping principles directly within their structure.

To implement this strategy, the controller requires a coarse model of the object. This marks a clear departure from the previous approach, which operated entirely without knowledge of the object. However, in practice, acquiring such a model is not particularly challenging. Simple geometric approximations are sufficient: a cylinder for the medium wrap, a sphere for the power sphere, and a box for the lateral pinch. The only parameters required are estimates of the object dimensions, which can be readily obtained using an RGB-D camera.

This approach lies at the intersection of two extremes. On one end, traditional grasp synthesis and learning-based methods demand highly accurate object models. On the other end, the hand-centric approach described in the previous chapter is completely blind to object geometry and relies solely on robust, pre-defined motions. The object-centric method proposed here aims to combine the strengths of both paradigms: using approximate sensory information to guide the grasp, while relying on robust and generalizable policies to execute it. We believe this hybrid strategy represents a promising direction for developing adaptive and efficient robotic systems.

This chapter is structured as follows. Section 5.2 introduces the virtual mechanism systems used to implement the three object-centric controllers. As previously mentioned, these controllers require an approximate model of the object to be grasped. Consequently, the section begins by explaining how such an object model can be integrated within the virtual mechanism environment. The remainder of the section then details the specific architecture and functioning of each of the three grasping motions. Section 5.3 presents the experimental evaluation of the proposed method. In contrast to the hand-centered strategy described in the previous chapter, the object-centric approach is more sensitive to parameter variations, as the grasping motion fully emerges from the

interaction of virtual elements. The first part of this section is therefore devoted to analyzing the influence of the principal parameters and their tuning. Subsequently, experimental results obtained using the same procedure as described in Section 4.3.2 are reported. Finally, the results are discussed in detail, and potential perspectives for future improvements are outlined.

5.2 Virtual mechanism architectures

5.2.1 Objects models

The first essential component required to implement the aforementioned strategy is an **object model**. The goal is to integrate a geometric shape within the virtual mechanism environment in such a way that the virtual hand is prevented from penetrating the object. Once this object model is in place, additional virtual mechanisms can be connected to it to generate the desired grasping motion.

As previously mentioned, the VMRobotControl package does not provide any built-in collision detection system capable of enforcing such physical constraints. To overcome this limitation, we propose using nonlinear elements, specifically the **Reverse ReLU springs** introduced in Section 2.2.2. These springs exert zero force until the associated coordinate reaches a predefined threshold. Below this threshold, they apply a linearly increasing repulsive force, as illustrated in Fig. 2.2b. By assigning these springs a stiffness much higher than that of any other elements in the system, a collision-like behavior is effectively emulated.

A practical issue encountered with this method is that when any part of the robot comes into contact with the object, the resulting force response is excessively abrupt and produces a highly elastic or "bouncy" effect. This is due to the sudden transition from zero to a high stiffness value. To mitigate this problem, each repulsive ReLU spring is paired with a corresponding damper. By setting the damper activation threshold slightly higher than that of the spring (typically by a factor of 1.1), the system ensures that the robot is gradually decelerated before direct contact occurs. This results in a more realistic and stable interaction with the object.

Each robot frame that should be repelled by the object can be connected to the object using a pair of ReLU spring and damper elements, thereby emulating the collision mechanism. The specific coordinate on which these elements act, along with their threshold value, defines the geometric shape of the resulting virtual object.

Sphere

The simplest object to model is a sphere. Since a sphere is defined as the set of points at a constant distance from a given center, the coordinate to which the spring/damper pair is applied is simply the norm of the distance between the robot frame and the center of the sphere. The threshold value for activation is then equal to the radius of the sphere.

From a software implementation perspective, this corresponds to computing the *CoordDifference* between the robot frame and the sphere center, followed by applying *CoordNorm* to this result, as illustrated in Fig. 5.1.

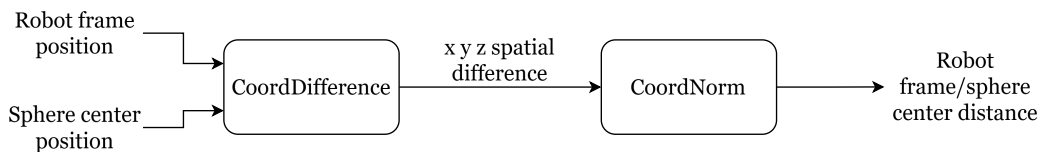


Figure 5.1: Sphere collision model

Cylinder

Modeling a cylindrical object is slightly more complex. To simplify the process, the cylinder is assumed to have infinite length. This allows the modeling of the top and bottom faces to be ignored, focusing only on the

lateral surface. This assumption is justified by the fact that all cylindrical objects considered in this study are longer than the hand width, thus behaving effectively like infinite-length cylinders. If finite-length cylinders were strictly required, techniques similar to those used for the box model (presented below) could be employed.

To compute the distance between the robot frame and the cylindrical surface, a *CoordDifference* coordinate is used. Only the two spatial dimensions orthogonal to the cylinder main axis are relevant. For simplicity, the cylinder is assumed to be aligned with the x -axis, which also corresponds to the natural orientation one would use when grasping a cylindrical object. If the cylinder is not aligned with the global frame, the VMRobotControl package provides a *RotatedCoord* element to transform coordinates into a local frame.

Once the coordinate difference is expressed in the cylinder frame, *CoordSlice* is used to extract the two relevant perpendicular components. Applying *CoordNorm* on this slice yields the distance to the cylinder central axis. Setting the threshold of the spring/damper pair to the cylinder radius effectively emulates a cylindrical object. The entire procedure is depicted in Fig. 5.2.

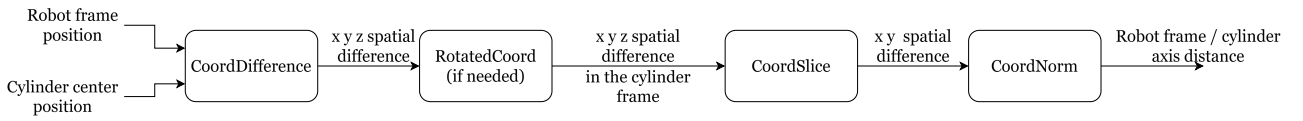


Figure 5.2: Cylinder collision model

Box

Modeling a box-shaped object is considerably more challenging due to its finite faces, unlike the sphere and cylinder, which feature continuous radial symmetry. Since it is not feasible to construct a generic continuous mechanism to emulate a box, a workaround was developed that consists of conditionally activating repulsive springs associated with each face.

At every simulation timestep, the position of the considered robot frame is evaluated. Springs/dampers corresponding to the faces the frame might contact are activated, while the others are disabled by setting their stiffness/damping to zero. This dynamic activation ensures realistic interaction with the finite geometry of a box.

The practical implementation proceeds as follows: first, the position difference is computed, and each spatial dimension is extracted using *CoordSlice*, as different thresholds may be required along each axis. The norm of each sliced coordinate is then used to enable bidirectional repulsion, simulating both opposing faces along that axis. The corresponding ReLU spring and damper pairs are continuously toggled depending on the robot position. While this approach could, in theory, introduce computational overhead due to the frequent updates, in practice, it has no significant impact on simulation performance. The diagram in Fig. 5.3 summarizes the implementation pipeline of the virtual box model described above.

Figure 5.4 illustrates the visual representation of the three object models within the package plotting environment. The position of each object was chosen arbitrarily to align with intuitive grasping configurations. In the current implementation, the object is assumed to be static. Although it would be technically feasible to make the object movable (by connecting its center to the world frame through three prismatic and three revolute joints and assigning it a mass and an inertia), this approach was considered unnecessary. Indeed, the object models used in this work are approximations: their geometry is simplified to one of three canonical shapes, friction is not modeled, and contact dynamics are only coarsely emulated. As a consequence, there is little reason to expect that the object behavior in simulation would accurately reflect its motion in the real world. Moreover, if a grasp is successfully executed, the object is expected to remain relatively stationary with respect to the hand. Therefore, fixing the object position was considered a reasonable and pragmatic modeling choice.

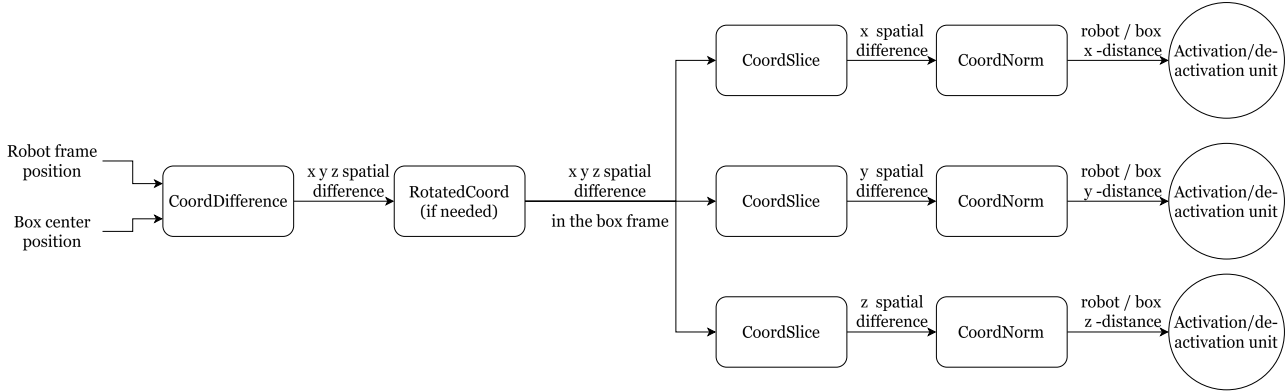


Figure 5.3: Box collision model



Figure 5.4: Visual models inside the package environment

5.2.2 Medium wrap

Once the controller has access to a collision model of the object to be grasped, virtual mechanisms can be introduced between the hand and the object. The idea is to identify the main features of the grasp and translate them into virtual mechanisms. The most critical aspect of grasping is the establishment of **contact** between the fingers and the object. For a medium wrap grasp, one expects contact from each of the three phalanges of the palm fingers as well as from the two phalanges of the thumb. To emulate these contacts, springs are connected between the cylindrical object and each relevant phalanx.

This raises a natural question: where exactly should these springs be attached on the cylinder? The goal is for the controller to adapt to variations in object size. However, when grasping cylinders of different diameters, the positions of the fingers on the surface of the cylinder are not fixed. This is illustrated in Fig. 5.5, where a human

subject naturally adapts finger placement to grasp cylinders of increasing diameters.

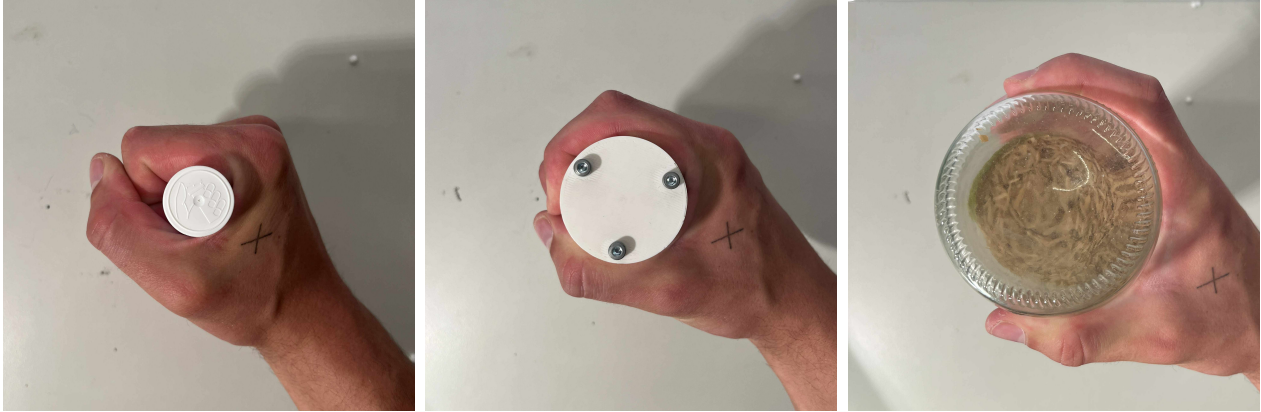


Figure 5.5: Fingers configuration varies with the object size.

To allow the control policy to adapt to objects of varying diameters, the attachment points of the springs on the cylinder must be allowed to move freely along its surface. To implement this additional degree of freedom in the virtual mechanism framework, **prismatic joints** are introduced, enabling the spring endpoints to slide along the cylinder surface. This concept is schematically represented in Fig. 5.6. To ensure the system is fully mechanically defined, each endpoint of the prismatic joints is assigned a mass. This parameter influences the behavior of the overall mechanism, a point that will be discussed later in this chapter.

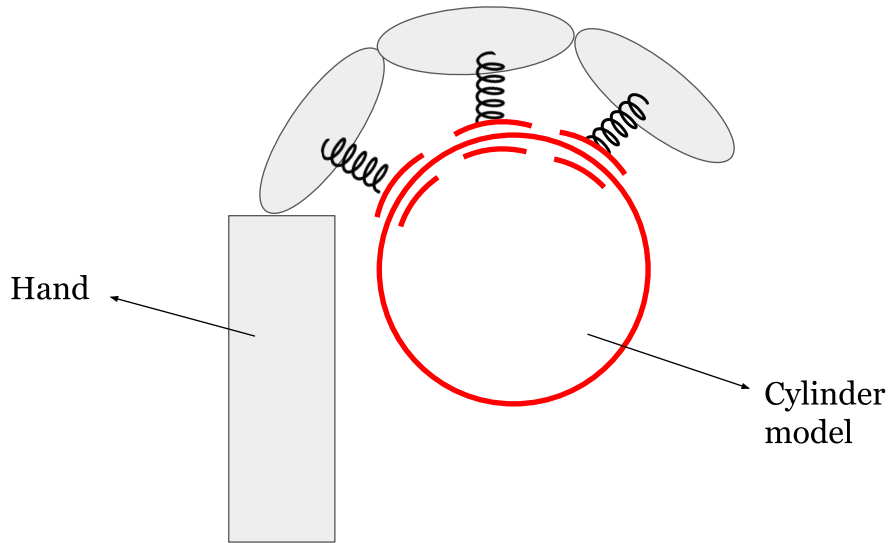


Figure 5.6: Virtual mechanism enabling adaptive contact with the object.

The second important aspect of the grasp is the **wrapping** of the fingers around the object. Indeed, establishing contact alone is not sufficient: the contact points must also be positioned such that the object is mechanically constrained and cannot escape from the grasp. While this constraint could theoretically be enforced using dedicated angular springs, it was observed in practice that a proper wrapping behavior naturally emerges, provided that both the initial hand shape and the stiffness values are appropriately chosen.

Concerning the hand configuration, the pre-shapes introduced in the previous chapter were used. As for the stiffness parameters, the objective was to avoid tuning each individual spring independently. Such manual tuning would risk overfitting the grasp to specific object geometries, thereby reducing generalization to new objects. To address this, the stiffness values were defined in a principled and structured manner: the five fingers and their

three phalanges form a two-dimensional grid, and the stiffness values are generated from a base value that is then modulated using uniform scaling factors along both the finger axis and the phalanx axis, as illustrated in Fig. 5.7.

This approach reduces the problem from fifteen independent parameters to only three: the base stiffness, a scaling factor across the fingers, and another scaling factor across the phalanges. The effect of varying these three parameters and their impact on grasping performance will be analyzed in Sec. 5.3.1.

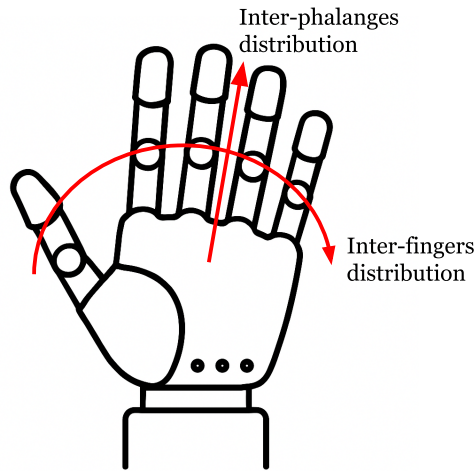


Figure 5.7: The 15 stiffness coefficients are parametrized with only two distributions.

This spring/prismatic joint architecture, combined with an appropriate choice of the three aforementioned parameters, allows the system to reach a satisfying steady-state hand configuration. When applied to real objects, this controller is capable of achieving stable and robust grasps, provided that the object remains stationary during the grasping phase. This limitation stems from an aspect that has not yet been addressed: the **transient phase**.

Indeed, even if the final finger configuration corresponds to a stable grasp, the intermediate trajectories followed by the fingers to reach this configuration are not explicitly defined. As a result, the fingers often strike the object before it is fully constrained, which tends to push it out of the hand. To mitigate this issue, appropriate damping mechanisms must be introduced.

The first step consists of coupling each spring with a linear damper, thereby establishing a uniform level of damping across the system. A critical requirement to prevent the object from being ejected during the transient phase is to ensure that all fingers are close to their final configuration at the time of first contact. This minimizes the object available motion range within the hand. To achieve this synchronization smoothly, a new type of nonlinear damper has been implemented: **the exponential damper**. The damping coefficient of this damper follows a negative exponential function of the distance, reaching its maximum when the distance approaches zero. By carefully selecting the decay rate of the function (i.e., the exponential coefficient), the damping becomes important only within a narrow range close to the contact point. This behavior is shown in Fig. 5.8, where the distance at which the damping reaches 20% of its maximal value is marked by a dotted line. In practice, this causes the fingers to slow down as they approach the object, giving each finger sufficient time to converge toward its final configuration.

Additionally, slowing down the motion near the object prevents premature or unintended collisions that could also kick out the object. Since the controller only has access to a simplified model of the object, early contact may occur if, for instance, the real object is wider than the modeled cylinder at a given height.

While the introduction of exponential damping led to improved grasping success rates in practical tests, many objects were still pushed out before a grasp could be successfully completed. This failure can be attributed to a second factor: **the thumb**. Until now, the thumb has been treated in the same way as the other fingers, using

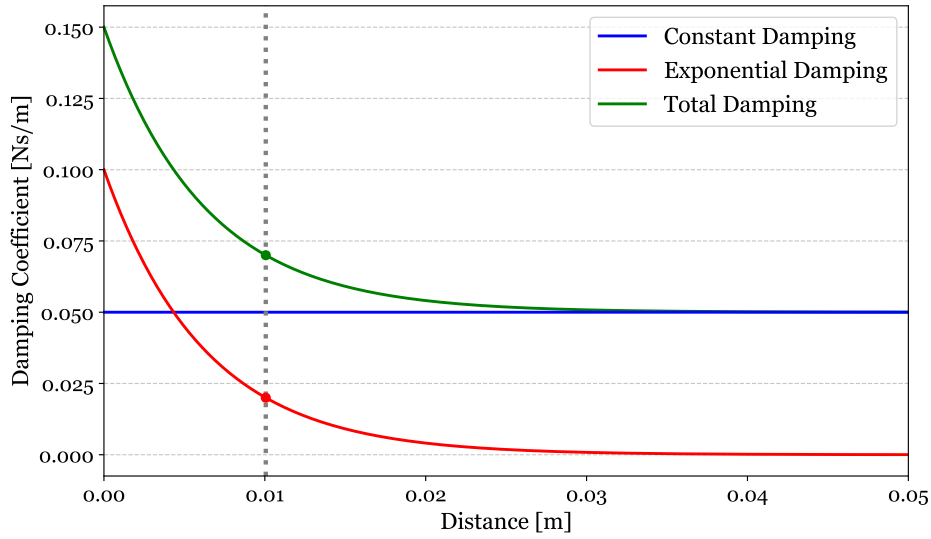


Figure 5.8: Exponential damping enables localized finger slowdown.

similar damping levels and stiffness values derived from the uniform scaling distributions described previously. However, in practice, the thumb executes a substantially different motion. In particular, its range of motion in a wrapping grasp is considerably smaller than that of the other fingers. Consequently, even with exponential damping, the thumb tends to reach the object much earlier than the other fingers, often displacing it before it can be securely enclosed. To resolve this, a stronger damping was applied specifically to the thumb. This was implemented by connecting a linear damper to the middle phalanx of the thumb, with a damping coefficient much higher than that used elsewhere in the system. This targeted damping effectively delayed the thumb movement, thereby preventing premature contact and greatly improving grasping performance.

This virtual mechanism architecture, when evaluated on real objects, yielded very promising results. However, similarly to the previous strategy, its performance deteriorated when the object reached extreme dimensions, either very small or very large. This outcome was somewhat disappointing, as providing the controller with a model of the object was intended to improve its adaptability. Upon further investigation, a critical parameter affecting the grasping performance was identified: the **object position**. As described earlier, the object is manually placed in the robotic hand at a natural grasping position. Simultaneously, the virtual representation of the object within the controller is also positioned at a natural grasping location. The problem lies in the fact that this virtual position was **fixed**. In practice, however, the natural grasping position varies depending on the object dimensions. This led to a mismatch between the real and virtual object positions, particularly for extreme object sizes.

An initial idea was to place the real object at the fixed virtual position defined within the controller. However, this solution proved ineffective: neither the real object nor the virtual one was then positioned naturally within the hand, which resulted in poor grasping performance. The alternative solution, which was successfully implemented, involved providing the controller with a **parametric description of the object position**. This allowed the controller to autonomously compute a natural placement of the object as a function of its dimensions. The real object was then positioned according to this computed placement, rather than being fixed a priori. The challenge was then to develop a parametric model describing the relative positioning between the object and the hand as a function of the cylinder radius.

When analyzing human grasping behavior with cylinders of increasing diameters, a clear trend emerges: people tend to initially place the cylindrical object such that its circular cross-section is tangent to two lines: one defined by the thumb axis and the other by the alignment of the palm and the remaining fingers. A small margin

is typically left on both sides. This behavior changes noticeably when the object diameter decreases. For small objects, the initial position moves higher up into the palm and no longer satisfies the tangency condition. Both trends are illustrated in Fig. 5.9.



Figure 5.9: Variation of the initial object position in human grasping with respect to the cylinder diameter. A smooth transition is observed for large objects, while a bifurcation occurs for smaller diameters.

The behavior associated with large diameters evolves smoothly. In contrast, the shift observed for smaller diameters is more abrupt and resembles a bifurcation: below a certain threshold diameter, the cylinder becomes narrow enough to be fully enclosed by the four palm fingers without requiring the thumb. At this point, the preferred initial position transitions to a fixed location in the palm that enables an efficient wrap using only the four fingers. This critical diameter was empirically estimated to be approximately 3 cm.

According to these observations, the final object placement policy adopted in the controller is as follows: if the cylinder diameter exceeds 3 cm, the object is placed tangent to both the thumb and the palm, as described above. Otherwise, the object is placed at the fixed location corresponding to the wrap-only configuration. This strategy is schematically illustrated in Fig. 5.10.

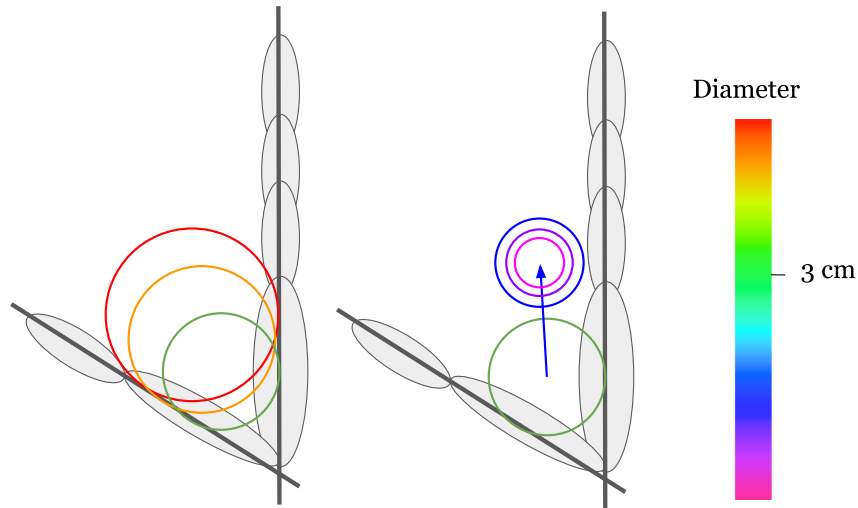


Figure 5.10: Object-centric medium wrap: virtual object position with respect to its diameter. A bifurcation occurs at $d = 3$ cm.

In summary, the main components of the object-centric medium wrap architecture are the following:

- **Virtual cylinder:** Implemented using ReLU springs and dampers, and positioned using a parametric placement strategy. This provides the controller with a simplified yet effective collision model of the object to be grasped.
- **Contact springs:** Each phalanx is connected to the surface of the virtual cylinder via prismatic joints. The stiffness values are parametrized using two uniform scaling distributions, reducing dimensionality while

enabling generalization. These springs ensure a stable and well-constrained steady-state grasp configuration.

- **Damping mechanisms:** Damping is incorporated at three levels: a global constant damping across the entire system, an exponential damping function that increases near contact to synchronize finger arrival, and a much higher damping specifically on the thumb. Together, these elements improve the transient behavior and prevent premature ejection of the object before the grasp is fully formed.

5.2.3 Power sphere

Once again, for the power sphere grasp, the underlying idea is to translate the natural principles observed in human grasping into a virtual mechanism framework. Many of these principles are similar to those employed for the medium wrap grasp. In particular, contact is expected to be established by all phalanges, with the precise contact points depending on the object dimensions. As such, the same spring and prismatic joint pair can be reused to model these contact interactions. Likewise, since a wrapping behavior is desired, the same principled stiffness modulation using uniform scaling distributions can be applied to enable adaptable and robust steady-state configurations.

As in the medium wrap, transient motion may result in the object being unintentionally ejected before all fingers make contact. To mitigate this, the exponential damping strategy introduced earlier is also employed in this grasp. Furthermore, the thumb typically starts notably closer to the object than the other fingers, resulting in premature contact unless corrected. To address this, a dedicated high-damping mechanism on the thumb, previously proven effective, is also integrated here.

The medium wrap and power sphere grasps are conceptually very similar: both aim to wrap the object securely and share many architectural features. The distinction lies in the nature of the objects being grasped. This is accounted for by adjusting the object model, but two **main differences** have been identified, requiring adjustments to the virtual mechanism architecture.

The first of these concerns finger wrapping and containment. Unlike the medium wrap, the object targeted by the power sphere grasp has a finite length. As a result, the object could potentially escape the grasp laterally if no countermeasure is implemented. To prevent this, **finger spacing** becomes essential. The only joints capable of adjusting lateral spacing between fingers are those located at the palm level, specifically the $J4$ joints of each finger (metacarpal joints). To encourage the fingers to spread and cover the widest possible area, angular springs are introduced between these joints. These springs are configured with a rest angle greater than any possible resting configuration, ensuring that they continuously apply a repulsive force, effectively pushing the fingers away from one another and increasing the lateral stability of the grasp.

The second major difference concerns the **positioning of the object**. As in the medium wrap case, a parametric positioning strategy proved highly effective, yet the trends observed differ remarkably for the power sphere grasp. Observations of natural human grasping behavior reveal that, for increasing sphere diameters, the object tends to be consistently positioned tangent to the palm, at a fixed vertical height. However, as the diameter decreases, a clear bifurcation in behavior occurs: once the object becomes too small to be enclosed within the hand (i. e. when its diameter falls below the size of the virtual sphere formed by all five fingertips touching at a common point) the grasping strategy changes. In this regime, the object is no longer placed close to the palm but rather shifted forward, so that it can be pinched using only the fingertips.

This critical transition was empirically observed at a diameter of approximately 4 cm. Knowing this, a simple placement policy was devised: if the object diameter exceeds 4 cm, it is positioned tangent to the palm at a fixed height. Conversely, if the diameter falls below this threshold, the object is placed at a fixed distance from the palm, in a configuration suitable for a fingertip pinch. This strategy is schematically illustrated in Fig. 5.11.

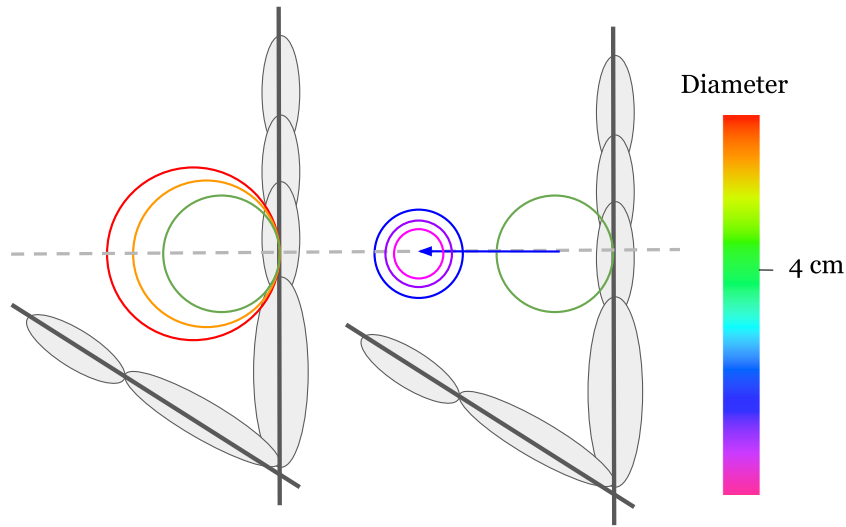


Figure 5.11: Power sphere: virtual object position with respect to its diameter. A bifurcation occurs at $d = 4$ cm.

5.2.4 Lateral pinch

The lateral pinch differs markedly from the two previously discussed grasps in several respects: it does not attempt to fully wrap around the object, it involves only two fingers instead of five, and is typically used to grasp objects with sharp angles and planar faces. Due to these differences, the corresponding virtual mechanism architecture also diverges from the previously presented models, although several of the main features are still reused.

The first crucial aspect remains **the contact** between the fingers and the object. The method of attaching virtual springs to the object via prismatic joints is retained, but with specific adaptations.

First, only the index finger and the thumb are active in this grasp; hence, they are the only fingers connected to the object through springs. Second, unlike the previous grasps where the actuation was primarily driven by the phalanges, the lateral pinch is characterized by a more prominent use of the joints themselves. In natural human grasping, the articulating bones of the fingers often serve as the primary points of contact and support. To replicate this behavior, the architecture connects each of the three joints and the fingertip to the object, rather than the center of each phalanx. This results in four springs per finger instead of three, as illustrated in Fig. 5.12.

The final difference concerns the nature of the prismatic joints. In both the cylindrical and spherical cases, fingers were free to move along the object surface to find their optimal contact location. However, in the lateral pinch, it is desirable that the two fingers end up positioned specifically on the opposing planar faces of the object, namely the top and bottom surfaces of a box-like object. Consequently, the motion allowed by the prismatic joints must be restricted: they are no longer free to translate over the entire object surface, but instead constrained to a single rectangular region per finger. Furthermore, to ensure stability, the fingers must not reach the boundaries of these regions. This is particularly important due to the angular features of the object: contacts too close to edges or corners may lead to unstable configurations. For this reason, a safety margin is introduced, and the prismatic joints are constrained to remain within an inner sub-region of the face, never reaching the outer edges.

Contrary to the previously discussed grasps, the notion of wrapping is no longer present in the lateral pinch. Nevertheless, it is still undesirable for the contact points to be placed arbitrarily along the object surfaces. Instead, the goal is for the fingers to **cover** a sufficiently large area in order to provide stable support to the grasped object.

Careful observation of multiple steady-state lateral pinch configurations across different object sizes revealed

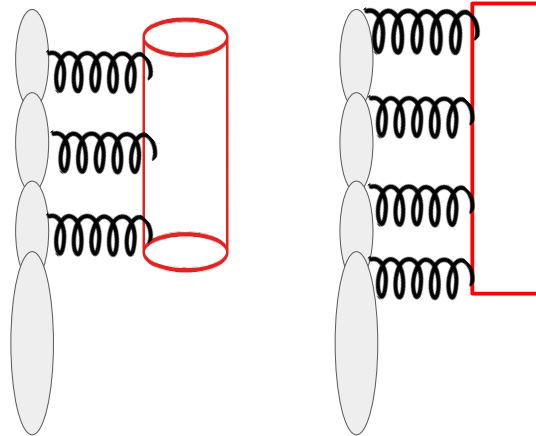


Figure 5.12: The springs are now connected at the joint level instead of the phalanx level.

consistent trends, as can be seen in Fig. 5.13. Typically, the thumb is positioned near the center of the contacted surface, while the index finger tends to extend across the opposite face such that its extremities approach the respective corners of the object. This configuration increases the contact area and contributes to grasp stability by maximizing the mechanical advantage provided by the finger span. To replicate this behavior in the virtual model, additional springs are introduced to pull each prismatic joint toward these target positions. These guiding forces ensure that the final configuration approximates the typical human lateral pinch strategy, promoting both stability and robustness of the grasp.

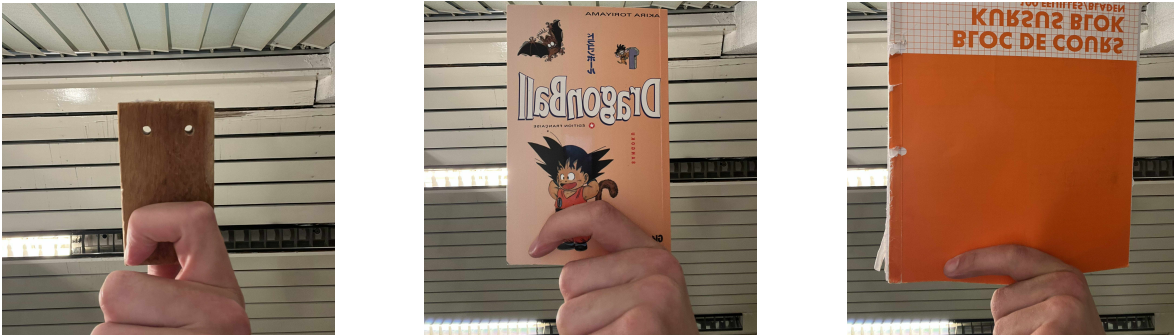


Figure 5.13: First finger placement depends on object width.

As in previous cases, the **transient phase** must also be carefully addressed. The underlying principle remains the same: the fingers should ideally be as close as possible to their final configuration at the moment of first contact with the object. In previous grasps, this was effectively achieved using exponential dampers in combination with massive damping of the thumb. While exponential damping again proved highly beneficial in the lateral pinch context, thumb damping turned out to be ineffective. This was expected, as in this configuration the thumb starts farther from the object than the index finger.

An intuitive alternative would be to reverse the roles: applying massive damping to the index finger instead of the thumb. However, this strategy also failed to yield major improvements. The key difference with the previous grasp types lies in the initial posture of the index finger: although spatially close to the object, it starts in a fully extended position. Consequently, it must perform an important motion in the plane parallel to the box surface, while requiring minimal movement in the perpendicular direction.

To address this issue, a new solution was introduced: the use of spatially non-uniform stiffness in the virtual springs. Specifically, the *CoordDifference* used in the virtual mechanism model is a 3-component vector, and

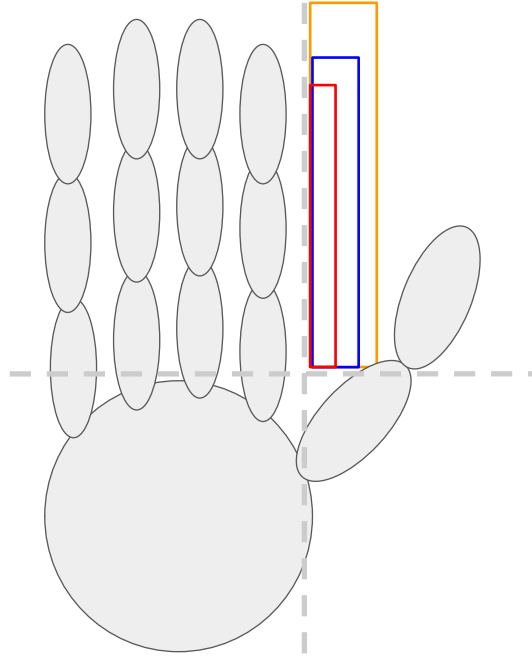


Figure 5.14: Lateral pinch: virtual object position with respect to its dimensions.

until now, a uniform scalar stiffness had been applied equally along all three directions. However, the VM-RobotControl package supports a more flexible approach, allowing the stiffness to be defined as a matrix and thus enabling directional tuning. Using this feature, the springs connecting the index finger to the object were assigned a reduced stiffness in the x-axis direction (corresponding to the perpendicular direction), while maintaining higher stiffness in the other axes. This adjustment effectively slows down the finger motion perpendicular to the object, while preserving speed in the lateral plane, thereby improving the likelihood of a near-final contact configuration upon first touch. When combined with the usual exponential damping, this method notably enhanced transient performance.

Finally, the last aspect that must be addressed is the **positioning** of the virtual object relative to the hand. As in previous grasps, properly parameterizing the object position based on its dimensions plays a major role in enhancing the controller robustness, especially when dealing with objects at the extremes of the considered size range. In the case of the lateral pinch, no abrupt bifurcation in behavior has been observed. Instead, a continuous and consistent trend emerges: regardless of the object dimensions, humans tend to maintain a relatively constant distance between their fingers and the object. This natural tendency, schematically illustrated in Fig. 5.14, has been incorporated into the control environment through a parametric placement strategy.

5.3 Results and discussion

5.3.1 Simulation results: parameter analysis

This section aims to analyze the influence of each of the previously introduced parameters by systematically varying their values within the simulation environment. The objective is to qualitatively assess how these changes affect the grasping dynamics, in order to identify the most effective parameter configurations. Since the three grasp types share several common features, the discussion will be kept as general as possible, highlighting trends and observations that are relevant across all grasping strategies.

Mass: prismatic joints ends

To ensure a fully defined mechanical system, each end-point of the prismatic joints sliding along the virtual object surface must be assigned a mass. The expected behavior is that these points closely follow the motion of the robot frames to which they are attached, such that they remain at the location on the object surface that minimizes the distance to the corresponding frame. To achieve this, a **low** mass value is typically used, rendering the dynamics of these sliding points negligible compared to the rest of the system.

However, setting the mass too low can lead to instability during control, as these lightweight points may undergo excessively high accelerations. Conversely, assigning a too high mass value produces undesirable effects: instead of the sliding points being passively guided by the finger motions, the dynamics invert and the fingers are instead attracted toward the initial positions of the joint end-points. This results in incorrect grasp configurations and failure to properly wrap the object.

This phenomenon is illustrated in Fig. 5.15. In subfigures (a) and (b), a low mass configuration leads to a correct steady-state grasp, with prismatic joint ends successfully aligning with the finger poses. In contrast, subfigures (c) and (d) show the result of increasing the mass: the fingers are now drawn toward the vertical initial position of the joints, preventing a proper wrapping of the object. A final mass value of 10 g was selected as it offers a good compromise: it is low enough to avoid influencing the global dynamics of the hand, thereby ensuring correct wrapping behavior, while still being sufficient to maintain control stability.



(a) Low mass - correct grasp (b) Low mass - joint visualization (c) High mass - failed grasp (d) High mass - joint visualization

Figure 5.15: Impact of the prismatic joint end mass. Subfigures (a) and (b) illustrate the correct configuration with low mass values, while (c) and (d) demonstrate the negative effect of excessive mass. Each pair shows the hand-object interaction and the corresponding prismatic joint positions.

Damping: base damping, decay rate, exponential damping, and thumb damping

Three levels of damping have been introduced into the system: a base damping applied globally to all motions, an exponential damping component that slows down the fingers as they approach the object, and a localized damping applied solely to the thumb in the medium wrap and power sphere grasps to prevent object ejection.

Increasing damping generally improves stability, reduces oscillations, and limits excessive velocities during interaction. However, this comes at the cost of a slower grasping phase. From a practical standpoint, a compromise must be reached to ensure efficient yet robust grasps. As an upper bound, it was decided that all grasps should complete within a maximum of 10 seconds, a conservative margin compared to human execution times.

Base damping The purpose of the base damping is to suppress residual oscillations in the system. It was tuned empirically by gradually increasing its value until all visible oscillations were eliminated, while ensuring that motions remained reasonably fast. A common value of 0.05 Ns/m was found to be effective across all three grasp types.

Exponential damping This non-linear damping term is used to synchronize finger arrival times near the object, improving grasp robustness. Two parameters define its behavior: the maximal damping coefficient c_{\max} , which sets the strength of the damping at contact (i.e., when the input coordinate reaches zero), and the decay rate α , which defines how fast the damping effect fades with distance.

Rather than tuning α directly, a more interpretable approach was adopted: choosing a distance d from the object at which the damping coefficient should reach 20% of its maximum. This leads to the relation:

$$\alpha = \frac{-\ln(0.2)}{d}.$$

Setting d too large results in excessive damping far from the object, while setting it too small undermines the effectiveness of the damping. Empirical testing showed that $d = 1$ cm yielded good synchronization for both the medium wrap and power sphere. For the lateral pinch, which involves closer interactions, a reduced value of $d = 0.5$ cm provided good responsiveness without compromising stability.

Regarding c_{\max} , values had to be sufficiently greater than the base damping to have a meaningful effect. A value of 0.1 Ns/m (twice the base damping) proved effective for the medium wrap and lateral pinch. For the power sphere, which induces the highest asymmetry in finger dynamics, a stronger coefficient of 0.2 Ns/m was required.

Thumb damping This damping is specific to the thumb and was introduced in the medium wrap and power sphere to mitigate a recurring issue: object ejection due to excessive thumb velocity at the point of contact. Once again, the strategy was to progressively increase the thumb damping until this effect was fully suppressed.

Figure 5.16 illustrates the contact time between the thumb and the virtual object for different damping coefficients. As can be seen, a damping value of 10 Ns/m leads to a synchronized finger contact and ultimately results in a robust grasp.

Spring stiffness: base value, inter-finger scaling, and inter-phalanx scaling

The next point to be addressed concerns the stiffness of the virtual springs that model the contact between the fingers and the object. This stiffness is governed by three parameters: a base stiffness value, a scaling factor for inter-finger stiffness, and a scaling factor for inter-phalangeal stiffness.

All previous experiments were conducted using a base stiffness value of 0.05 N/m. Consequently, the associated damping parameters were calibrated relative to this baseline. Increasing the base stiffness speeds up the grasping motion. However, it also reduces the relative effect of damping, which can lead to the reappearance of oscillations and improper finger synchronization. These instabilities often result in object ejection. For this reason, the initial value of 0.05 N/m was maintained and used as a reference throughout the tuning process.

The inter-finger and inter-phalangeal stiffness scaling factors play a central role in shaping the grasping behavior in the medium wrap and power sphere grasps. These scaling laws allow the modulation of stiffness between different fingers and between phalanges of the same finger, thereby influencing the contact sequence and enveloping motion. Due to the distinct mechanical nature of the medium wrap and power sphere grasps, different parameterizations were required in each case. Accordingly, the discussion of these scaling strategies will be separated and presented individually.



Figure 5.16: Snapshots of the contact moment between the thumb and the virtual object for increasing damping values.

Medium wrap Figure 5.17 shows the precise moment at which the fingers make initial contact with the object, for increasing values of the finger scaling factor f . A value of $f = 3.0$ implies that the thumb stiffness is three times higher than that of the little finger, whereas $f = 0.33$ produces the opposite effect. The intermediate case, $f = 1.0$, assigns the same stiffness to all fingers. As illustrated, varying this parameter directly influences which finger initiates contact with the object.

When applied to real objects, an interesting phenomenon was observed. For $f < 1$, the little finger typically reaches the object first. Since the thumb is generally already very close to the object at that moment, the object tends to be pushed simultaneously by the little finger and the thumb. However, as these two fingers are not aligned and do not face each other symmetrically, their combined action generates a torque, destabilizing the object and often resulting in a failed grasp. This effect can be understood in the extreme case shown in Fig. 5.17a, where the thumb and little finger induce a rotational moment on the object.

To avoid this effect, values of f greater than one were found to be more effective. In such configurations, the thumb and the first contacting finger are located opposite one another, and the forces they exert tend to balance, thereby canceling out the net torque. This stabilizing behavior is visible in Fig. 5.17e, where the object is effectively pinched between opposing fingers.

The core idea is as follows: the object should first be stabilized by a "pinching" action between the thumb and the index, before the remaining fingers progressively close in to secure the grasp. Ideally, this pinching should occur when the other fingers are already near the object, allowing them to constrain it and thereby improve robustness, especially when grasping non-cylindrical objects. In light of these observations, a value of $f = 1.5$ was selected as the final setting.

The second parameter under investigation is the phalanx scaling factor p . This parameter is of crucial importance, as it directly influences the emergence of the wrapping motion. Figure 5.18 presents snapshots of the grasp at the moment of contact for increasing values of p . A value of $p = 2.0$ implies that the stiffness of the proximal phalanges (closer to the palm) is twice that of the distal phalanges (closer to the fingertips), whereas $p = 0.5$ reverses this relationship.

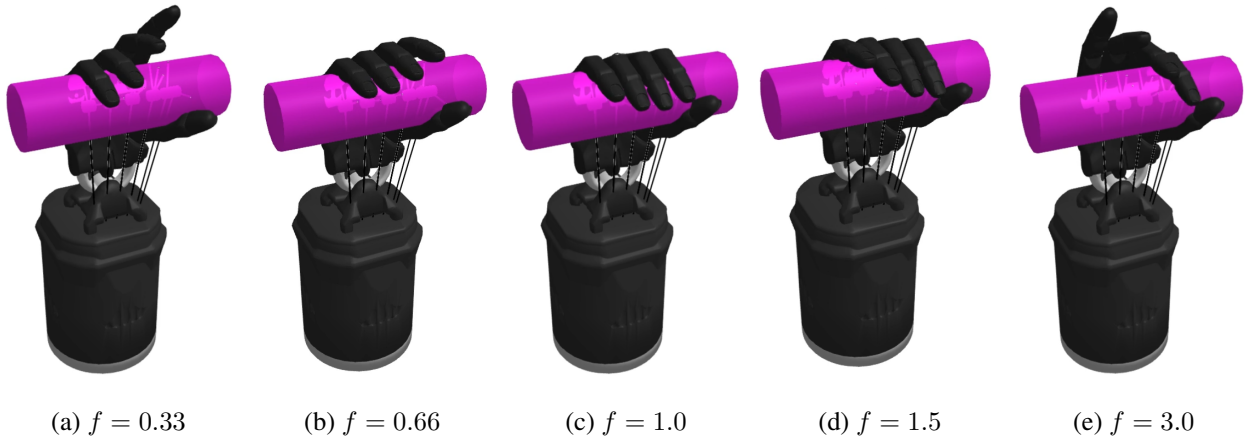


Figure 5.17: Medium wrap: Effect of the finger stiffness scaling factor f on contact sequence and grasp dynamics.

As shown in the figure, values of $p < 1$ tend to yield better enclosure of the object, while higher values ($p > 1$) degrade the grasp quality. This can be explained by considering the initial configuration: the proximal phalanges are already closer to the object than the other segments at the beginning of the motion. Assigning them a stronger stiffness further amplifies their influence during the early stages of the grasp, causing the palm to push prematurely on the object which often leads to grasp failure. This effect is particularly pronounced in the extreme case of $p = 10.0$, shown in Fig. 5.18e.

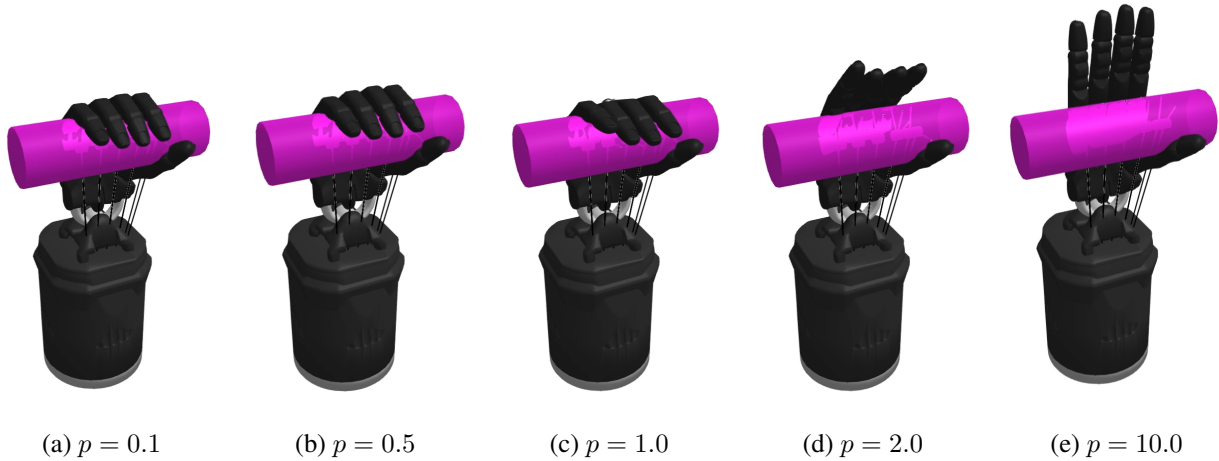


Figure 5.18: Medium wrap: Effect of the phalanx stiffness scaling factor p on contact sequence and grasp dynamics.

Conversely, when $p < 1$, the grasp tends to unfold more naturally. Regardless of the precise value of p , the hand closely approximates its final configuration at contact time, resulting in more stable and robust performance. For this reason, a value of $p = 0.5$ was selected for the final implementation, as it consistently produced reliable and effective grasping behavior.

Power sphere Varying the finger scaling factor f produces similar effects in the power sphere grasp as in the medium wrap: either the index or the little finger initiates contact prematurely, depending on the direction of the scaling. However, unlike the medium wrap, the power sphere does not allow for the "pinch" behavior previously described, due to the geometry of the object. In this case, both $f < 1$ and $f > 1$ result in the object being pushed

out of the hand, either by the little finger or by the index finger, as shown in Fig. 5.19.

The spherical shape requires a symmetric approach to ensure successful enclosure. Therefore, the optimal strategy is to synchronize the motion of all fingers so that they reach the object simultaneously. This behavior is achieved by setting the scaling factor to $f = 1.0$, ensuring uniform stiffness across all fingers and preventing any one finger from disrupting the grasping process.

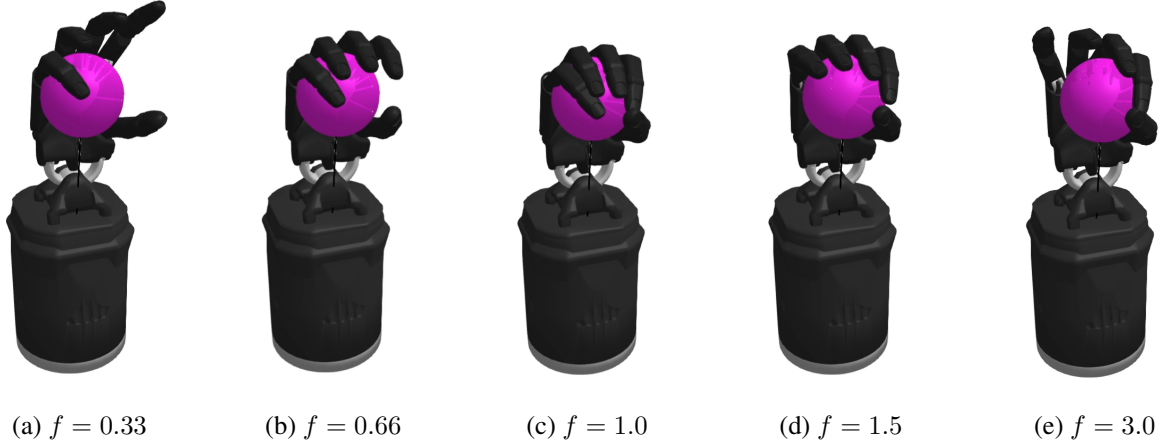


Figure 5.19: Power sphere: effect of the finger stiffness scaling factor f on contact sequence and grasp dynamics.

Regarding the phalanx scaling, the effect is identical to that observed in the medium wrap, as illustrated in Fig. 5.20. When the proximal phalanges are assigned a higher stiffness compared to the intermediate and distal ones, their initial proximity to the object is amplified, causing the entire hand to push the object away from the grasp. This effect is even more critical in the case of the power sphere, since the spherical object must be securely enclosed and compressed by the fingertips to avoid ejection.

To ensure that the fingertips initiate contact and effectively lead the grasping motion, the phalanx stiffness scaling factor was set to $p = 0.1$. This choice promotes a wrapping behavior that starts from the distal segments, enabling a more stable and controlled grasp of the spherical object.

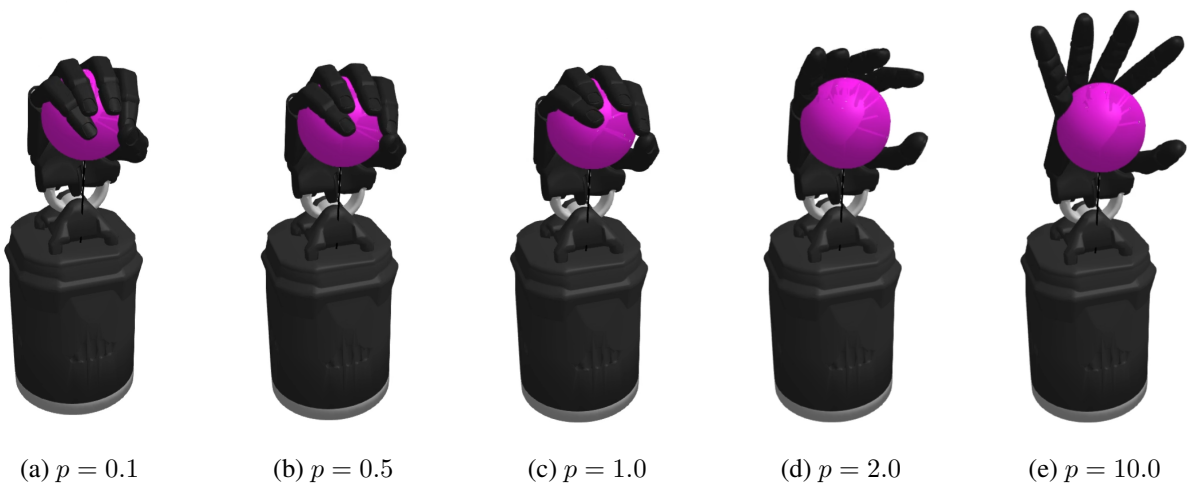


Figure 5.20: Power sphere: effect of the phalanx stiffness scaling factor p on contact sequence and grasp dynamics.

Less sensitive parameters

Some parameters have not been discussed in the previous sections due to their low sensitivity. These parameters typically exhibit a more binary behavior: they either fulfill their intended function or they do not. A representative example is the stiffness of the repulsive springs that emulate the virtual object. This stiffness must be considerably higher than any other stiffness in the system to accurately simulate the presence of a rigid object, but its precise value has little influence on the grasping dynamics.

The same reasoning applies to the springs used to space the fingers in the power sphere grasp, as well as those pulling the fingers toward the corners in the lateral pinch. Their values were chosen to ensure that the resulting grasp behaves as expected, but they are neither sensitive nor critical in the tuning process. For the sake of completeness, Appendix C lists the final values of all parameters used for the experiments.

5.3.2 Experimental results

The presentation of the experimental results follows the same structure as that introduced in Sec. 4.3.2. It begins with snapshots of the free-motion trajectory. Unlike the previous case, this trajectory is no longer fixed but instead varies depending on the geometry of the object to be grasped. Consequently, it is not feasible to illustrate all possible motion paths. Nonetheless, it has been decided to show the trajectory corresponding to nominal input parameters located at the center of the considered range to provide a representative example.

Subsequently, quantitative tests are conducted on the same set of objects previously described, using an identical evaluation protocol. Two scores are assigned to each grasp: one corresponding to the outcome of the shaking test, and the other to manual inspection.

Finally, a selection of representative grasping configurations is presented to offer qualitative insight into how natural the resulting postures are when compared to typical human grasps. As before, complete images of all final grasping poses for the three grasp types are included in Appendix B for the sake of completeness. All results obtained through this evaluation procedure are discussed in detail in the subsequent discussion section.

Medium wrap

As illustrated in the trajectory shown in Fig. 5.21, several of the aforementioned features are clearly observable: the unequal stiffness distribution across fingers is evident in the middle snapshot, while the influence of the virtual object is noticeable in the final configuration, where the hand remains partially open to accommodate the shape of a cylindrical object.



Figure 5.21: Object-centric medium wrap: snapshots of the free motion trajectory.

Object	Shaking test	Manual inspection
Medicine tube	5	5
3D printed cylinder	5	5
Metal can	5	5
Rice bottle	5	5
3D printed prism	5	5
Tea box	5	3
Glass	5	5
Beer bottle	5	5
Jungle speed	5	5

Table 5.1: Object-centric medium wrap: grasping results for the different objects under test.

The objects present in Fig. 5.22 have been specifically selected to highlight the improvements introduced by this method. Notably, the tea box and the medicine tube, which were previously difficult to grasp, are now securely and naturally held. Additionally, the bottle is no longer ejected during the grasping process. These results reflect the enhanced adaptability of the proposed approach to a broader range of object dimensions.



Figure 5.22: Object-centric medium wrap: the hand adapts to the object dimension, thereby replicating human natural behavior.

A final observation can be made regarding this specific grasp. Due to the vertical orientation of the virtual cylinder in the controller environment, the objects tend to be grasped in a more upright position by the Shadow Dexterous Hand. As a result, when released, many of the objects remain standing rather than falling. This outcome suggests the potential for implementing a first simple pick-and-place task.

Power sphere

Thanks to the presence of the virtual object, the final configuration of the power sphere grasp no longer exhibits finger interpenetration, as illustrated in Fig. 5.23. Instead, the trajectory naturally encloses the virtual ball, leading to a more realistic and physically consistent grasp posture.

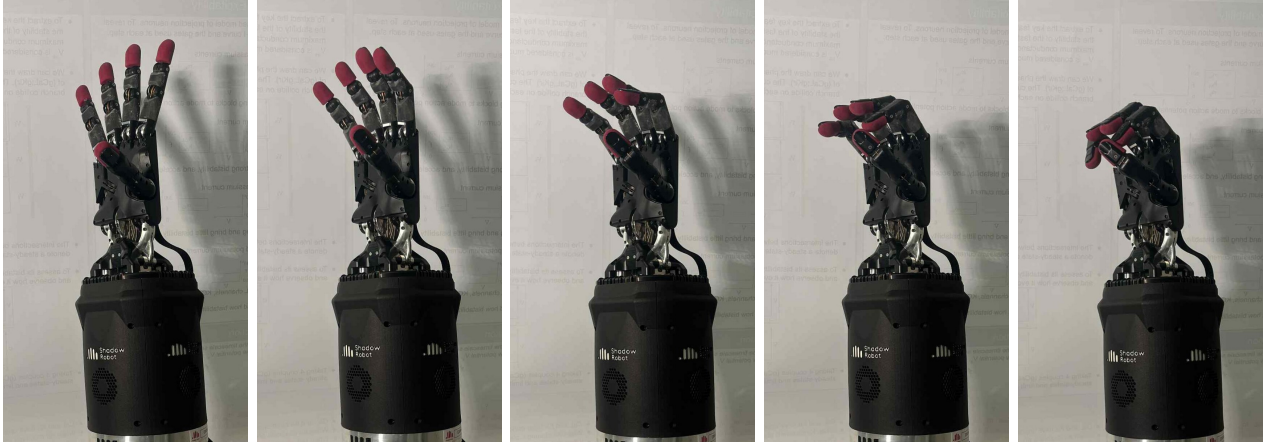


Figure 5.23: Object-centric power sphere: snapshots of the free motion trajectory.

Object	Shaking test	Manual inspection
Ping pong ball	5	4
Billiard ball	5	5
Tennis ball	5	5
Petanque ball	5	5
Speaker	3	2
Egg	5	5
Tupperware	5	5

Table 5.2: Object-centric power sphere: grasping results for the different objects under test.

Once again the focus has been set on the objects that could not be successfully grasped using the previously proposed strategy, namely the small ping pong ball and the larger speaker. In Fig. 5.24, both objects are now effectively and naturally grasped by the robotic hand.

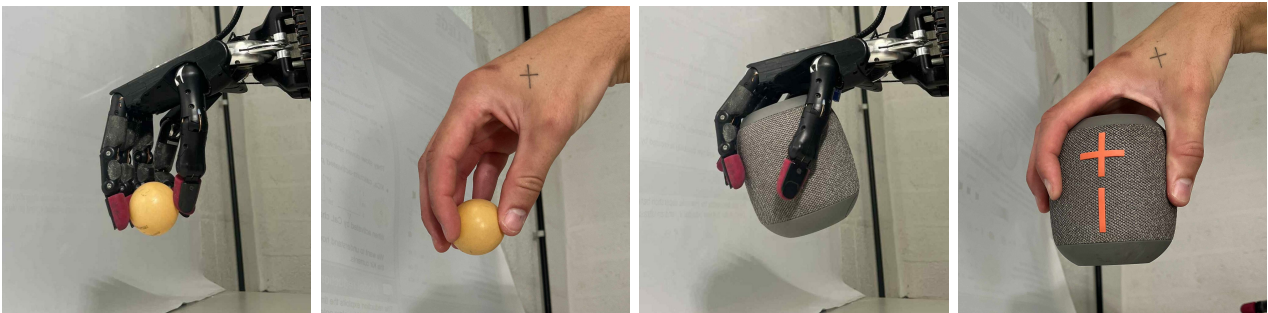


Figure 5.24: Object-centric power sphere: the ping pong ball and speaker can now be successfully grasp, with natural motions.

Lateral pinch

The scores corresponding to the object-centric lateral pinch are presented in Tab. 5.3. Although the improvement is less pronounced compared to the two previously discussed grasps, the overall performance remains better.

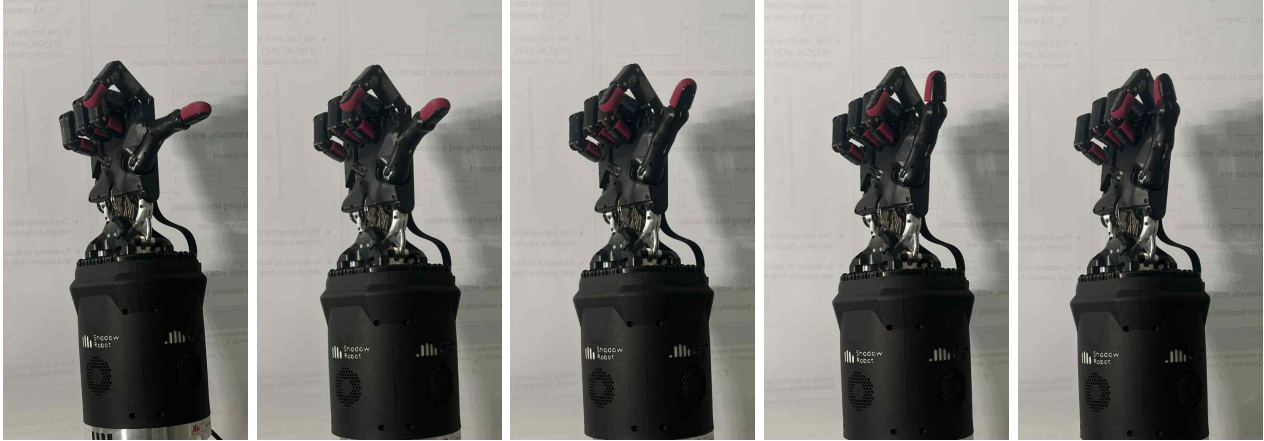


Figure 5.25: Object-centric lateral pinch: snapshots of the free motion trajectory.

Object	Shaking test	Manual inspection
Sheet pad	5	3
Dragon ball	5	3
Hunger games	4	2
Sapiens	4	2
Small wood piece	5	3
Perforated wood piece	5	3
Tupperware lid	5	3

Table 5.3: Object-centric lateral pinch: grasping results for the different objects under test.

In Fig. 5.26, it can be observed that the index finger adapts to the object width by extending in the case of the sheet pad and compressing for the wooden piece. This adaptive behavior may contribute to the improved performance of the object-centric policy compared to the previous approach. Notably, the sheet pad and the wooden piece are the two objects for which performance gains were observed, and they also represent the extreme cases in terms of object width within the tested set.



Figure 5.26: Object-centric lateral pinch: the first finger naturally adapts to the object width, as humans would do.

Deformability test

An additional experiment was conducted to highlight another main advantage of the object-centric approach: the ability to modulate grasp firmness. By slightly adjusting the dimensions of the virtual object used as input,

the user can effectively control the tightness of the resulting grasp, therefore adapting to the level of fragility of the considered object.

To illustrate this property, a 9 cm diameter deformable foam ball was sequentially grasped using power spheres of decreasing diameters. The resulting steady-state configurations are shown in Fig. 5.27. As the input diameter decreases, the foam ball becomes increasingly compressed, providing a clear visual indication of grasp firmness. This result confirms that the object-centric strategy allows fine-tuning of the grasp to better accommodate the mechanical properties of the object.

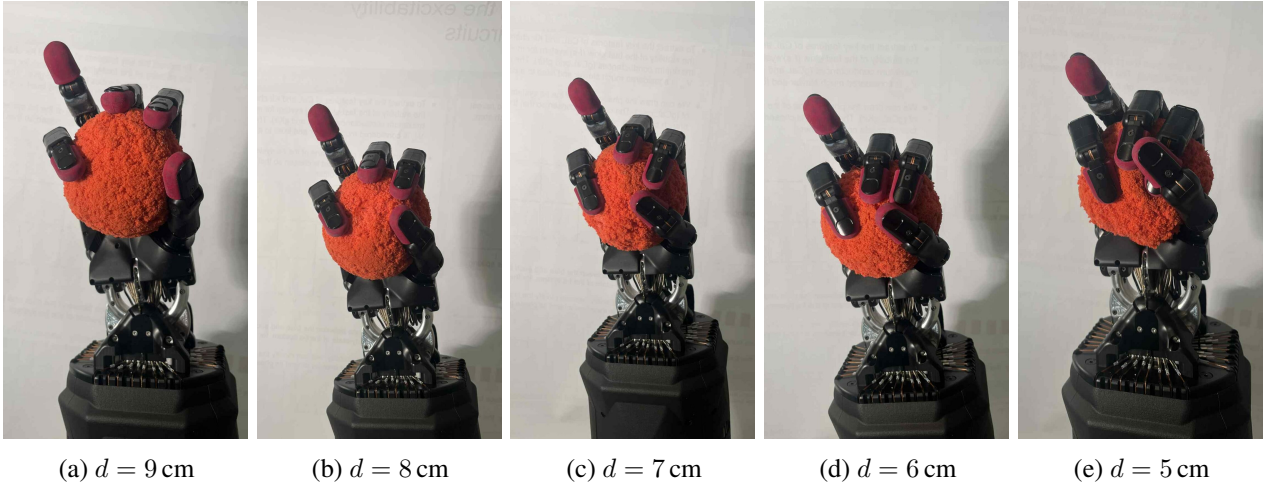


Figure 5.27: Deformability test: object-centric policy allows grasp firmness control.

5.3.3 Discussion and perspectives

From the results presented above, one can conclude that this original object-centric method outperforms the previously considered grasping strategy. The first notable advantage lies in the robustness of the grasping policy. By equipping the controller with a model of the object geometry, it becomes **adaptable** to a wider range of dimensions while maintaining natural, human-like motions. The controller is able to perform slight adjustments to the grasp, which was not possible with the previous static approach.

Another major advantage of the object-centric strategy is that it introduces a degree of freedom for the user to modulate the grasp. By altering the input dimensions of the virtual object, one can **control the firmness of the grasp**: providing slightly smaller dimensions results in a very firm grip, while slightly larger dimensions lead the hand to enclose the object without exerting compressive force. This property makes the approach suitable for manipulating fragile or deformable objects. Once again, an improvement over the previous method.

A strength of the previous strategy, highlighted in the preceding chapter, was the absence of a requirement for an **object model**. Indeed, obtaining an accurate model of an object can be challenging. However, as mentioned in the introduction, rough estimations of an object shape and dimensions can be easily obtained using a 3D camera. Not using this readily available and valuable information would represent a missed opportunity. The proposed method, on the other hand, makes effective use of such information without demanding excessive precision.

Finally, perhaps the most interesting insight is the following: although the three controllers execute completely different motions, their virtual mechanisms architectures share several common features. This reveals an underlying structure, characterized by the following elements:

- Contact is established through prismatic joints that slide along the object surface.
- Transient behaviors are corrected using exponential damping and thumb-specific damping.
- The object position relative to the hand is parametrized.

- There is a notion of "object span," which manifests as finger spacing in the power sphere, corner pulling in the lateral pinch, and could theoretically extend to the vertical alignment of the fingers in the medium wrap (if the cylindrical object were not assumed to have infinite length).

From these similarities, one can envision the emergence of a **general-purpose grasping architecture** that depends primarily on the object model, while all other parameters are derived automatically as a function of geometric data. Indeed, reducing the size of the virtual ball in the power sphere configuration results in a grasp that more closely resembles a "quadpod" or "precision sphere" grasp, according to Feix's taxonomy [15]. Yet, this behavior still emerges from the exact same virtual mechanism, with the only changes being the virtual object size and position.

In summary, this object-centric approach fully demonstrates the potential of the virtual mechanisms framework by introducing a method in which no specific motions are encoded: grasping instead emerges naturally from the underlying mechanical structure. This contrasts with many existing grasping techniques that tend to overfit specific object geometries and consequently struggle to generalize. Despite the absence of explicitly programmed motion features, the resulting control policy outperforms the previous baseline strategy, which served as a reference throughout this work. This finding opens up new avenues for robust and adaptable grasping solutions. Moreover, the recurring similarities observed across the three final controller implementations, despite not being planned from the outset, suggest that this approach may extend beyond the initial "template-based" inspiration and provide a more global and unified grasping strategy, which remains a missing component in the current robotics landscape.

Although the results obtained with both proposed approaches are very encouraging, it is important to acknowledge a source of bias in the testing procedure: the positioning of the objects within their environment. While the placement of the object relative to the hand was manually adjusted, this aspect is not considered critical, as one can reasonably assume the existence of an autonomous strategy that could reproduce similar configurations. In contrast, the position of the object with respect to its environment is a much more constrained factor, over which the robotic system typically has little to no control. During all experiments, objects were deliberately placed in favorable conditions to facilitate grasping: cylinders were oriented vertically, spheres were held on dedicated stands, and flat objects were positioned to stick out from the surface, thereby optimizing grasping performance. It is sure that, if the same policies had to be applied in real-world applications, such ideal conditions would rarely be encountered, which could seriously degrade the overall performance.

Chapter 6

Feedback integration

6.1 General idea

In Chapter 4, Fig. 4.5 introduced the concept of a **virtual hand**. This hand is connected to the virtual mechanisms and serves as a reference for the actual Shadow Dexterous Hand. However, as also shown in this figure, the position of the real robot is fed back to the controller, although this information has not been utilized so far. The objective of this chapter is to investigate whether the virtual and real hands could be interconnected within the controller environment using virtual springs and dampers. In doing so, the motion would no longer be purely feedforward: the interaction of the real hand with its environment, reflected through its position, would instead be fed back to the controller, thereby influencing the position of the virtual hand.

The introduction of such position-based feedback would enable the automatic control of grasp firmness. Indeed, once the hand comes into contact with an object, its motion is constrained by the object geometry, and it can no longer follow the virtual reference. This results in a mismatch between the commanded and actual hand positions (between the poses of the virtual and real hands). This mismatch would serve as the input to the virtual spring-damper pairs, thereby resisting further feedforward motion of the virtual hand. The final reference position would then correspond to an equilibrium between the virtual mechanisms pulling the hand toward the object center and the real hand being pulled toward the object surface. In this configuration, the grasp firmness is governed by the stiffness of the feedback loop: the higher the stiffness, the closer the reference is to the real hand position, and consequently, the smaller the torque exerted by the fingers.

6.2 Mismatch evaluation

Before introducing the aforementioned feedback strategy, a preliminary evaluation step is necessary. Indeed, the feedback spring/damper pair is intended to act on the discrepancy between the reference and the actual hand positions **caused by interaction with an object**. However, in practice, such a mismatch may also arise simply from the internal controller of the robot, which cannot track the reference perfectly and instantaneously. This section aims to quantify this "natural" mismatch by comparing the reference and actual hand positions during free motion, in the absence of any object. This evaluation is performed using the "ROS bridge" node (see Fig. 4.11), which listens to both state and command topics and logs the corresponding values into a spreadsheet for later visualization.

A first test was conducted with the hand completely still, by sending zero values to all joints. The results are shown in Fig. 6.1. It can be observed that the two wrist joints exhibit more noise than the others. Moreover, the overall mismatch already reaches approximately 0.03 rad, even in the absence of any motion.

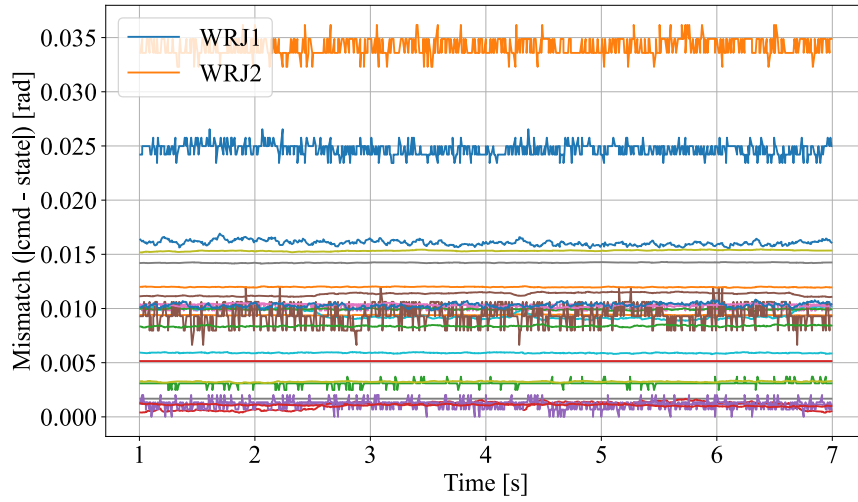


Figure 6.1: Mismatch values for the hand in a fully static configuration. A mismatch of up to 0.03 rad is observed.

Subsequently, a second test was carried out with the hand performing a medium wrap motion in free space. The resulting mismatch values are presented in Fig. 6.2. A few joints exhibit remarkably larger discrepancies, reaching up to 0.7 rad, while most others remain below 0.1 rad. These outliers correspond to the $J1$ and $J2$ joints of the four fingers on the palm, which are mechanically coupled. As such, they do not follow the reference commands individually. Instead of closing simultaneously, the first joint ($J2$) reaches its limit at 1.57 rad before the second joint ($J1$) begins to actuate. This sequential behavior is illustrated in Fig. 6.3, where two of these joints have been isolated.

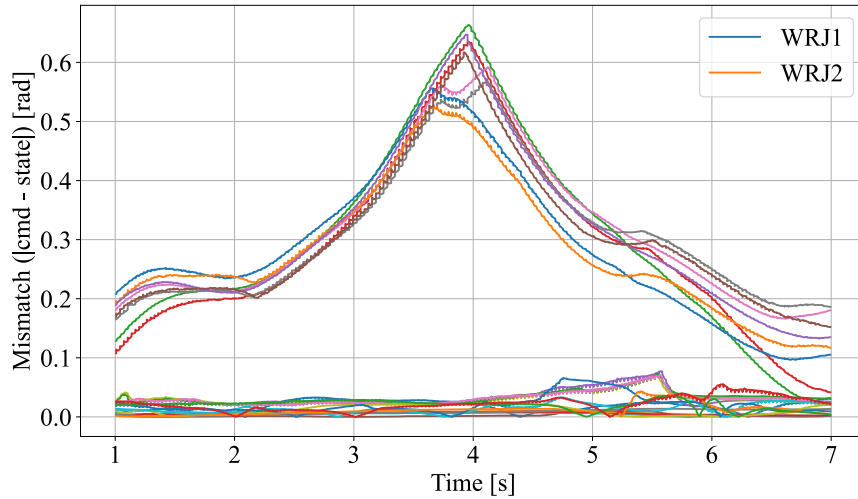
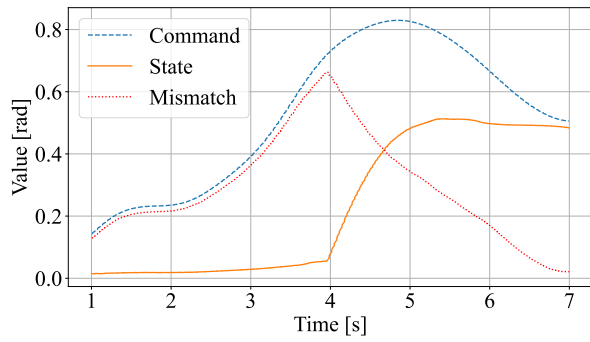
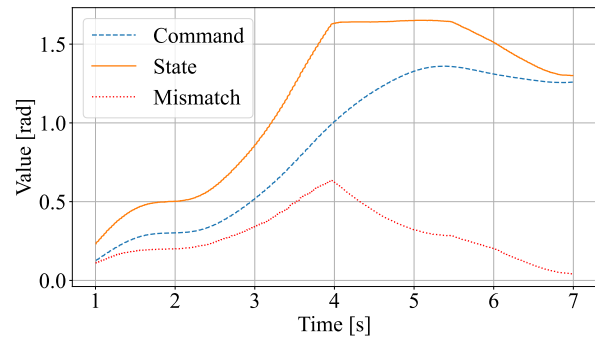


Figure 6.2: Mismatch values for the medium wrap motion in free space. Significant peaks are observed.

Due to this mechanical coupling, it is not meaningful to apply feedback at the individual joint level for these specific joints, as the observed mismatch is not caused by interaction with the environment but rather by their internal kinematic behavior. To address this, the same approach used for their control is adopted: instead of applying feedback on $J1$ and $J2$ separately, the feedback is applied to their sum, denoted as $J0$, which behaves in a deterministic manner. The mismatch is then reevaluated using this sum value, and the results are shown in Fig. 6.4.



(a) FFJ1



(b) FFJ2

Figure 6.3: Sequential actuation of mechanically coupled joints, instead of simultaneous motion.

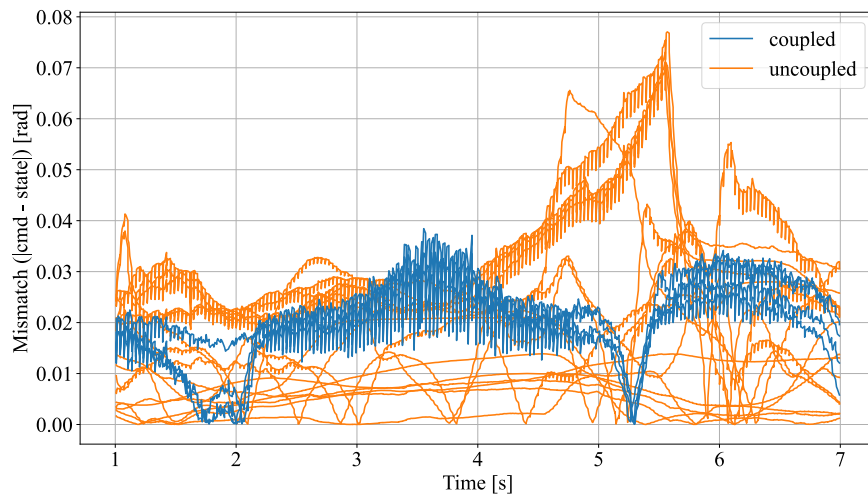
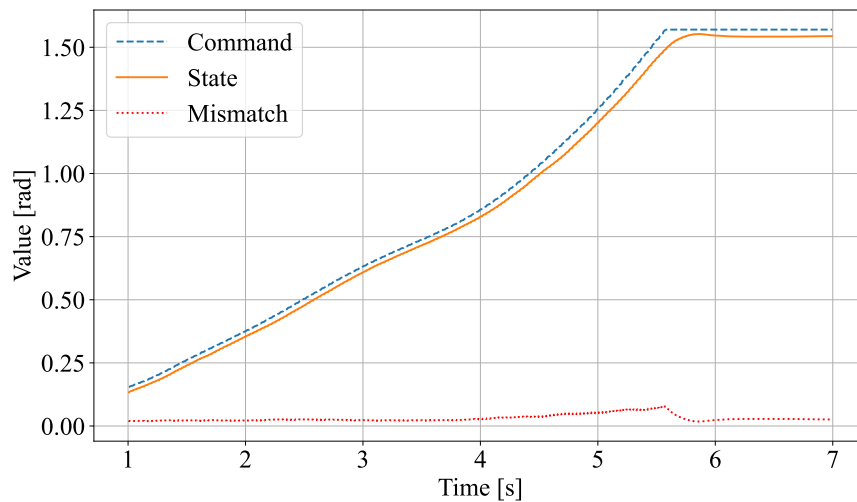
Figure 6.4: Mismatch values after grouping mechanically coupled joints using their summed value J_0 .

Figure 6.5: Observed mismatch peak on FFJ3 resulting from an accumulated delay between reference and actual motion.

With this modification, the mismatch remains below 0.08 rad. More precisely, only a few joints exhibit peaks above 0.05 rad. A detailed view of one such joint is shown in Fig. 6.5, where it is observed that the peak originates from a slight delay between the reference and the actual joint position, accumulated over the full motion. In light of this observation, such peaks can be considered exceptional and negligible for the overall feedback design. As a result, the "natural" mismatch level (i.e., the level not induced by interaction with objects) is finally set at 0.05 rad.

6.3 Implementation and discussion

With the natural mismatch level previously evaluated, a practical implementation of the virtual mechanism can now be realized. The connection between each robot joint and its corresponding virtual hand joint is achieved via **deadzone springs** (see Fig. 2.2b), which incorporate a deadband of 0.05 rad. This design ensures that no feedback is applied within the mismatch tolerance range. The stiffness coefficient k_f associated with these springs governs the intensity of the feedback, effectively modulating the grasp firmness and adaptability.

To prevent potential instabilities or oscillations introduced by these additional feedback elements, a complementary damping mechanism should, in principle, be included. However, since this section is intended solely as a proof of concept to explore the feasibility and potential benefits of this approach, only the steady-state grasping configuration is considered. For this reason, the architecture is deliberately kept as simple as possible, in order to make the influence of the feedback springs clearly observable and interpretable.

Given the exploratory nature of this experiment, only one grasp type was investigated, namely the medium wrap. The experiment consisted of grasping the rigid 3D-printed cylindrical object while progressively increasing the feedback stiffness k_f . Due to the object rigidity, directly and visually assessing the grasp firmness across different trials is challenging. To address this, a data-driven visualization strategy was employed: both the real and virtual hand joint states were recorded and stored in a spreadsheet. These data were subsequently used to animate both hand models, enabling a more intuitive analysis of the internal controller behavior and the influence of the feedback mechanism. The resulting steady-state configurations of both real and virtual hands for increasing k_f values are presented in Fig. 6.6.

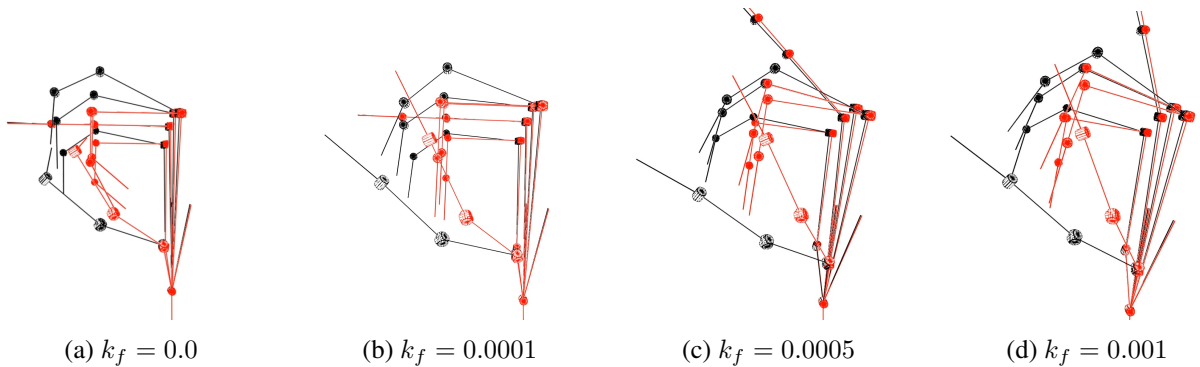


Figure 6.6: Real (black) vs virtual (red) hand final postures for increasing feedback stiffness.

Due to a malfunction in the *RF J0* joint, the original experimental protocol had to be slightly adapted to prevent potential hardware damage. This resulted in abnormal behavior of the ring finger throughout the trials. Except this limitation, the feedback mechanism appears to function as intended for the remaining fingers: while the configuration of the real robotic hand (displayed in black) remains nearly constant across the trials (because constrained by the rigid cylindrical object), the virtual hand configuration (in red) progressively approaches the real hand configuration. As a consequence, the grasp becomes increasingly loose, since the reduced discrepancy between the commanded and actual states results in lower motor effort.

In summary, incorporating feedback based on the robot actual position offers a simple yet effective method to enhance the adaptability of predefined grasping policies. This is particularly relevant for the initial open-loop

strategy, which was entirely blind to the object being grasped and only relied on sending predefined trajectories that penetrated the object geometry. Introducing a feedback mechanism allows for modulation of this effect, broadening the range of graspable objects to fragile ones.

Furthermore, the integration of this feedback strategy could also prove beneficial in the second object-centric approach. In scenarios where the grasped object substantially deviates from the simplified virtual template, the feedback loop would provide the flexibility needed to maintain successful and gentle grasping, adapting to unforeseen object features.

Despite the promising steady-state results, this approach also introduced several undesirable transient oscillations. As previously mentioned, these are primarily attributed to the lack of adequate damping mechanisms. However, the whole system still appears to be quite sensitive to parameter values. Consequently, fully integrating this feedback mechanism into the current grasping architectures would require extensive tuning to achieve the desired performance. For these reasons, this strategy has been retained as a proof-of-concept only, and no further evaluation was conducted regarding its practical impact on grasping performance with fragile objects.

Part III

Conclusion

Chapter 7

Conclusion and perspectives

7.1 Conclusion

The objective of this project was to implement grasping policies using virtual model control. The work began with a review of the current state-of-the-art methods for robotic grasping, followed by the introduction of the theoretical and practical foundations necessary for the developments presented throughout this thesis. A key observation was that most existing approaches tend to overfit the specific object being grasped, thereby lacking generalization. However, as emphasized in the introduction, the final goal of grasping with a dexterous robotic hand is usually to generalize to as many objects as possible.

The proposed strategy takes the opposite view by implementing a generic grasping controller aimed at generalizing across a wide variety of objects. This is achieved by drawing inspiration from human grasping taxonomies, thereby adopting a template-based approach: objects are categorized according to their shape and subsequently grasped using one of the three predefined grasp types.

A first implementation of these three grasp types was carried out by generating static, human-like grasping trajectories using the intuitive and modular design of the virtual mechanisms framework. This method focused solely on hand motion, without considering the geometry or presence of the object. Essentially, it recreated a one-degree-of-freedom synergy, relying on the hypothesis that human-optimized motions, developed over millions of years, should be inherently effective.

After extensive testing on a variety of objects, this approach proved to be surprisingly effective regardless of the precise object geometry, thus validating the generic nature of the template-based strategy. However, when the object dimensions were significantly increased or decreased, performance degraded. Indeed, humans naturally adapt their grasping motions to object size, something that was not captured in this initial approach.

To address this limitation, an original strategy termed **object-centric behavior shaping** was proposed. This method assumes the availability of sensory input (e.g., an RGB-D camera) to provide an approximate estimate of object dimensions. From this, a simplified object model is constructed within the controller environment, and virtual mechanisms are connected to this virtual object. The hand motion then emerges as a result of the virtual mechanical structure, rather than being entirely pre-encoded as in the previous strategy.

Experiments with this new method on the same set of objects revealed a major improvement: the hand is now capable of adapting to varying object sizes, thereby expanding the set of objects it can grasp. Moreover, the approach allows for control over grasp firmness by adjusting the input dimensions, thus generalizing the policy to include fragile and deformable objects.

Finally, a position-based feedback mechanism was introduced for both policies, and its feasibility was demonstrated.

Altogether, this project highlights the strength of the "**robust policies + raw approximations**" paradigm, which emerges as a promising direction in the field of robotics. Indeed, as robots become increasingly integrated into our daily lives, they will inevitably be required to operate under uncertainty, necessitating reliable controllers capable of accomplishing tasks despite imperfect information. In the present work, this philosophy was embodied by the use of generic motion templates that adapt to approximate object models. This strategy substantially broadened the range of objects that could be successfully grasped, leading to highly satisfactory practical outcomes. This template approach was ultimately discussed as a potential foundation for the development of more general-purpose grasping policies.

7.2 Perspectives

Several directions can be explored to extend this work and further enhance the grasping capabilities of the robotic hand. A first potential improvement concerns the **actuation** system. The virtual mechanisms framework was initially designed for torque-controlled robots. Applying it to a position-controlled platform inevitably limits compliance and adaptability. In the current setup, the stiffness parameters only influence the motion dynamics and have no impact on the forces applied by the hand. In contrast, a torque-controlled hand would allow these parameters to directly modulate contact forces, enhancing the grasp adaptability and naturalness.

Nevertheless, even without hardware modifications, many potential improvements lie on the side of **perception**, which is not fully exploited in this work.

In Chapter 5, the use of a 3D camera is proposed for estimating object shape and dimensions. A natural continuation of this work would be to implement this sensory input and evaluate whether it leads to satisfactory grasping outcomes.

Beyond basic dimensional estimation, a 3D point cloud from the camera could enable more accurate modeling of objects, going beyond the primitive shapes considered so far. This would require some adjustments to the software infrastructure to accommodate asymmetric or more complex geometries, but could lead to better grasping configurations, more tailored to the exact object geometry.

Another major enhancement would be to dynamically update the position of the virtual object in real-time using camera feedback. Until now, the virtual object is assumed to be fixed, and grasping succeeds only because the object remains sufficiently close to this static reference. For more delicate or sensitive grasps, a real-time adaptation of the virtual object position could mitigate the risk of object ejection, an issue encountered several times in this study.

In addition, the 3D camera could be exploited for more than just geometric estimation: visual perception could provide information about the nature of the object. For example, assessing whether it is fragile or deformable by employing computer vision algorithms would allow the grasp "firmness" previously identified as a degree of freedom controlled by the user to be automatically adjusted based on object properties.

Beyond visual inputs, other sensory inputs could further improve grasping performance. Notably, the Shadow Dexterous Hand is equipped with highly sensitive tactile sensors at the fingertips, which have not been utilized in the present work. These sensors provide an array of force measurements across the whole fingertip surface, allowing for rich feedback regarding the state of the grasp. Such data could be used to confirm contact, detect or prevent slippage, assess grasp firmness, and adapt the control policy accordingly. Incorporating these tactile signals opens up a wide range of possibilities for improving robustness and adaptability.

Finally, while this work has focused on grasping, grasping is merely the gateway to the broader field of **in-hand manipulation**, which presents even greater challenges. However, there is a promising path to extend the object-centric architecture toward manipulation: by moving the virtual object within the control environment, the same magic that allowed grasping to emerge from the virtual mechanical structure might once again apply. This time, the emergence would correspond to in-hand object motion naturally resulting from the new virtual object position. To be continued.

Part IV

Appendices

Appendix A

Human taxonomies

A.1 Cutkosky's taxonomy

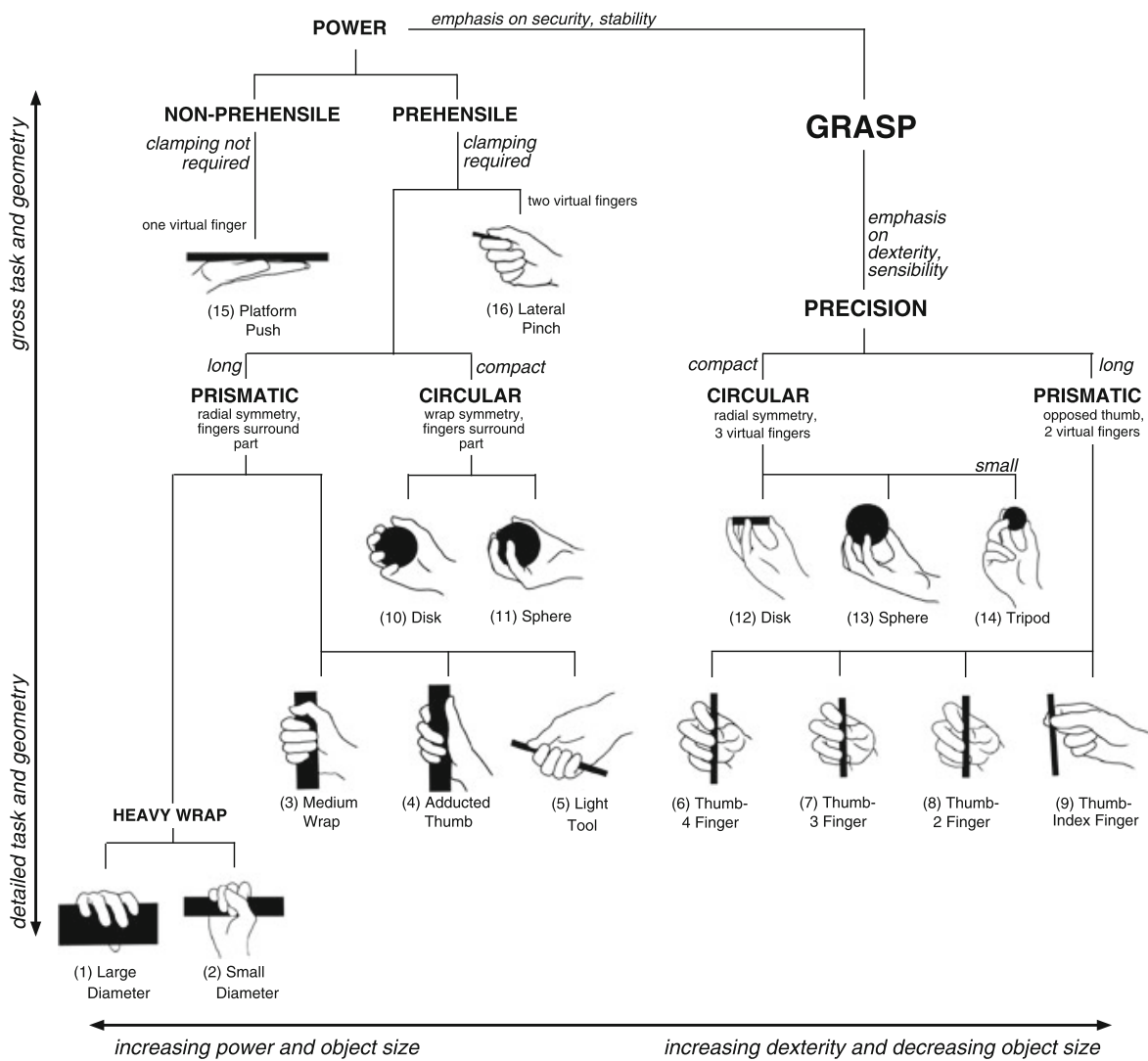


Figure A.1: Cutkosky's taxonomy [72], adapted from [22]

A.2 Feix's taxonomy


































Opp: VF:	Power						Intermediate			Precision				
	Palm		Pad				Side			Pad				Side
	3-5	2-5	2	2-3	2-4	2-5	2	3	3-4	2	2-3	2-4	2-5	3
Thumb Abducted		1: Large Diameter  2: Small Diameter  3: Medium Wrap  10: Power Disk  11: Power Sphere 	31: Ring 	28: Sphere Finger 	18: Extension Type  26: Sphere 4-Finger 	19: Distal Type 	23: Adduction Grip 		21: Tripod Variation 	9: Palmar Pinch  24: Tip Pinch  33: Inferior Pincer 	8: Prismatic 2 Finger  14: Tripod 	7: Prismatic 3 Finger  27: Quadpod 	6: Prismatic 4 Finger  12: Precision Disk  13: Precision Sphere 	20: Writing Tripod 
Thumb Adducted	17: Index Finger Extension 	4: Adducted Thumb  5: Light Tool  15: Fixed Hook  30: Palmar 					16: Lateral  29: Stick  32: Ventral 	25: Lateral Tripod 					22: Parallel Extension 	

Figure A.2: Feix's taxonomy [15]

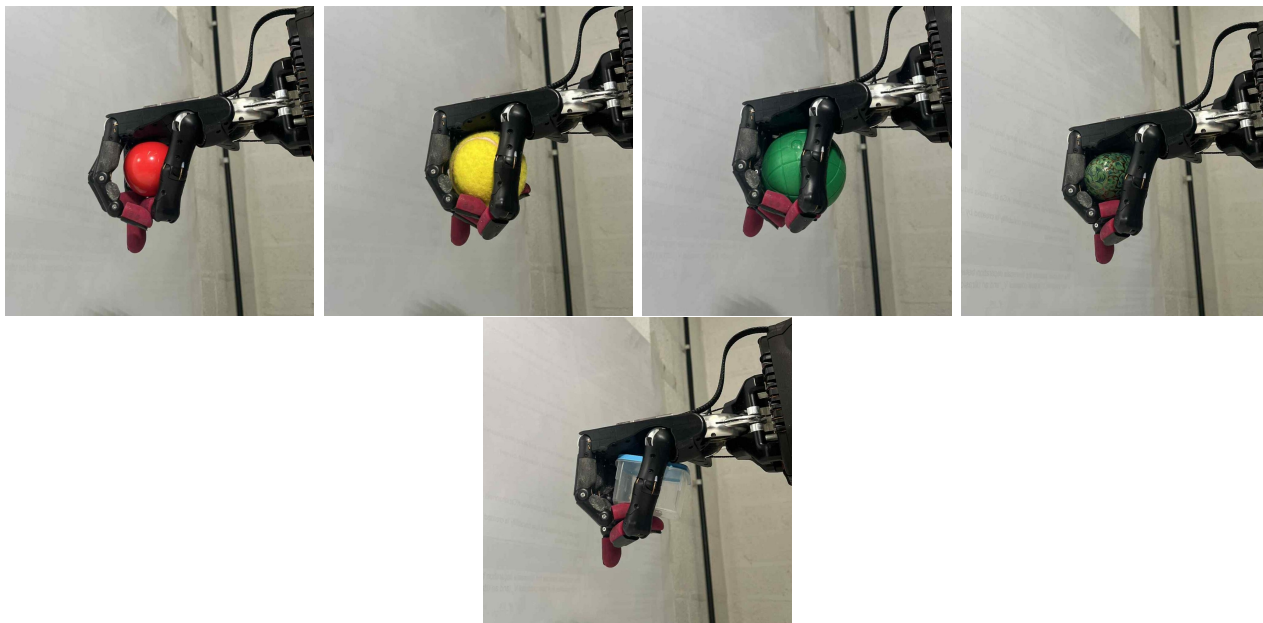
Appendix B

Steady state grasping configurations

B.1 Trajectory planners

Medium wrap



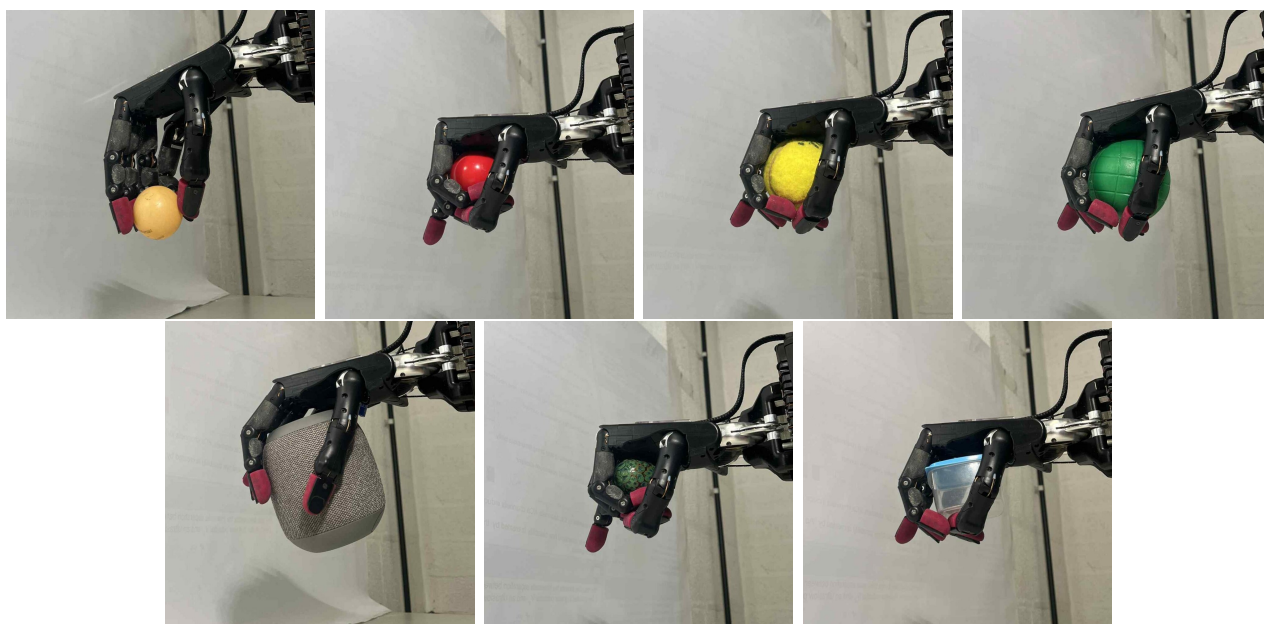
Power sphere**Lateral pinch**

B.2 Object-centric grasping

Medium wrap



Power sphere



Lateral pinch

Appendix C

Final parameters values

Medium wrap

Parameter	Description	Value
MEDIUM WRAP		
k_{vo}	Virtual object stiffness	5.0 N/m
c_{vo}	Virtual object damping	5.0 Ns/m
m	Prismatic joint end mass	0.01 kg
k_b	Base stiffness	0.05 N/m
p	Phalanx scaling factor	0.5
f	Finger scaling factor	1.5
c_b	Base damping	0.05 Ns/m
c_{max}	Maximum exponential damping coefficient	0.1 Ns/m
α	Exponential damping decay rate	161
d	Associated 20%-distance	0.01 m
c_{th}	Additional thumb damping	10.0 Ns/m

Table C.1: Object-centric medium wrap: final parameter values

Power sphere

Parameter	Description	Value
POWER SPHERE		
k_{vo}	Virtual object stiffness	5.0 N/m
c_{vo}	Virtual object damping	5.0 Ns/m
m	Prismatic joint end mass	0.01 kg
k_b	Base stiffness	0.05 N/m
p	Phalanx scaling factor	0.1
f	Finger scaling factor	1.0
k_{sp}	Finger spacing stiffness	0.001 Nm/rad
c_b	Base damping	0.05 Ns/m
c_{max}	Maximum exponential damping coefficient	0.2 Ns/m
α	Exponential damping decay rate	161
d	Associated 20%-distance	0.01 m
c_{th}	Additional thumb damping	10.0 Ns/m

Table C.2: Object-centric power sphere: final parameter values

Lateral pinch

Parameter	Description	Value
LATERAL PINCH		
k_{vo}	Virtual object stiffness	5.0 N/m
c_{vo}	Virtual object damping	5.0 Ns/m
m	Prismatic joint end mass	0.01 kg
k_x	X-axis fingers stiffness	0.01 N/m
k_{yz}	YZ-axis fingers stiffness	0.1 N/m
k_{pull}	Corners-pulling stiffness	0.1 N/m
c_b	Base damping	0.05 Ns/m
c_{max}	Maximum exponential damping coefficient	0.1 Ns/m
α	Exponential damping decay rate	460
d	Associated 20%-distance	0.005 m

Table C.3: Object-centric lateral pinch: final parameter values

Appendix D

Additional Materials

Video demonstration

A 4-minute video summarizing the main results of this thesis is available at the following link. It showcases representative hand trajectories, several successful and failed object grasps, as well as the shaking procedure.

https://youtu.be/Ng_mj9hBu9k

Codes

A GitHub repository containing the main files used to obtain the results presented throughout this project is available at the following link. It is accompanied by a README file that explains the purpose of each folder and file.

https://github.com/julienvanderheyden/TFE_Robotic_Grasping#

Bibliography

- [1] Chunmiao Yu and Peng Wang. Dexterous manipulation for multi-fingered robotic hands with reinforcement learning: A review. *Frontiers in Neurorobotics*, Volume 16 - 2022, 2022.
- [2] Aude Billard and Danica Kragic. Trends and challenges in robot manipulation. *Science*, 364(6446):eaat8414, 2019.
- [3] Hanbo Zhang, Jian Tang, Shiguang Sun, and Xuguang Lan. Robotic grasping from classical to modern: A survey, 2022.
- [4] Yinlin Li, Peng Wang, Rui Li, Mo Tao, Zhiyong Liu, and Hong Qiao. A survey of multifingered robotic manipulation: Biological results, structural evolvments, and learning methods. *Frontiers in Neurorobotics*, Volume 16 - 2022, 2022.
- [5] A. Sahbani, S. El-Khoury, and P. Bidaud. An overview of 3d object grasp synthesis algorithms. *Robotics and Autonomous Systems*, 60(3):326–336, 2012. Autonomous Grasping.
- [6] Zexiang Li, Ping Hsu, and Shankar Sastry. Grasping and coordinated manipulation by a multifingered robot hand. *The International Journal of Robotics Research*, 8(4):33–50, 1989.
- [7] Neville Hogan. Impedance control: An approach to manipulation. In *1984 American Control Conference*, pages 304–313, 1984.
- [8] Marco Santello, Martha Flanders, and John F. Soechting. Postural hand synergies for tool use. *Journal of Neuroscience*, 18(23):10105–10115, 1998.
- [9] M. G. Catalano, G. Grioli, A. Serio, E. Farnioli, C. Piazza, and A. Bicchi. Adaptive synergies for a humanoid robot hand. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 7–14, 2012.
- [10] Kilian Kleeberger, Richard Bormann, Werner Kraus, and Marco F Huber. A survey on Learning-Based robotic grasping. *Current Robotics Reports*, 1(4):239–249, December 2020.
- [11] Daniel Larby and Fulvio Forni. Optimal virtual model control for robotics: Design and tuning of passivity-based controllers, 2024.
- [12] Daniel Larby, Joshua Kershaw, Matthew Allen, and Fulvio Forni. Collaborative drill alignment in surgical robotics, 2025.
- [13] Yi Zhang, Daniel Larby, Fumiya Iida, and Fulvio Forni. Virtual model control for compliant reaching under uncertainties. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 795–801, 2024.
- [14] A.M. Okamura, N. Smaby, and M.R. Cutkosky. An overview of dexterous manipulation. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 1, pages 255–262 vol.1, 2000.

BIBLIOGRAPHY

- [15] Thomas Feix, Javier Romero, Heinz-Bodo Schmiedmayer, Aaron M. Dollar, and Danica Kragic. The grasp taxonomy of human grasp types. *IEEE Transactions on Human-Machine Systems*, 46(1):66–77, 2016.
- [16] W.S. Howard and V. Kumar. On the stability of grasped objects. *IEEE Transactions on Robotics and Automation*, 12(6):904–917, 1996.
- [17] H. Bruyninckx, S. Demey, and V. Kumar. Generalized stability of compliant grasps. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, volume 3, pages 2396–2402 vol.3, 1998.
- [18] Stephen L. Chiu. Task compatibility of manipulator postures. *The International Journal of Robotics Research*, 7(5):13–21, 1988.
- [19] Jihong Lee. A study on the manipulability measures for robot manipulators. In *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97*, volume 3, pages 1458–1465 vol.3, 1997.
- [20] Z. Li and S.S. Sastry. Task-oriented optimal grasping by multifingered robot hands. *IEEE Journal on Robotics and Automation*, 4(1):32–44, 1988.
- [21] G. Schlesinger. *Der mechanische Aufbau der künstlichen Glieder*, pages 321–661. Springer Berlin Heidelberg, Berlin, Heidelberg, 1919.
- [22] Mark R Cutkosky et al. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on robotics and automation*, 5(3):269–279, 1989.
- [23] Ryuta Ozawa and Kenji Tahara and. Grasp and dexterous manipulation of multi-fingered robotic hands: a review from a control view point. *Advanced Robotics*, 31(19-20):1030–1050, 2017.
- [24] J. K. Salisbury and B. Roth. Kinematic and force analysis of articulated mechanical hands. *Journal of Mechanisms, Transmissions, and Automation in Design*, 105(1):35–41, 03 1983.
- [25] Hooshang Hemami. Robot hands and the mechanics of manipulation (t. mason and j.k. salisbury, jr. (cambridge, ma: M.i.t., 1985) [book reviews]. *IEEE Transactions on Automatic Control*, 31(9):879–880, 1986.
- [26] Tsuneo Yoshikawa. Multifingered robot hands: Control for grasping and manipulation. *Annual Reviews in Control*, 34(2):199–208, 2010.
- [27] Maximo A. Roa and Raul Suarez. Grasp quality measures: Review and performance. *Autonomous Robots*, 38:65–88, 07 2014.
- [28] Yu Zheng. An efficient algorithm for a grasp quality measure. *Trans. Rob.*, 29(2):579–585, April 2013.
- [29] Z. Li and S.S. Sastry. Task-oriented optimal grasping by multifingered robot hands. *IEEE Journal on Robotics and Automation*, 4(1):32–44, 1988.
- [30] Ravi Balasubramanian, Ling Xu, Peter D. Brook, Joshua R. Smith, and Yoky Matsuoka. Human-guided grasp measures improve grasp robustness on physical robot. In *2010 IEEE International Conference on Robotics and Automation*, pages 2294–2301, 2010.
- [31] Javier Romero, Hedvig Kjellstrom, and Danica Kragic. Modeling and evaluation of human-to-robot mapping of grasps. In *2009 International Conference on Advanced Robotics*, pages 1–6, 2009.
- [32] Satoshi Ueki, Haruhisa Kawasaki, and Tetsuya Mouri. Adaptive coordinated control of multi-fingered robot hand. *Journal of Robotics and Mechatronics*, 21, 01 2010.
- [33] Hideo HANAFUSA and Haruhiko ASADA. Stable prehension of objects by the robot hand with elastic fingers. *Transactions of the Society of Instrument and Control Engineers*, 13(4):370–377, 1977.

BIBLIOGRAPHY

- [34] R. Fearing. Implementing a force strategy for object re-orientation. In *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, volume 3, pages 96–102, 1986.
- [35] J. Kenneth Salisbury and John J. Craig. Articulated hands: Force control and kinematic issues. *The International Journal of Robotics Research*, 1(1):4–17, 1982.
- [36] Thomas Wimbock, Christian Ott, and Gerd Hirzinger. Analysis and experimental evaluation of the intrinsically passive controller (ipc) for multifingered hands. In *2008 IEEE International Conference on Robotics and Automation*, pages 278–284, 2008.
- [37] Matei T. Ciocarlie, Corey Goldfeder, and Peter K. Allen. Dexterous grasping via eigengrasps : A low-dimensional approach to a high-complexity problem. In *Robotics: Science and systems manipulation workshop-sensing and adapting to the real world*, 2007.
- [38] Thomas A. Easton. On the normal use of reflexes: The hypothesis that reflexes form the basic language of the motor program permits simple, flexible specifications of voluntary movements and allows fruitful speculation. *American Scientist*, 60(5):591–599, 1972.
- [39] M. Gabiccini, A. Bicchi, D. Prattichizzo, and M. Malvezzi. On the role of hand synergies in the optimal choice of grasping forces. *Autonomous Robots*, 31(2):235–252, Oct 2011.
- [40] Antonio Bicchi, Marco Gabiccini, and Marco Santello. Modelling natural and artificial hands with synergies. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1581):3153–3161, November 2011.
- [41] Giorgio Grioli, Manuel Catalano, Emanuele Silvestro, Simone Tono, and Antonio Bicchi. Adaptive synergies: an approach to the design of under-actuated robotic hands. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1251–1256. IEEE, 2012.
- [42] Christopher Y. Brown and H. Harry Asada. Inter-finger coordination and postural synergies in robot hands via mechanical implementation of principal components analysis. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2877–2882, 2007.
- [43] Josiah B. Rosmarin and H. Harry Asada. Synergistic design of a humanoid hand with hybrid dc motor - sma array actuators embedded in the palm. In *2008 IEEE International Conference on Robotics and Automation*, pages 773–778, 2008.
- [44] Markus Grebenstein, Alin Albu-Schäffer, Thomas Bahls, Maxime Chalon, Oliver Eiberger, Werner Friedl, Robin Gruber, Sami Haddadin, Ulrich Hagn, Robert Haslinger, Hannes Höppner, Stefan Jörg, Mathias Nickl, Alexander Nothhelfer, Florian Petit, Josef Reill, Nikolaus Seitz, Thomas Wimböck, Sebastian Wolf, Tilo Wüsthoff, and Gerd Hirzinger. The dlr hand arm system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3175–3182, 2011.
- [45] Clement Gosselin, Frederic Pelletier, and Thierry Laliberte. An anthropomorphic underactuated robotic hand with 15 dofs and a single actuator. In *2008 IEEE International Conference on Robotics and Automation*, pages 749–754, 2008.
- [46] R. Ortega, A.J. Van Der Schaft, I. Mareels, and B. Maschke. Putting energy back in control. *IEEE Control Systems Magazine*, 21(2):18–33, 2001.
- [47] Jerry Pratt, Chee-Meng Chew, Ann Torres, Peter Dilworth, and Gill Pratt. Virtual model control: An intuitive approach for bipedal locomotion. *The International Journal of Robotics Research*, 20(2):129–143, 2001.
- [48] M.C. Smith. Synthesis of mechanical networks: the inerter. *IEEE Transactions on Automatic Control*, 47(10):1648–1662, 2002.

BIBLIOGRAPHY

- [49] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.
- [50] Daniel Larby and Fulvio Forni. A generalized approach to impedance control design for robotic minimally invasive surgery. *IFAC-PapersOnLine*, 56(2):8548–8555, 2023. 22nd IFAC World Congress.
- [51] Guangrong Chen, Sheng Guo, Bowen Hou, and Junzheng Wang. Virtual model control for quadruped robots. *IEEE Access*, 8:140736–140751, 2020.
- [52] D. Williams and O. Khatib. The virtual linkage: a model for internal forces in multi-grasp manipulation. In *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pages 1025–1030 vol.1, 1993.
- [53] Anna Kochan. Shadow delivers first hand. *Industrial Robot: An International Journal*, 32(1):15–16, Jan 2005.
- [54] Daniel Larby. VMRobotControl.jl. <https://github.com/Cambridge-Control-Lab/VMRobotControl>, 2024.
- [55] Shadow Robot Company. Shadow dexterous hand, 2024. Accessed: 2025-05-01.
- [56] OpenAI, Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation, 2019.
- [57] Yueh-Hua Wu, Jiashun Wang, and Xiaolong Wang. Learning generalizable dexterous manipulation from human grasp affordance, 2022.
- [58] J R Napier. The prehensile movements of the human hand. *J Bone Joint Surg Br*, 38-B(4):902–913, November 1956.
- [59] J M Landsmeer. Power grip and precision handling. *Ann Rheum Dis*, 21(2):164–170, June 1962.
- [60] Noriko Kamakura, Michiko Matsuo, Harumi Ishii, Fumiko Mitsuboshi, and Yoriko Miura. Patterns of static prehension in normal hands. *The American Journal of Occupational Therapy*, 34(7):437–445, 07 1980.
- [61] Angel P. del Pobil, Mario Prats, and Pedro Sanz. *Robot Physical Interaction through the combination of Vision, Tactile and Force Feedback*, volume 84. Springer, 01 2013.
- [62] Thea Iberall. Grasp planning from human prehension. In *IJCAI*, volume 87, pages 1153–1157. Citeseer, 1987.
- [63] Thea Iberall. Human prehension and dexterous robot hands. *The International Journal of Robotics Research*, 16(3):285–299, 1997.
- [64] Ian M. Bullock, Joshua Z. Zheng, Sara De La Rosa, Charlotte Guertler, and Aaron M. Dollar. Grasp frequency and usage in daily household and machine shop tasks. *IEEE Transactions on Haptics*, 6(3):296–308, 2013.
- [65] Ian M. Bullock, Thomas Feix, and Aaron M. Dollar. Finding small, versatile sets of human grasps to span common objects. In *2013 IEEE International Conference on Robotics and Automation*, pages 1068–1075, 2013.
- [66] M. Jeannerod and. The timing of natural prehension movements. *Journal of Motor Behavior*, 16(3):235–254, 1984. PMID: 15151851.

BIBLIOGRAPHY

- [67] Johanna Jahn, William E Janes, Maryam Saheb-Al-Zamani, Caitlin M Burbank, Justin M Brown, and Jack R Engsborg. Identification of three movement phases of the hand during lateral and pulp pinches using video motion capture. *Hand (N Y)*, 8(2):123–131, June 2013.
- [68] Bruce D Keefe, Pierre-Arthur Suray, and Simon J Watt. A margin for error in grasping: hand pre-shaping takes into account task-dependent changes in the probability of errors. *Exp Brain Res*, 237(4):1063–1075, February 2019.
- [69] Ashutosh Saxena, Justin Driemeyer, and Andrew Y. Ng. Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, 27(2):157–173, 2008.
- [70] A. Morales, E. Chinellato, A.H. Fagg, and A.P. del Pobil. Experimental prediction of the performance of grasp tasks from visual features. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 4, pages 3423–3428 vol.3, 2003.
- [71] Junggon Kim, Kunihiro Iwamoto, James Kuffner, Yasuhiro Ota, and Nancy Pollard. Physically based grasp quality evaluation under pose uncertainty. *Robotics, IEEE Transactions on*, 29:3258–3263, 04 2012.
- [72] Aaron M. Dollar. *Classifying Human Hand Use and the Activities of Daily Living*, pages 201–216. Springer International Publishing, Cham, 2014.