UNIVERSITÀ DEGLI STUDI DI SIENA

FACOLTÀ DI INGEGNERIA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE E SCIENZE MATEMATICHE

Data-Driven Design and Control Techniques for Soft Hands with Embedded Rigid Constraints

Valerio Bo

Ph.D Thesis in Information Engineering

Supervisor

Prof. Domenico Prattichizzo

Examination Commitee

Prof. Jùlia Borràs

Prof. Maria Cristina Valigi

Prof. Gianni Bianchini

Thesis reviewers

Prof. Jùlia Borràs

Prof. Maria Cristina Valigi

SIENA, 29/01/2023

To my family
and friends

"*A good aphorism is the wisdom of a whole book in a single sentence.*"
Theodor Fontane

Acknowledgements

I would like to express my sincere gratitude to all those who have contributed to the successful completion of my doctoral journey.
First and foremost, I am deeply thankful to my advisor, Prof. Domenico Prattichizzo, for their unwavering guidance, invaluable insights, and continuous support throughout the entire research process. Their expertise, patience, and encouragement have been instrumental in shaping the direction of this thesis.
I am grateful to Istituto Italiano di Tecnologia and University of Siena for providing the resources and environment conducive to research and academic exploration.
Special appreciation goes to my colleagues and fellow researchers, Enrico Turco and Maria Pozzi, who have shared their knowledge and provided a stimulating academic atmosphere. The collaborative spirit within our group has been a source of motivation.
My sincere thanks go to my friends and family for their unwavering encouragement, understanding, and patience throughout this challenging yet rewarding journey. Their emotional support has been my anchor.
Finally, I express my deepest gratitude to all those whose names may not appear here but who, in various ways, have contributed to the completion of this thesis.
Thank you all for being part of this significant chapter in my academic and personal growth.

Summary

Soft manipulation goes beyond traditional manipulation paradigms, introducing innovative strategies to facilitate the robust interaction between robotic hands and their sorroundings. The approach known as Environmental Constraint Exploitation (ECE), harnesses the features of the environment to enhance grasp robustness while minimizing planning efforts. The pivotal factor enabling ECE is the inherent compliance of soft hands, which allows them to comply and adapt to environmental features. The foundational step toward achieving stable grasps involves emulating how humans naturally exploit environmental constraints. However, the accurate detection of environmental constraints can be complex or not possible in many cases. A possible solution to deal with this issue is to embed, directly in the hand structure, purposefully designed parts that work themselves as "embedded constraints". This paradigm shift, for example, can be obtained by endowing a robotic gripper with an additional soft-rigid palm that can slide over and in between flat surfaces. This Thesis presents design and control techniques that allow the development and use of soft-rigid grippers with embedded constraints to effectively exploit environmental constraints and robustly grasp a variety of objects. Three distinct approaches are employed for the control: the first calculates the pre-grasp pose of the gripper through analytical optimization, the second relies on data-driven approaches based on human demonstrations, while the third is based on Deep Reinforcement Learning. Regarding the design, this Thesis shows that the application of optimization techniques combined with machine learning algorithms can support the automated design of novel gripperss, which can work efficiently. The proposed control and design strategies aim at broadening the application of robotic manipulation beyond industrial settings, enabling robotic systems to operate effectively in unstructured and previously unseen scenarios.

List of publications

International Journals

1. Turco, E., Bo, V., Pozzi, M., Rizzo, A., and Prattichizzo, D. (2021). Grasp planning with a soft reconfigurable gripper exploiting embedded and environmental constraints. IEEE Robotics and Automation Letters, 6(3), 5215-5222.

2. Bo, V., Turco, E., Pozzi, M., Malvezzi, M., and Prattichizzo, D. (2022). Automated design of embedded constraints for soft hands enabling new grasp strategies. IEEE Robotics and Automation Letters, 7(4), 11346-11353.

International Conferences with Peer Review

1. Turco, E., Bo, V., Tavassoli, M., Pozzi, M., and Prattichizzo, D. (2022, August). Learning Grasping Strategies for a Soft Non-Anthropomorphic Hand from Human Demonstrations. In 2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN) (pp. 934-941). IEEE.

2. Bo, V., Turco, E., Pozzi, M., Malvezzi, M., and Prattichizzo, D. (2023, April). A Data-Driven Topology Optimization Framework for Designing Robotic Grippers. In 2023 IEEE International Conference on Soft Robotics (RoboSoft) (pp. 1-6). IEEE.

3. Turco, E., Bo, V., Pozzi, M., Salvietti, G., Malvezzi, M., and Prattichizzo, D. (2024). Learning to Exploit Embedded and Environmental Constraints for Grasping in Clutter with a Soft-Rigid Gripper. Submitted to IEEE International Conference on Robotics and Automation (ICRA). IEEE.

4. Franco, L., Turco, E., Bo, V., Pozzi, M., Malvezzi, M., Prattichizzo, D. and Salvietti, G. (2024). The Double-Scoop Gripper: A Tendon-Driven Soft-Rigid End-Effector for Food Handling Exploiting Constraints in Narrow Spaces. Accepted to IEEE International Conference on Robotics and Automation (ICRA). IEEE.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"If you want a happy ending,
it just depends on where you close the book!"
Orson Welles

1.1 Soft Manipulation

Traditional robots are often limited by their rigidity. While they excel in tasks that require strength and precision, they struggle with tasks that demand compliance, adaptability, and interaction with delicate objects or environments. In industry, the application of rigid parallel grippers is justified by the presence of higly structured environments, in which the gripper only needs to interact with the target objects, accurately avoiding to enter in contact with other items. In unstructured scenarios, it becomes fundamental for the robot to deal with the possible interaction with the whole sorrounding while guaranteeing the integrity of the robot itself and the safety of humans if present [10]. Soft robotic hands stand at the forefront of modern robotics, embodying a transformative leap in automation technology [11]. Differently from the traditional paradigm of rigid mechanical structures, these robotic technologies have paved the way for a new era of adaptability, and versatility [10]. Soft grippers are designed to replicate flexibility of human hands, while integrating novel materials, bio-inspired design principles, and advanced actuation methods, including pneumatic, hydraulic or cable-driven systems with soft, pliable materials [12].

As the development of soft robotic hands continues to advance, their applications span a multitude of research fields, each harnessing their unique advantages. In the field of medicine and healthcare, soft robotic hands are proving to be game-changers. Soft robotic prosthetic hands (Fig. 1.1a) can improve the quality of life for amputees by providing them with more lifelike and functional alternatives to traditional prosthetics [13]. These hands are fitted with sensors and feedback systems [14] that allow users to experience a sense of touch, offering a profound advancement in the field of prosthetic technology. In agriculture and food processing, soft robotic hands are being employed in harvesting and packaging (Fig. 1.1b). Their ability to pick and sort objects without causing damage is revolutionizing the way fruits and vegetables are harvested and processed. The delicate touch of these hands ensures that the product remains intact, reducing waste and improving efficiency in the supply chain [15, 16]. Manufacturing and assembly are other sectors where soft robotic hands are making a significant impact. The adaptability of these hands is ideal for tasks that involve handling fragile components, such as assembling electronics (Fig. 1.1c) or quality control. Soft robotic hands can seamlessly adapt to varying product shapes and sizes, reducing the need for extensive reprogramming or custom

tooling, making them a cost-effective and efficient choice for manufacturers [17, 18].

(a) (b) (c)

Figure 1.1: Soft robotic hand applications. a) prosthetic hand Soft Hand Pro [1]. b) Soft robot for food handling [2]. c) Soft gripper dealing with cable manipulation [3].

Nowadays, soft robotic hands face several challenges that researchers and engineers are working to address [19]. The main challenges that are present can be summarized in control and programming, reconfigurability and scalability.

First, soft robotic systems are usually underactuated, making their control relatively easy. However, their modeling can be more complex to evaluate compared to their rigid counterparts, primarily due to their deformable structures [20]. Developing suitable models and sophisticated algorithms is essential to make these systems accessible and practical.

Being underactuated, soft hands are usually developed considering single-purpose design, often dictated by the actuation system. The presence of a single configuration can limit their application to high-varying scenarios.

Lastly, scaling up soft robotic systems for industrial use can be challenging. Research into scalable manufacturing methods and cost-effective solutions is required to expand their adoption in large-scale applications.

On the other hand, soft grippers offer an array of distinct advantages that set them apart from their rigid counterparts. Especially deriving from their intrinsic compliance, they find large application for:

- Fragility management: The compliant nature of soft robotic hands enables them to interact gently with objects. In healthcare, soft robotic hands can be used for delicate surgical procedures, patient care, and handling of fragile medical instruments.
- Adaptability: Soft robotic hands can conform to a wide range of object shapes and sizes without the need for constant reprogramming or recalibration. Their deformable structure allow them to grasp and manipulate objects of varying consistencies, from rigid components to delicate, irregularly shaped items.
- Enhanced Safety: Soft robotic hands are inherently safer when working alongside humans, thanks to their compliant structure. In industries where human-robot collaboration is adopted, such as manufacturing and warehousing, the reduced risk of injury in case of accidental contact makes them an ideal choice. Human-robot interaction is becoming more common in such settings, and the safety aspect is paramount.

- Environmental Adaptation: The flexibility of soft robotic hands allows them to adapt to complex and unstructured environments. This adaptability can improve the capabilities of the hands during grasping or manipulation tasks, using the interaction between the hand, the object and the environment.

In the next section, we will explore the environmental adaptation and exploitation more in detail.

1.2 Environmental Constraint Exploitation

Soft manipulation offers an enhanced approach to effectively handle unknown environments. In contrast to rigid manipulation, it allows for interaction with the environment, consequently enhancing the grasping capabilities when complex scenarios are present. Inspired by how humans deal with certain objects, soft grippers introduce a novel paradigm of robotic manipulation that involves *working with the environment* rather than manipulating avoiding *it* [21].

The physical limitations presented by objects within the environment are seen as potential ways for achieving optimal object manipulation [22]. Leveraging these opportunities, often referred to as environmental constraints, enables robust grasping and manipulation in dynamic, open, and highly unpredictable scenarios.

To effectively exploit the presence of environmental constraints, a unified approach of planning, design, and control algorithms is necessary. Together with perception, these aspects are fundamental to exploit the gripper-object-environment interactions in an optimal way.

Regarding the planning, traditional grasp planners are based on the evaluation of the contact points between the gripper and the object, and on the forces exerted by the robotic fingers. Nevertheless, this approach does not consider the presence of the environmental constraints, which are usually treated as obstacles to be avoided during grasp planning. On the other side, grasp planners developed for soft manipulation are able to generate feasible grasps through the use of sensing and with less accurate contact models, mainly inspired by how humans grasp [23]. These planners consider grasp quality measures to evaluate the generated grasp, but they do not execute the grasp replicating the exact same contacts considered during the planning. This choice is also dictated by the actuation technologies most adopted for soft hands [24]. Indeed, most of these grippers are underactuated, guaranteeing simple control of the actuation system and usually an intrinsic compliance obtained through the passitivity of the joints. This characteristic is fundamental for exploiting the environmental constraints, reason why the design plays a fundamental role in this area.

Then, also perception is needed to detect the environmental constraints and consequently to enable the execution of grasping strategies that exploit them. Opposed to what has been done with rigid grippers, the detected environment is not avoided.Furthermore, the planner can reconfigure the underactuated soft hand to better comply with a certain scenario or reposition the environment to enable certain strategies that would not be accessible in another configuration.

Lastly, regarding the control part, it can be useful to understand that the environmental constraint can be considered a passive component to be exploited during grasping tasks. To optimally control the motion of the robot while interacting with the environment, a precise modeling of gripper-environment-object contacts should be needed. However, this interaction

can be complex given the unpredictability and the variety of contacts and deformations between the hand, the object and the environment. In this case, the best approach consists in developing a less accurate model, approximating real deformations and interaction forces. Thus, precise and optimal control strategies are needed to evaluate the exact disposition of the environment and reduce the uncertainty of the grasp generation.

While there are cases where the soft hand can grasp the target object directly by exploiting the constraining surface (Fig. 1.2a), the exploitation of environmental constraints becomes even more crucial when the object presents characteristics that make the grasping task especially difficult. For example, a flat object lying on a table is difficult to be directly grasped from its top face (Fig. 1.2b). Also when the object is very close to obstacles, the grasp can result in a failure. In these cases, the environmental constraint exploitation proposes to adopt different non-prehensile grasping strategies to re-position the object in a more feasible pose. The most known strategies are flipping, pushing, or sliding. However, depending on the environment, these strategies are not always applicable and it can be complex to move the object from its current position. For instance, a slide-to-edge grasp necessitates the presence of an edge in the workspace [25].

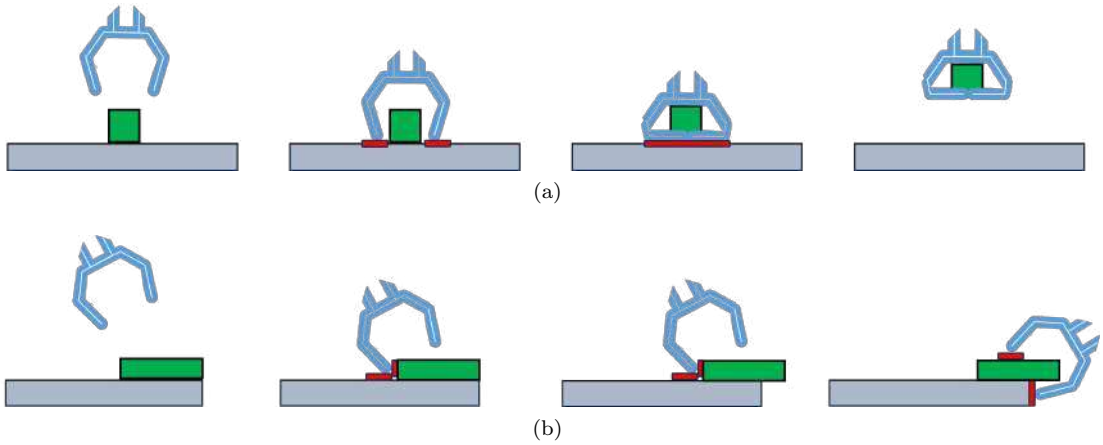


Figure 1.2: Grasps exploiting the environment. a) Sequence of grasping using the environment. b) Sequence of non-prehensile action, e.g., slide-to-edge grasp.

To solve this issue, to avoid the detection of the exploitable environment in the scene and to ease the burden of the control part, a further approach proposes to directly embed the environmental constraints in the hand itself, through the design of ad-hoc structures [6]. In this case we referred to them as embedded constraints. The embedded constraint adopted in [9], for instance, is a scoop-like structure designed as rigid parts and able to slide on different surfaces. Altough the embedded constraints are rigid, they can adapt to the environment through the use of compliant joints, making them able to perform in every situations.

The insertion of a "known" environment into a gripper can give different advantages:

- They can act as environmental constraints to support the grasp, when they are not present in the scenario. This feature also allows for simpler perception system, given that the detection of environmental constraints become unnecessary.

- They can simplify the control and sensing systems, given that the environment is embedded and there is no more necessity to detect it in the workspace. Moreover, because they are part of the hand, they can be designed as desired, integrated with sensors, and modeled as primarily suitable.
- They can be useful when dealing with fragile objects even if rigid, sliding between the environmental constraint and the object with a controlled strategy.

For these reasons, it can be stated that the application of embedded constraints can open a new paradigm in the environmental constraints exploitation, where the gripper-object-environment interactions become two pair: gripper-object and gripper-environment.

One of the first examples of embedded constraint has been presented in [9], where a novel soft gripper embodying the scoop has been presented, called SoftScoop Gripper (SSG). Nevertheless, the design proposed for the embedded constraint was prototypal and its control was not defined.

As detailed in the next section, the work described in this Thesis aims at developing optimal control and design algorithms to use embedded constraints to increase the grasping capabilities of soft grippers.

1.3 Contribution

The grasp planning and design techniques developed and described in this Thesis take into account and explicitly exploit the environmental and embedded constraints.

The first part of the Thesis is focused on how to generate suitable grasp configurations when dealing with embedded and environmental constraints, using optimization and machine learning algorithms. The second part deals with the design of these embedded constraints, through the combination of data-driven and optimization techniques. Lastly, the application of the embedded constraint into a real-world use-case scenario is presented.

1.3.1 Grasp planning with environmental and embedded constraints

The first main topic we will analyze consists in the development of grasp planning algorithms to produce optimal grasps dealing with embedded and environmental constraints. We adopted optimization and machine learning techniques to fully explore this topic.

First, we explored how to optimally execute grasps using the SSG and the embedded constraints, considering a prototypal non-optimal design of them [26]. We relied on a grasp planner that fully exploits the gripper features in combination with the environment around the manipulandum by solving an optimization problem. We performed a grasping strategy called *scoop grasp*, where the scoop slides onto a surface constraining the target object. In this work, we extend the concept of scooping grasps to include cases in which objects are constrained from two sides, and the scoop can exploit the surroundings by sliding on surfaces of different inclinations to fulfill the manipulation task. Through the experimental trials we demonstrated that in this setting a single motion is necessary to achieve the grasp, contrary to what would be needed when using other grippers in combination with non-prehensile actions.

The second contribution of this part focused on developing a grasp planning algorithm based on how humans would employ the embedded constraint to produce optimal grasps [27]. To

understand how humans would use this component, we proposed a data-driven methodology. Even if we simplified the hand-object-environment interactions through the embedding of the scoops, analytical approaches might fail due to the uncertainty deriving from them. We proposed to firstly observe how humans would approach the objects with this specific gripper, and then to generate grasping primitives based on the demonstrations. We developed a novel Learning from Demonstrations (LfD) method, which demonstrate to be faster in computing feasible grasps and needed few training data. We also considered different representations of the objects when extracting the grasping primitives to generalize the method. We applied this methodology to the SSG, but it can still be adopted for other soft grippers.

The last contribution regards the development of an algorithm for grasp planning in cluttered environment. This study was particularly useful to understand if non-prehensile actions are still necessary when exploiting the embedded constraints. We evaluated this hypothesis training and testing a state-of-the-art machine learning algorithms and analyzing the influence of the scoops on its learning process. In particular, we evaluated how a Deep Reinforcement Learning (DRL) based planner may exploit the mechanical intelligence of the SSG in a decluttering task. We compared two policies to test our theory. A policy consisted of only grasp actions, while the other contained also the pushing of the objects. While the baseline algorithm showed that the two-action policy was the best in terms of grasp success rate, in our case the best policy was the grasp-only. This approach demonstrated that the adoption of embedded constraints allows autonomous planning algorithm to use simpler grasp strategies while obtaining better results.

1.3.2 Data-driven design techniques for embedded constraints

The second part of the Thesis will focus on the design of the embedded constraints for achieving successful grasps. We propose design methods that incorporate data-driven techniques through the analysis of precedent grasping experiences. Thus, the design becomes intelligent, embodying information of precedent grasp plannings. Doing so, we are able to ease the grasp planning complexity because part of it has already been considered during the design.

First, we developed a data-driven framework to merge optimal design and planning of the scoops [28]. To do so, we performed several simulations to evaluate the optimal positioning and size of the embedded constraints on two different soft hands. In these simulations, we took into consideration the presence of the environment and we planned grasps orienting the scoop parallel to it, implicitely making the grasps generation for the execution simpler. We then clustered the optimal scoops to find a reasonable number of them, and we trained a Convolutional Neural Network (CNN) to recognize which of them to use depending on the target. We noticed that the optimal scoops outperformed with respect to the original and also when mounted on soft hands that natively do not consider their application.

Even if we optimally developed the embedded constraints implicitely considering the future grasp planning, their structural properties were not considered. Thus, we decided to developed a framework able to guarantee their mechanical stability while reducing the amount of material to be printed [29]. We evaluated the interaction between the scoop and different objects through simulation, and then we translated these force signals in instructions for Topology Optimization.Through extensive simulations we found that this analysis improved the gripper grasping success rate while also guaranteeing its resistance to mechanical stress. This methodology is

easily generalizable to any part of a given gripper.

Lastly, we decided to apply all the research on embedded constraints to develop a novel gripper able to fully exploit them. We designed this novel gripper composed of two soft fingers with a scoop placed at their fingertips. Through several experimental trials we showed that this novel gripper, called Double Scoop Gripper (DSG), was especially suitable for food handling. The reason for this particular behavior lies in the fact that the scoops can easily adapt to the environment and thus support the food instead of pinching or power-grasping it.

1.3.3 Thesis structure

The rest of the Thesis is structured as follows. Chapter 2 reviews the literature relative to the topics discussed in the rest of the Thesis. We will provide an overview on planning and design methods that exploit the environmental constraints with data-driven grasping strategies.

Chapter 3 presents the scoop grasp strategy. We developed this strategy such that the embedded constraint can adapt to a sliding surface to increase the probability of grasping the objects.

Chapter 4 reports a Learning from Demonstration approach to extract grasping primitives for soft non-anthropomorphic hands possessing embedded environmental constraints.

Chapter 5 describes a Deep Reinforcement Learning based planner for declutter that allows to evaluate and compare the ability of the scoop in complex scenarios.

Chapter 6 describes a data-driven framework for designing optimal embedded constraints considering the presence of restricted scenarios.

Chapter 7 introduces a pipeline to optimize the structure of the embedded constraints through the use of a topology optimization algorithm based on simulation data.

Chapter 8 presents the application of the embedded constraints to a real-world scenario through the design of a novel soft gripper that is comprising two scoops and is capable of grasping fragile food items.

Finally, Chapter 9 derives the conclusion of the Thesis together with future works.

Chapter 2

Related Works

"*Literature is my Utopia.*"
Helen Keller

This chapter provides an overview about the state of the art of the ideas and methods employed in the next chapters. Three main areas are covered: environmental constraint exploitation, grasp planning algorithms in robotic manipulation and design techniques for robotic hands.

2.1 Environmental Constraint Exploitation

Grasping objects in unstructured environments poses a considerable challenge for autonomous robots. Indeed, the uncertainty of the perception system can lead to dangerous interactions between the robot and its surroundings. An emerging trend to perform complex robotic manipulation involves leveraging external contacts or forces, such as the environment or gravity [30], avoiding non-prehensile actions. These strategies can be employed with both rigid and soft grippers. While several works present pivoting techniques exploiting gravity with completely rigid grippers [31, 32], it is complex and potentially dangerous to directly interact with the environment when employing this type of grippers. Still, it is conceivable to interact with the environment through impedance controllers or other techniques to make the robotic system compliant [33, 34, 35]. Another strategy to interact with the surroundings when adopting industrial, rigid grippers consists of leveraging the object-environment interactions by controlling the applied forces without making contact between the gripper and the environment. Tasks that necessitate manipulation in contact often incorporate the use of tools. These tools may be either firmly affixed to the robot arm or held by the robot for operational purposes [36, 37]. Examples of such manipulation are presented in [38, 39], where the authors utilize rigid grippers to push objects against the environment, enabling relative in-hand sliding motion. However, these strategies present high complexities and do not entirely prevent the possible damage caused by unwanted contact with the rigid gripper. Yet another possible concern arises from uncertainties encountered during interactions with the environment while utilizing a grasped tool. The act of holding a tool inherently introduces uncertainty regarding the precise location of the tooltip, necessitating compliance when in contact. Various methods exist to mitigate these uncertainties, particularly when sufficient information about the tool has been accurately measured [40].

Conversely, soft grippers can directly exploit the constraints present in the environment thanks to their intrinsic compliance [24, 41]. Thanks to their design, soft hands can interact with their surroundings, employing minimal sensing and control, reducing the required computational complexity and the risk of damage [10]. While some devices show compliance featuring passively

compliant joints [42, 43, 44], recent progress in soft robotics has led to the creation of devices entirely made of soft materials [45, 11, 46]. As Catalano *et al.*, demonstrated in [43], a hybrid configuration of deformable joints and rigid links can be effectively employed to simultaneously comply with environmental contact forces and adapt with the object to be grasped.

Despite these hardware advancements, a paradigm shift is essential for grasp planning and control strategies. Traditional grasp planners rely on precise hand and environment models [47, 48], using distinct finger-to-object contact points while avoiding interactions with the surrounding environment. However, these models become highly unpredictable in real-world scenarios, leading to poor grasping performance. Considering this, it is necessary to broaden classical grasp planning methods, or even surpass them, to effectively leverage the inherent characteristics of soft hands [49]. For instance, in studies such as [50], soft robotic hands are preconfigured and appropriately positioned above the object to achieve edge-grasps and top-grasps, respectively. Other approaches involve utilizing human demonstrations to establish motion primitives [51] or pre-grasp hand orientations [52] tailored for a soft anthropomorphic hand.

The presented grasp planners do not consider the exploitation of the environment, which, however, can lead to more robust grasps, as it happens in human manipulation [53, 54]. In Fig. 2.1, a summary of all the possible complex interactions that arise between a robotic hand, the grasped object and the environment are summarized in a scheme (on the left) and depicted in an example (on the right).

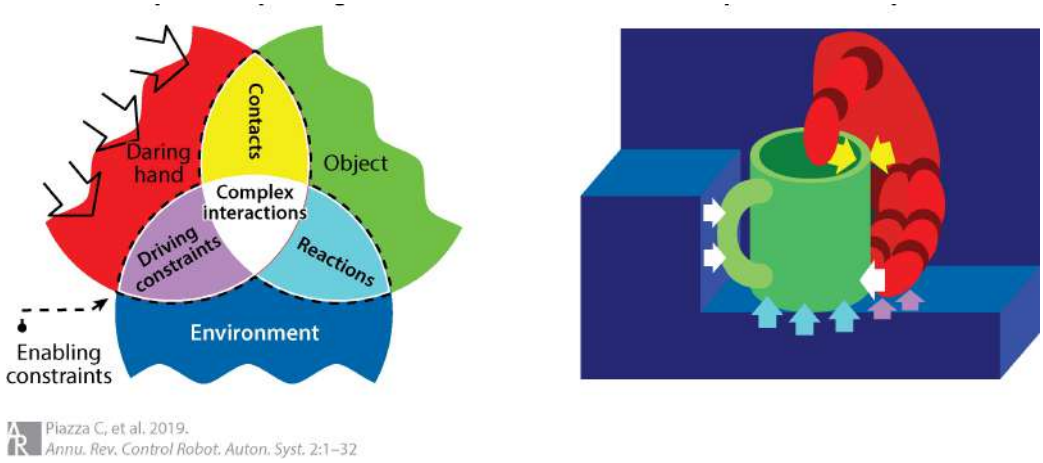


Figure 2.1: Paradigm and example of soft manipulation when the interaction with the environment is considered [4].

In [23, 55], a planner for automated grasp synthesis with a variable stiffness hand is presented, considering multiple factors such as the object neighborhood, environmental constraints (EC), and a vision confidence score. Yet, this approach is time-consuming, requiring the training of a stiffness decision tree to adjust hand compliance. Another method presented in [56] involves a computational framework for direct trajectory optimization, where exploiting environmental constraints is crucial for task accomplishment.

Human studies also highlight the tendency to exploit environmental features for grasping tasks, especially in the presence of uncertainties. Moreover, the Environmental Constraints

Exploitation (ECE) can potentially empower robotic systems with additional capabilities previously unattainable with rigid end-effectors. For instance, in numerous tasks, the handling of contact is not merely a matter of control, but it can be exploited, allowing robots to establish their location by touching the environment to execute tasks in the presence of uncertainties. The techniques for handling and leveraging contacts have also progressed to ensure that the computational demands remain manageable [57].

Various strategies for ECE include surface-constrained grasps from the top, where objects are caged within the hand, and fingers slide on the surface. For items on hard surfaces that cannot be grasped from the top, humans often use flip or slide-to-edge actions, requiring complex motions. Robotic implementations of these strategies pose challenges, but some studies have achieved flip-and-pinch grasps, slide-to-edge grasps, and scooping methods using compliant grippers [58, 25, 59]. Thanks to the effectiveness of the soft hands, these strategies can be performed with both open and closed-loop control. For instance, in [58], the authors demonstrated an open-loop control strategy to flip-and-pinch grasp with excellent results. On the other side, Bimbo *et al.*, adopted a hybrid force-velocity closed-loop controller to manage to pivot objects during a slide-to-edge grasp [25].

Models and measures assessing the grasp quality of soft hands have been developed, extending classic quasi-static models and synergy concepts to incorporate environmental interactions [60, 61, 62]. For instance, in [60], the authors formulate the hypothesis that the human capability to utilize environmental constraints can be characterized as a synergistic behavior in the creation of pre-grasp poses.

Lately, another trend has been rising, where the exploitation of environmental constraints is performed by embedding the constraint into the hands themselves. This novel idea can be groundbreaking in cases where the usual ECE strategies are not applicable (e.g., when needing a slide-to-edge grasp but the workspace does not present any exploitable edge). Several works demonstrated that the addition of these rigid parts, called scoops or embedded constraints, can enhance the effectiveness of soft hands when dealing with environmental constraints [5, 63, 64, 6, 9]. This kind of gripper is particularly beneficial when dealing with complex objects, such as food-handling tasks. These rigid parts can be mounted as opposing fingers or as nails at the finger's termination. For instance, the Quad-Spatula Gripper presented in [5] is a cable-driven gripper that can grasp food items without damaging, through four spatulas placed at the fingertips, as shown in Fig. 2.2a. On the other side, in [6], Babin *et al.*, developed a compliant parallel gripper that slides a scoop under the target object, exterting the minimum required grasping force, being able to hold very thin and fragile items. The developed gripper is shown in Fig. 2.2b.

The main similarity between the design of these different grippers lies in the intrinsic compliance of the joint linking the rigid scoop to the rest of the hand. These devices are adaptable, versatile, and robust thanks to passive compliance in their mechanical structures. Thus, we can say that part of the "intelligence" of the grasping system is already embodied in the hardware design, making the grasp planning phase more straightforward than in usual soft hands. Acknowledging the potential of designing soft robots with "embodied intelligence" [49], researchers have explored applications such as wrist extensions in constrained environments [65, 66]. These extensions simplify complex movements required for specific tasks, underscoring the adaptability and problem-solving capabilities achievable through innovative design approaches. At the same

(a) (b)

Figure 2.2: Soft hands integrating embedded constraints. a) Quad-Spatula Gripper [5]. b) gripper proposed by Babin [6].

time, the compliance of the hands allows the interaction with the environment. Another reason for embodying these components consists of the possibility of having the environmental constraint available whenever it is needed. Indeed, the scoops can act as environmental constraints to support the grasp when no constraint is present in the scenario.

2.2 Grasp planning algorithms in robotic manipulation

Grasp planning is a fundamental aspect of robotic manipulation that involves determining how a robot should position its end-effector to grasp an object securely and effectively. The ability to manipulate objects is crucial for robots to perform various tasks in environments designed for human interaction. Grasp planning algorithms enable robots to autonomously analyze and choose suitable grasps, ensuring successful and stable interactions with objects of varying shapes, sizes, and orientations.

The complexity of grasp planning arises from the diversity of objects and the uncertainties associated with real-world environments. Grasp planning algorithms must consider object geometry, surface properties, friction, and external disturbances to generate reliable and robust grasps. Over the years, researchers have developed a wide range of grasp planning techniques, leveraging advances in artificial intelligence, computer vision, and robotics. As highlighted by Sahbani *et al.*, two distinct approaches have been employed to address this challenge: analytic and empirical [67, 68].

Analytic grasp synthesis algorithms consider the physical and mechanical properties inherent in grasping [69, 70, 71]. Formulating an effective grasp solution tailored to a particular gripper and an object or a scene entails navigating a high-dimensional search. This complexity arises from the multitude of potential gripper-object poses, joint configurations, and contact conditions. Evaluating the merit of each grasp hypothesis involves considering various criteria,

including grasp stability, object or scene geometry, gripper geometry, kinematics, and suitability for a given manipulation task. Among these many criteria, a solution often recurring in literature is the adoption of grasp quality measures, i.e., indices that quantify the goodness of a grasp [72, 61]. These metrics for assessing grasp quality can be categorized into two groups based on the information utilized for evaluation. The first group pertains to the positions of contact points and includes grasp quality measures that exclusively take into account the properties of the object, friction constraints, as well as form and force closure conditions to quantify the effectiveness of the grasp [73, 74]. Conversely, the second group of quality measures includes those considering hand configuration to estimate the grasp quality. In this case, the quality index is assessed by evaluating the hand-object Jacobian matrix or through the hand joint positions [75, 76].

While the analytic approach is a viable solution for relatively simple hands [77], more complex and reconfigurable hands can face several problems with this method. The main issue of this technique lies in its computational complexity and incapacity to be applied in real-time [78]. Moreover, the need of accurate models for both the hand and the gripper makes this approach difficult to be employed in real-world scenarios [68].

On the other hand, data-driven grasp synthesis chooses the most appropriate hand configuration for the object and task using tools that consider prior grasp experience [79]. This approach selects the optimal hand configuration for a given object and task, employing techniques like learning from demonstration [80, 81, 82], neural networks [83, 84], or knowledge-based systems [85]. Data-driven models require a training phase that can be time-consuming, but after training the model, the execution of the algorithm is faster than in analytic models. For this reason, learning-based approaches can be adopted when online computation is needed.

Data-driven grasp synthesis methods can be divided into three main groups, depending on the learning process [86, 87]. The first method relies on labeled data derived from past grasp attempts and is often related to using deep or convolutional neural networks. The second is called Learning from Demonstration (LfD) and can be employed to record and analyze tasks performed by humans, extracting trajectories to be reproduced by the robot. Lastly, the third method consists of a trial and error process, where the robot autonomously updates some parameters under a set of specified rules to learn a specific task.

The first approach relies on labeled data, which can be supplied by humans or through a self-supervised process, where labels are automatically generated. This method can be based on camera images or specific object representations, such as voxels or meshes [88, 89], and it can produce candidate grasps or its grasp quality scores [90]. Typically, these approaches involve either sampling grasp candidates and ranking them with a neural network through discriminative techniques [91, 92, 93] or directly generating appropriate grasp poses with generative methods [94, 95, 96]. An example of sampling grasp candidates is provided in [51], where the authors labeled pre-grasp poses from human experience to train a deep neural network and predict how humans would grasp a specific object. This approach has also been exploited to evaluate the quality of a grasp through model-based techniques, as done in [97], where a grasp function evaluates the predictions provided by the network. The notable characteristic of the labeled approach is that similar architectures can be trained to perform significantly different tasks, i.e., labeled data can be employed to predict if objects in a scene are graspable, to generate grasping candidates, or to find the hand configuration to manipulate a specific object [98, 99, 100]. Other

interesting examples are provided in [101, 102]. The authors adopted similar Support Vector Machines for two different tasks in these two works. The first anticipates the graspability of an object within a scene, while the second identifies an appropriate grasp configuration, encompassing the approach vector, wrist orientation, and finger opening.

As it can be noticed, these methodologies are widespread and applicable to real-time grasp generation, but the labeled data needed to train these models can be massive. For instance, state-of-the-art Convolutional Neural Networks used for image data can contain more than 60 million parameters [103]. As demonstrated by Choi *et al.*, architectures that exploit more straightforward image representations can be trained with fewer parameters, but they still amount to 2 million [104]. Other more performant networks, such as the one proposed by Morrison *et al.*, exploit new types of architecture that are faster and need fewer parameters to be tuned [95]. Nevertheless, the authors required more than 50000 grasping label data to train their model. Since the Transfer Learning from simulation to the real world is far from attainable for complex cases like deformable objects or soft hands, collecting this data can be time-consuming with a real robot.

On the other hand, Learning from Demonstration (LfD) is a method that can be applied with few collected data. Moreover, it is beneficial to reproduce human behaviors that would be complex with a model-based approach [105]. Typically, the data is collected from humans executing some tasks, and later, the learning process is performed offline. This first process is usually done by teleoperating the robot [106, 107, 108] or through kinesthetic guidance, moving the robot manually across the relevant positions [109]. While the first method is preferable because of its higher maneuverability, it can be counterintuitive for a novel user, making the kinesthetic teaching more convenient for simple tasks. Many techniques have been employed in literature for the second step of the process. One of the most common is the Gaussian Mixture Model (GMM) together with Gaussian Mixture Regression (GMR) [110, 111, 112, 113]. This method consists of an analysis to parametrize the collected data as Gaussian distributions, and later, the output is provided through a regression of partial input data. Nevertheless, other techniques apply to this kind of problem, such as Dynamic Motion Primitives (DMP) [114], Decision Tree Regressors [52], Hidden Markov Models [115], and Probabilistic movement primitives (ProMP) [116]. Often, a combination of these methods is effective, exploiting the strengths of each. For instance, in [117], the authors propose a combination of Gaussian Mixture and Hidden Markov Models to address the generic resolution of extracting task constraints within a programming by demonstration framework and the subsequent challenge of extending the acquired knowledge to diverse contexts. This combination is advantageous when both trajectories and force/torque information are needed. In [118], the Hidden Semi-Markov Model is employed with Gaussian Mixture Regression to model motion as well as force data for in-contact tasks. While these methods are versatile and usually need few demonstrations, they are mainly targeted at reproducing human behavior, failing when complex tasks are presented.

For these tasks, e.g., grasping in a cluttered environment, learning through *trial and error* approaches were demonstrated to be outperforming. These methods have been employed for two applications: the first consists of tuning grasping strategies rather than choosing among a predetermined set of grasp candidates [119, 120], and the second consists of generating grasp candidates by setting a set of rules that the robot should obey [121]. One of the most widely adopted trial-and-error techniques is Reinforcement Learning (RL) [122]. RL requires an agent

to interact with the environment and, by updating specific parameters, can learn the best policy for a given task [87, 123, 124, 125].

Recently, RL has been combined with deep learning, leading to deep-RL algorithms that are vastly employed in complex robotic grasping scenarios like decluttering, bin picking, and pick and place in cluttered and narrow scenes [8, 126, 121, 127].

For instance, in [126], the authors propose a deep RL method for learning 6-DoF pick and place poses for objects in clutter (e.g., pick a mug among others and place it upright). To improve sample efficiency, the robot focuses on the most relevant parts of the scene for the performed task.

While the method presented in [126] exploits only prehensile actions for picking up target objects in clutter, other works focus on using non-prehensile motions (e.g., pushing, pulling, shifting) to remove the clutter and facilitate object grasping. This approach has been fully adopted in learning-based methods, including deep-RL [8, 127]. In [8], a synergy between pushing motions and grasping actions (called Visual Pushing-Grasping, VPG) was learned through self-supervised trial and error, considering pushing movements valid only if they lead to grasping success, whereas in [127], grasping in clutter was achieved using a combination of manipulation primitives like clamping or shifting to rearrange the surroundings. Typically, RL is employed with rigid industrial grippers, but some works exist in literature where soft hands have been adopted [123, 122].

While this methodology is effective for both simple and complex tasks and can be generalized for different purposes, it requires a lot of time to learn, and most of the time, the learning process can only be performed in simulation because of safety issues. To overcome this issue, many approaches propose using human demonstrations with reinforcement learning approaches to fasten the learning process [128, 129, 130]. For instance, Kroemer *et al.* [124], proposes a reinforcement learning method to finely tune a controller to reproduce an initial human demonstration. In [131], Gaussian Mixture Regression is employed with Reinforcement Learning to instruct a robot in an object grasping task. Additionally, reinforcement learning is utilized to explore novel approaches for task execution.

Even if RL techniques are becoming increasingly widely spread, they see few applications in the soft manipulation area [123, 122]. An engaging example is presented in [132], where the authors present a technique combining reinforcement learning and learning from demonstration to teach a robot how to manipulate objects with a soft anthropomorphic hand.

2.3 Design techniques for robotic hands

Creating capable robot hands stands out as one of robotics's most intricate and delicate aspects. The existing iterative and time-consuming process for gripper design needs to meet the requirements of agile manufacturing. In the work by Napier *et al.*[133], a standardized procedure for designing robotic hands is proposed, encompassing the following stages: *i*) kinematic architecture, *ii*) actuation technology, *iii*) actuation transmission, *iv*) type of sensors, *v*) materials, and *vi*) manufacturing method. The first can be interpreted as the number of Degrees of Freedom (DoF) and Degrees of Control (DoC) that the hand should comprise. From these parameters, the dexterity of the hand is defined [134]. Then, the actuation steps describe the force that the hand can exert and how it is transmitted to the fingertips [135, 136, 12]. The integration of

sensors can help perform closed-loop manipulation or object detection through tactile information [137]. Ultimately, the last two steps are essential to define the final product's stiffness and robustness [138].

Starting from these considerations, many researchers have pursued a design approach that aims to closely emulate the appearance and skills of human hands, trying to interpret and adapt the steps above. The first examples of taxonomies of human-centered grasps have been presented in [139, 140]. These methods involve sophisticated designs incorporating numerous actuators and sensors [141, 142, 143]. Even if several prototypes exist, this approach must be more effectively applicable. While achieving perfect structural and functional anthropomorphism, i.e.the resemblance to the human hand not just in appearance but also in movement and function, may be overly complex, recent innovations in hand design are focused on creating robust and easily programmable robotic hands capable of performing valuable subsets of the functions of human hands. Novel approaches have been proposed to increase relevance in real-world scenarios by simplifying the design process and the hands' structure. A simplification in design and control can radically reduce the complexity of the robotic system for several aspects [144, 145]. Nevertheless, this simplification process still needs to be improved to ease the design procedure. Typically, the process of designing a robotic hand is characterized by an iterative trial-and-error approach, which could extend over several weeks when executed by an expert. However, the agile market necessitates a rapid design process [146].

A viable solution may be found in a generalizable, hand-independent procedure to design complete hands or parts of them autonomously. Consequently, the automated design of robotic grippers represents a significant leap forward in developing highly adaptable grasping systems capable of manipulating diverse objects across varied environments. This transformative capability has been exemplified in various methodologies, showcasing the potential for rapid prototyping and synthesis of gripper structures tailored to specific manipulation strategies. In particular, the application of automated design methodologies proves instrumental in the evolving field of soft robotics, where the intricate relationship between morphology and functionality necessitates innovative approaches.

In the literature, several analytical methods have been presented. Honarpardaz *et al.*, propose to divide the automated design techniques into three categories: modular design, re-configurable design, and customized design [147]. First, modular design-based strategies typically simplify the workpiece's shape to basic geometries (e.g., cylinders, spheres, cubes, etc.). From a hand library, appropriate components are then selected [148, 149]. While these methods do not necessitate the production of fingers for each workpiece, an initial investment is required to create libraries, rendering them moderately cost-efficient. A framework for rapid prototyping of modular grippers, as proposed in [150, 151], exemplifies the potential for swift adaptation to different tasks. However, these methods encounter challenges in handling complex objects.

Re-configurable design approaches rely on algorithms that automatically determine finger configurations to grasp the workpiece and fulfill the task [152, 153, 154]. Despite being the most cost-efficient, they exhibit moderate design flexibility. Moreover, re-configurable methods demonstrate low design genericity and reliability since they neglect the workpiece's surface contour in the design process. In contrast, customized design approaches, distinct from the above-mentioned methods, provide a specific solution for each workpiece. These approaches are highly generic and capable of achieving both form and force closures. Nevertheless, they exhibit high

design flexibility and reliability, handling objects with any shape complexity and ensuring that the designed parts perfectly match the workpiece's shape. Despite their advantages, the need to design and produce separate components for each workpiece results in lower cost-efficiency for these methods [155, 156, 157, 158]. Among these methods, the Velasco and Newman [155] approach stands out as it can be integrated with grasp synthesizers and analyzers in CAD software. Although this method designs fingers to fully encompass the workpiece without relying on friction, practical feasibility concerns arise. For example, it may struggle to provide solutions with frictionless contacts that entirely fix the position of axisymmetric workpieces in space.

One notable technique employed in automated gripper design is Topology Optimization, focusing on maximizing system performance while minimizing material usage [159]. Topology optimization algorithms provide versatility in developing grippers tailored to specific requirements [160] or actuation technologies [161, 162, 163]. For instance, Liang *et al.*[160], proposes to design a micro-gripper through a two-step nonlinear topology optimization method to obtain a specific motion during the closure of the fingers. The two steps consist of a conventional linear topology optimization problem with a loose boundary condition and the subsequent nonlinear topology optimization based on the previous solution. Conversely, in [161], the authors present a method to systematically design soft robots, proposing to develop each gripper finger as a continuum, compliant mechanism to achieve its maximal bending deformation.

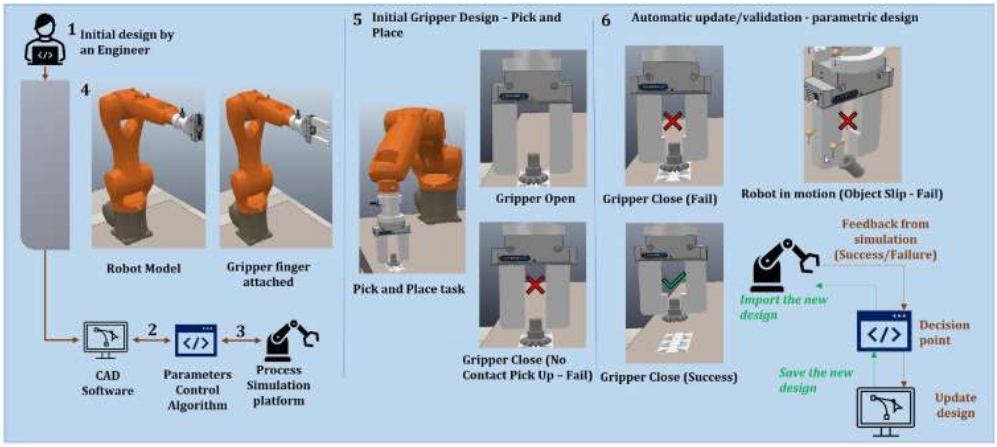


Figure 2.3: Schematic of the closed-loop framework for automated design of a parallel gripper proposed in [7].

Hiller and Lipson [164] utilized topological optimization to craft freeform soft robots capable of dynamic locomotion, showcasing the potential for innovation in the design of soft robots. Wang et al. [165] defined motion patterns for a soft gripper and employed topological optimization to design a gripper tailored to these patterns, showcasing the adaptability achievable through automated design. Similarly, Maloisel et al. [166] introduced a computational method for co-optimizing the routing of thin artificial muscle actuators and their actuation in hyperelastic soft robots, pushing the boundaries of what is achievable in soft robotic design. Similarly, for all these works, contacts between the gripper and objects are often modeled using pressure loading and friction tractions [165] or as constraints [167]. However, while reasonable for simpler

grippers, the a priori positioning of contacts may need to be more realistic for more complex, possibly soft, robots.

On the other hand, automated design methodologies based on data-driven methods offer a promising avenue for developing versatile grasping systems. In literature, few works propose to design exploiting data-driven considerations [168, 169, 170, 171, 7]. The approaches proposed in [168, 170] are computationally inexpensive and adaptable to various contacts and workpieces of varying shapes. However, they experience many issues when the process needs to be automated. Indeed, these methods necessitate simulations of operational procedures, which predominantly involve manual steps. Automating this process demands a substantial computational effort to identify the assembly process and simulate each task. Consequently, the efficiency of these approaches experiences a significant decline.

On the other side, an interesting example has been proposed by Deimel *et al.*, who developed a framework based on co-design methods where gripper structure and control strategy are simultaneously optimized and verified for feasibility through simulation [171]. Similarly, in [7], the authors propose a closed-loop system to test and optimize the fingers of a parallel gripper (diagram in Fig. 2.3).

This approach, as demonstrated in other works [172], harnesses simulation data for informed and efficient design iterations without the need for time-consuming and potentially unsafe actual experiments. Similarly, [173] introduces a procedure for synthesizing hand structures capable of executing specific in-hand manipulation strategies, underlining the adaptability achievable through automated design.

However, all these works propose to design the entire hand or gripper, while no partial design of just some components has been considered.

Part I

Grasp Planning for Embedded Constraints

Chapter 3

Scoop grasp strategy for exploitation of embedded and environmental constraints

I fanciulli trovano il tutto nel nulla,
gli uomini il nulla nel tutto.
Giacomo Leopardi

Grasping in unstructured environments requires highly adaptable and versatile hands together with strategies to exploit their features to get robust grasps. This chapter presents a method to grasp objects using a novel reconfigurable soft gripper with embodied constraints, the Soft ScoopGripper (SSG) [9]. The considered grasp strategy, called *scoop grasp*, exploits the SSG features to perform robust grasps. The embodied constraint, i.e., a scoop, is used to slide between the object and a flat surface (e.g., table, wall) in contact with it. The fingers are first configured according to the object geometry and then used to establish reliable contacts with it. Given the object to be grasped, the proposed grasp planner chooses the best configuration of the fingers and the scoop based on the object point cloud, and then suitably aligns the gripper to it. Several experimental trials in different scenarios confirmed the effectiveness of the proposed method.

The chapter is organized as follows. Sec. 3.1 defines the proposed strategy and relates it to literature works. The grasp planning algorithm is introduced in Sec. 3.2, and its experimental validation through multiple grasping trials is presented in Sec. 3.3. The discussion of the results is included in Sec. 3.4, and the conclusions are outlined in Sec. 3.5.

3.1 Introduction

In this chapter, we propose a method to plan grasps with the Soft ScoopGripper [9], a novel hand composed of two soft modular fingers and a tendon-driven scoop connected through a flexible hinge to the hand palm. In [9], Salvietti *et al.*, introduced a different point of view, embedding a constraint directly in the robotic hand. This work aims to explicitly take into account and exploit the gripper features. The considered grasp strategy is the so-called *scoop grasp* (Fig. 3.1), where the scoop adapts its orientation to the surface where it slides (e.g., table or wall), while the soft fingers establish reliable contacts with the object and cage it over the scoop.

Two main situations are considered: *i)* the object to be grasped is in contact with only one environmental constraint (e.g., table, Fig. 3.1a), and *ii)* the object touches two different surfaces (e.g., wall and table, Figs. 3.1b-c). The grasps depicted in Fig. 3.1 take inspiration from the so-called *surface-constrained grasp* and *slide-to-wall grasp* [53]. In [53], however, the two strategies are performed by a multifingered soft hand and require two very different motions. A scoop grasp, instead, can be performed on surfaces with different inclinations without substantially modifying the adopted strategy. This is possible because the exploitation of environmental features is facilitated by the presence of the actuated scoop, which offers an additional embedded constraint besides the palm.

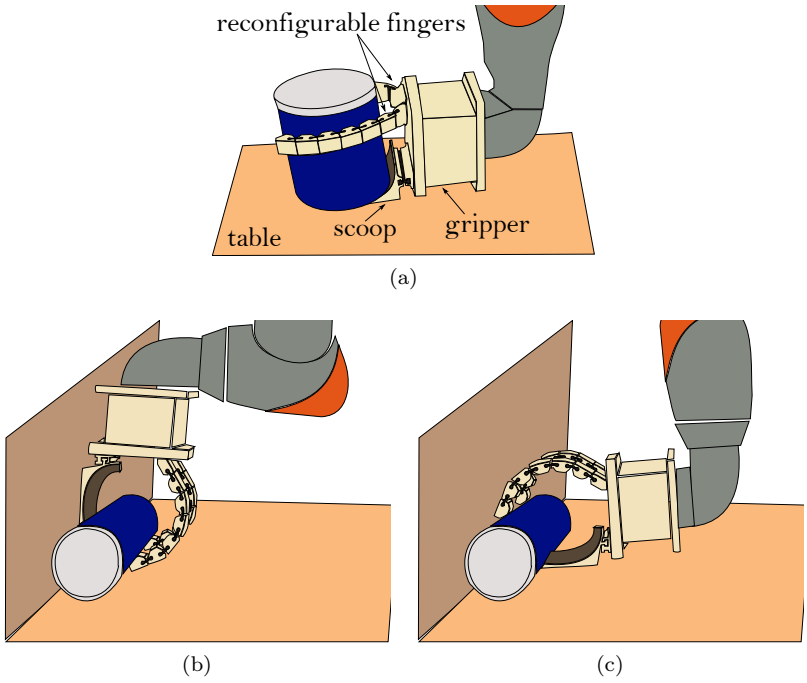


Figure 3.1: Scoop grasp: a soft gripper uses an embedded constraint (scoop) to slide over an environmental constraint (table or wall) and reach the object. Then, reconfigurable soft fingers grasp it.

Indeed, similarly to [21] and [53], the passive compliance of the SSG allows shape adaptation

between hand, object, and environment, and further improves grasp success. While there are works that consider completely rigid embedded constraints [174], passive adaptability in the gripper structure was found to be very important in other works presenting a "scooping grasp" strategy [175, 6].

We introduce an algorithm that, given the object point cloud, computes the best pre-grasp gripper configuration from which to start the scoop grasp strategy. The scoop grasp is then executed starting from the optimization results.

We extend the concept of scooping grasps to include cases in which objects are constrained from two sides and the scoop can either slide on horizontal surfaces, or on vertical ones. This level of flexibility is obtained thanks to a grasp planning algorithm that, based on the point cloud of the object, computes the desired pre-grasp configuration of the hand with respect to the object, rather than the exact position of contact points as done in [175].

In our method, pre-grasp parameters are computed by solving an optimization problem in simulation, then the real SSG is configured accordingly, and the hand-environment-object interaction is exploited to achieve the grasp. In this way, we leave the grasp execution to the gripper "intelligence", without attempting to exactly control the unpredictable interplay between hand, environment, and object. This principle is at the basis of recent methods for planning grasps with soft and underactuated robotic hands. Thanks to the presence of passive compliance in their mechanical structures, these devices are adaptable, versatile, and robust. Thus, we can say that part of the "intelligence" of the grasping system is already embodied in the hardware design, and it is not all demanded to the planning and control algorithms.

To summarize, in this work, we build on a method to achieve pre-grasp configurations that, combined with the hand adaptability, allow to get successful grasps. Additionally, we propose an algorithm that is suitable for hands with embedded constraints, like the SSG.

3.2 Materials and methods

3.2.1 The Soft ScoopGripper

The Soft ScoopGripper is a non-anthropomorphic underactuated robotic gripper composed of two soft fingers and a flat surface connected through a flexible hinge to the hand palm (the "scoop"). The fingers can be simultaneously flexed thanks to a tendon-driven differential system actuated by one motor. The scoop can be closed towards the fingers through a tendon-driven mechanism actuated by another motor. While in [9] this motion was obtained manually, the new version of the gripper used in this work has two additional degrees of actuation that allow to rotate the dovetail joints placed at fingers' bases up to 180°. Fig. 3.2 shows the new version of the gripper highlighting its components.

Similarly to multimodal grippers, the SSG can be reconfigured according to the object to grasp. However, in the SSG, the performed grasp is always the result of an enveloping movement (i.e., the fingers close towards each other, as in Fig. 3.1a, or both against the scoop, as in Figs. 3.1b-c), while, in multimodal grippers, different grasping modes, not necessarily based on closing motions (e.g., suction cups [176]), can be combined.

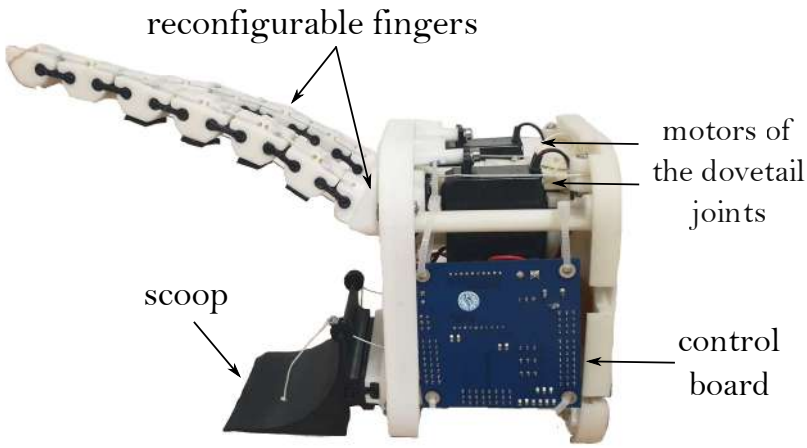


Figure 3.2: New version of the Soft ScoopGrippper.

3.2.2 Optimization problem

In this work, we present a method to choose the pre-grasp pose of the SSG based on an optimization algorithm. The vector $\mathbf{x}$ of decision variables contains the angles of the dovetail joints θ_R and θ_L, the distance between object and scoop centers d, the orientation of the scoop γ with respect to the object, and the inclination of the gripper α (see Fig. 3.3):

$$\mathbf{x} = [\theta_R, \theta_L, d, \gamma, \alpha]. \tag{3.1}$$

The values of γ vary in an interval whose end points depend on the portion of workspace that is reachable by the robot for a certain object pose.

The optimization problem is formulated as follows:

$$\underset{\mathbf{x}}{\text{maximize}} \quad GQI(\mathbf{x}) + \frac{A_{scoop}(\mathbf{x})}{A_{tot}}. \tag{3.2}$$

The main components of the cost function are a grasp quality index and the ratio between the area of the scoop occupied by the object (A_{scoop}) and the total area of the scoop (A_{tot}). In our simulations and experiments, we chose the Grasp Isotropy Index (GII)[2] [72] as grasp quality index.

While the GII is useful to evaluate the quality of the grasps, the second term tells us how much the objects are kept robustly, letting the scoop deal with the weight of the object. Thus, values of GII close to 1 translate into better grasps, while values of A_{scoop} close to A_{tot} give us insights on how much the object's weight is sustained by the scoop. Indeed, from a physical point of view, the more the object is lying on the scoop, the more the scoop can support the object's weight.

[2]The GII is the ratio between the minimum and the maximum singular values of the Grasp Matrix $\mathbf{G}$: $GII = \sigma_{min}(\mathbf{G})/\sigma_{max}(\mathbf{G})$. This index approaches 1 when the grasp is isotropic (optimal case), and falls to zero when the grasp is close to a singular configuration. For more details on the computation of the Grasp Matrix the reader can refer to [177].

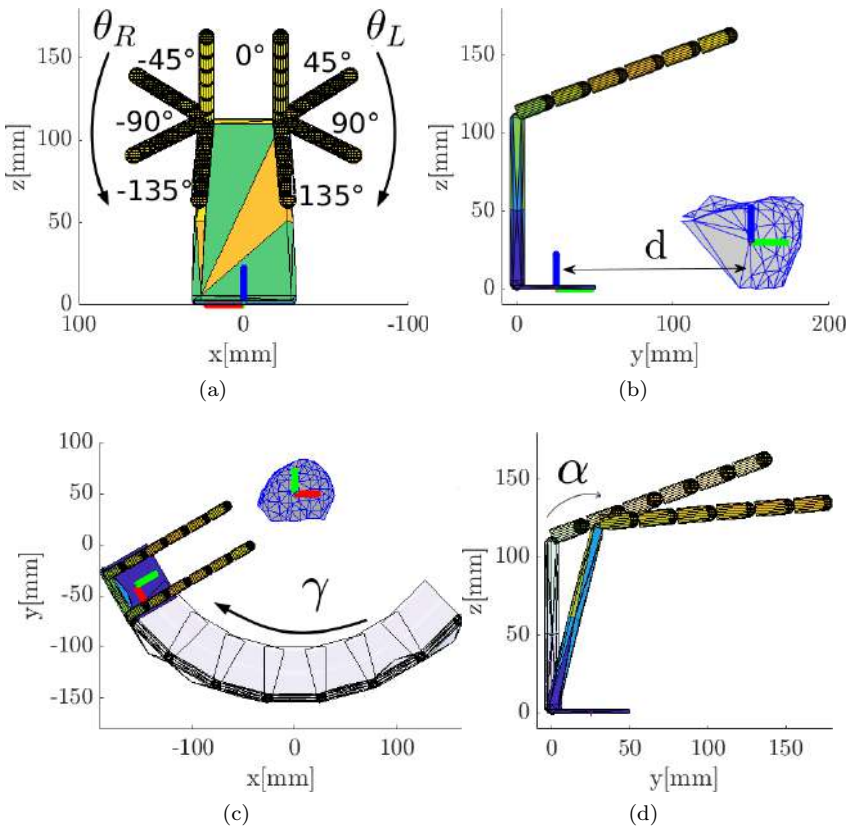


Figure 3.3: Decision variables of the optimization problem in Eq. (3.2): a) θ_R and θ_L express rotations of right and left dovetail joints, respectively. Joint axes point towards the palm of the hand; b) d is the distance between the object center and the scoop center; c) γ represents a rotation referred to the scoop frame around the *z-axis* of the object reference frame; d) α represents a rotation referred to the scoop frame around its *x-axis*. The axes are represented in red (x), green (y), and blue (z).

In principle, however, it is possible to choose any other metric that can be computed based on the knowledge of the quasi-static model of the grasp [177]. The only precaution to be taken consists in the normalization of the index. The second term of the cost function is meant to give more importance to the solutions which maximize the use of the embodied constraint. An analysis of its role is presented at the end of this section. Given that both the terms vary between 0 and 1, the value of the cost function varies in the interval $[0, 2]$. The closer is the value to 2, the better is the solution.

Our problem is non-convex, and as a solver, we adopted a genetic algorithm (GA) [178]. The procedure for the development of the genetic algorithm follows the prototypical organization of "standard" GAs, as illustrated from the flowchart in Fig. 3.4, which is based on the guidelines explained in [179].

We generated an initial population of 50 individuals where the genes represent our con-

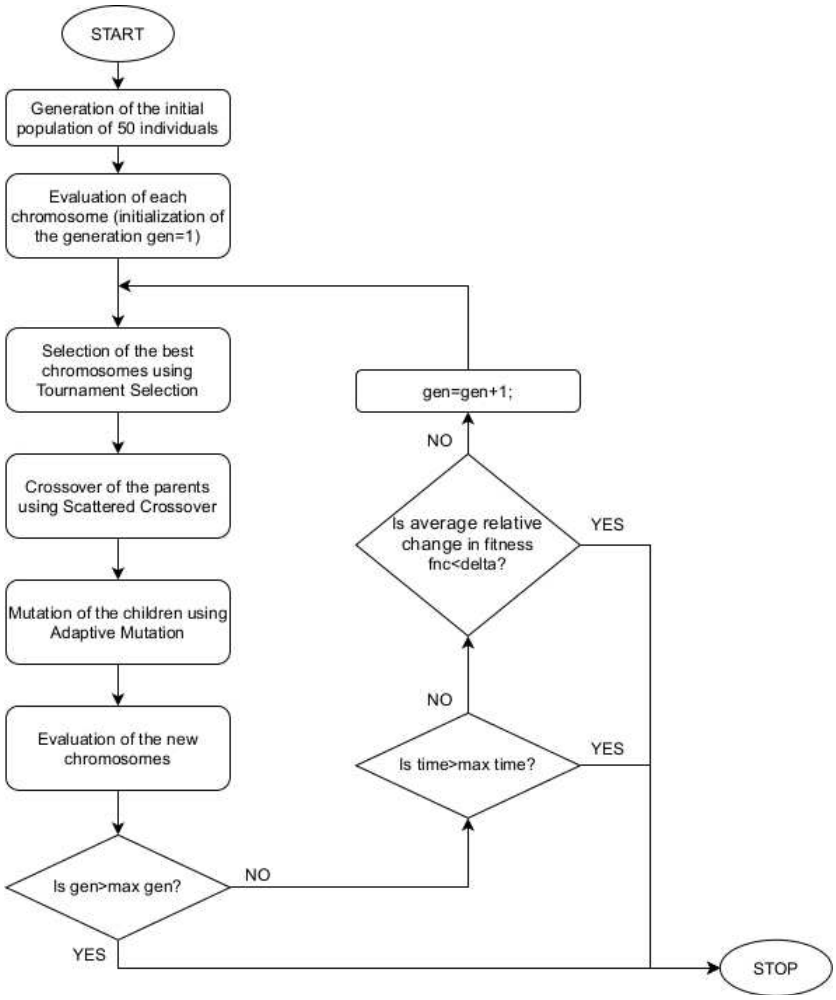


Figure 3.4: Flowchart representing the steps followed by the genetic algorithm to find a solution to the optimization problem. The algorithms we adopted for the selection, the crossover, and the mutation have been reported, together with the stopping criteria.

strained variables in (4.1). We adopted (4.2) as a fitness function. The best individuals were chosen using the Tournament Selection to speed up the entire process. After each generation, we generated the children with a scattered crossover function, such that the variables are inherited by the new individuals with a random binary vector. We adopted an adaptive mutation algorithm that randomly generates adaptive directions for the last successful or unsuccessful generation. The algorithm stops when the average relative change in the fitness function value is less than a certain threshold δ. We assumed $\delta = 0.05$. We noticed that, in the worst case, the algorithm stops after the 10th generation. To determine the number of generations necessary to obtain reasonable solutions, we performed several tests before the experimental phase.

Fig. 3.5 illustrates the fitness score of each individual (in blue), the best score (in green), and the mean score (in red) for each generation until the algorithm reaches the stop condition.

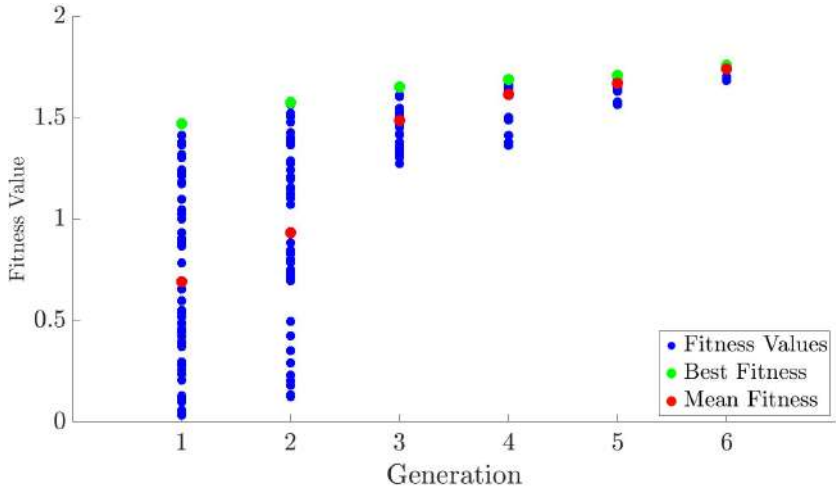


Figure 3.5: Fitness score evaluated for the apple. We reported the fitness value (on the y-axis of each individual (in blue), the best (in green), and the mean (in red). As can be noticed, the optimization algorithm converges after the 6th generation (represented on the x-axis). The closer the fitness score to 2, the more the grasp is considered good.

The illustrated behavior has been obtained by running the optimization algorithm for the apple object.

The grasp simulations needed to solve the optimization problem were carried out using the SynGrasp MATLAB Toolbox [48], in which we defined the model of the SSG depicted in Fig. 3.3. To simulate grasps with the SSG we first defined its model based on DH-parameters, and included it in the SynGrasp Toolbox. The optimization algorithm takes as input the normal to the plane on which the scoop should slide and the point cloud of the object to be grasped. Then, the hand model is rotated in the frame of the selected plane and for each combination of decision variables selected by the solver, a grasp is generated by closing the simulated SSG over the model of the object obtained from the point cloud. Since the gripper is underactuated, we implemented in SynGrasp a function to simulate its closing motion based on the concept of synergy [180]. In this way, the gripper model closes with a coordinated motion of all its joints, simulating the fact that the single joints of the device are not independently actuated. For each iteration of the optimization process, the gripper model is positioned with respect to the object model according to the pre-grasp configuration chosen by the genetic algorithm, and the closure of the gripper over the object is simulated in a quasi-static framework [177]. The closure of a finger is stopped when a contact point between its mesh and the object mesh is detected by a contact detection algorithm. The latter is based on geometric considerations and is an adapted version of the one included in the SynGrasp Toolbox. The $\mathbf{G}$ matrix is computed based on the contact points detected by the contact detection algorithm and according to the methods described in [177]. Once the $\mathbf{G}$ matrix is obtained, the GII is computed as in [72].

As regards the scoop, we could have considered several contact points placed on its surface in the GII's estimation. However, we found that it was necessary to insert the second term $A_{scoop}(x)/A_{tot}$ in the cost function for the following reason. The contact detection algorithm

computes the intersections between every contact point and the mesh of the object. Thus, increasing the number of scoop's contact points translates into an increase in the computational complexity of the whole simulation. For this reason, we decided to represent the scoop as a single contact point for the calculation of the GII and to evaluate its real contribution with a separate term in the cost function.

More details on scene segmentation and grasp execution are in Sec. 3.2.3.

To better characterize the contribution of the second term in (3.2), we analysed the results of the optimization carried out with two different objective functions:

$$f_{o1} = GII(\mathbf{x}), \quad f_{o2} = GII(\mathbf{x}) + \frac{A_{scoop}(\mathbf{x})}{A_{tot}}.$$

We acquired the point cloud of 11 different objects (see Table 3.3). We ran the optimization algorithm five times for each objective function, and for each object, considering the object always in the same pose. We noticed that the presence of the second term remarkably affects the solutions found for the distance d between scoop and object centers. Obtained results are reported in Table 3.2 and show that both the mean values and the standard deviations of the distance d are lower when the second term is present. A lower average distance translates into a pre-grasp pose in which the gripper is closer to the object, and thus the scoop is exploited more. Lower standard deviations, instead, indicate that f_{o2} gives more repeatable and reliable solutions, increasing the robustness of the algorithm [181]. Using the optimization results reported in Table 3.2, we made a short comparison between f_{o1} and f_{o2} in terms of grasp success rate in real experiments. We chose 3 objects in the dataset (apple, chips can, box) and performed 5 trials per object, per function. The success rate obtained when using f_{o2} (93%, 14 out of 15) is remarkably higher with respect to that achieved with f_{o1} (60%, 9 out of 15).

Table 3.1 summarizes the result we obtained.

Table 3.1: Success rates obtained for the two cost functions.

ID	Object	Exp. 1 (f_{o1})	Exp. 1 (f_{o2})
1	apple	3/5	5/5
2	chips can	4/5	5/5
3	box	2/5	4/5
	TOTAL	9/15	14/15

3.2.3 Scene segmentation and grasp execution

We adopted the Object Recognition Kitchen Tabletop pipeline [182], to recognize the planes that are present in the scene (captured through an RGB-D camera), and identify a cluster of 3D points belonging to the object. The extraction of the cluster is performed using the Point Cloud Library (PCL) [183], that allows to process the point cloud coming from the camera. The point cloud of the object is then opportunely processed using the Crust algorithm to reconstruct the object shape [184].

Among the detected planes, the one that should be exploited with the scoop can be selected

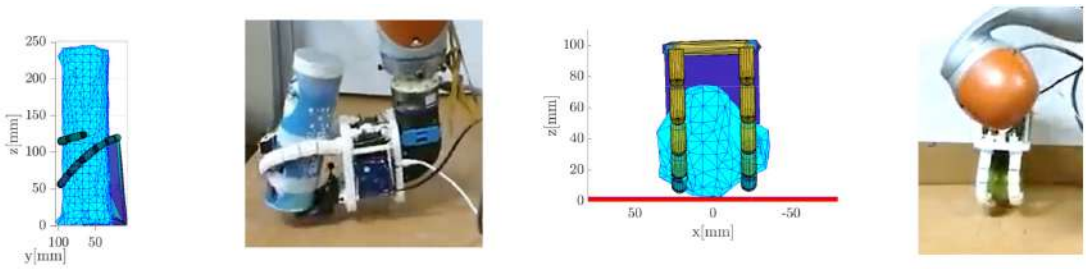


Figure 3.6: Simulated result of the optimization and corresponding real experimental trial for a cylindrical object (chips can) grasped from the table (left), and for a spherical object (apple) grasped by sliding the gripper over a vertical wall (right).

Table 3.2: Optimization results for the variable d averaged over 5 trials for each condition.

Object	Obj. fct.	d (mm)	**Object**	Obj. fct.	d (mm)
apple	f_{o1}	28.2(±4.26)	banana	f_{o1}	8.45(±10.7)
	f_{o2}	9.39(±1.45)		f_{o2}	0.47(±1.82)
bowl	f_{o1}	11.1(±15.1)	chips	f_{o1}	26.4(±10.7)
	f_{o2}	0.22(±2.37)	can	f_{o2}	21.2(±0.89)
spring	f_{o1}	3.59(±7.03)	plastic	f_{o1}	20.5(±8.93)
clamp	f_{o2}	0.53(±0.92)	funnel	f_{o2}	4.57(±0.91)
gelatin	f_{o1}	11.2(±8.11)	mug	f_{o1}	24.9(±13.5)
box	f_{o2}	0.8(±0.51)		f_{o2}	12.3(±5.27)
pasta	f_{o1}	18.9(±8.72)	screw-	f_{o1}	4.74(±1.83)
pack	f_{o2}	9.49(±3.56)	driver	f_{o2}	1.36(±0.73)
toy	f_{o1}	3.39(±1.66)			
dolphin	f_{o2}	1.67(±0.83)			

either by the operator, as we did to implement the experiments presented in Sec. 3.3, or automatically. In the second case, the plane to slide on could be chosen, for example, based on the distance between the object and the plane, or according to task constraints (e.g., obstacles to avoid, objects affordances to use, etc.). An example of the outcome of the scene segmentation is shown in Fig. 3.7.

Once the main features of the scene are correctly detected and the sliding plane is selected, the grasp planning algorithm based on the optimization problem in Eq. (3.2) can start searching for the optimal pre-grasp pose, that we indicate with $\mathbf{x}^* = [\theta_R^*, \theta_L^*, d^*, \gamma^*, \alpha^*]$.

Based on the result of the optimization, the execution of the grasp proceeds as follows. The orientation of the fingers is set using the two servo motors to achieve θ_R^* and θ_L^*. Then, the robot arm supporting the SSG is moved to let the scoop rotate of an angle γ^*. Hence, a hybrid force-velocity controller is implemented such that the SSG moves towards the surface to slide on until it touches the surface itself. Upon termination, the scoop is moved towards the object

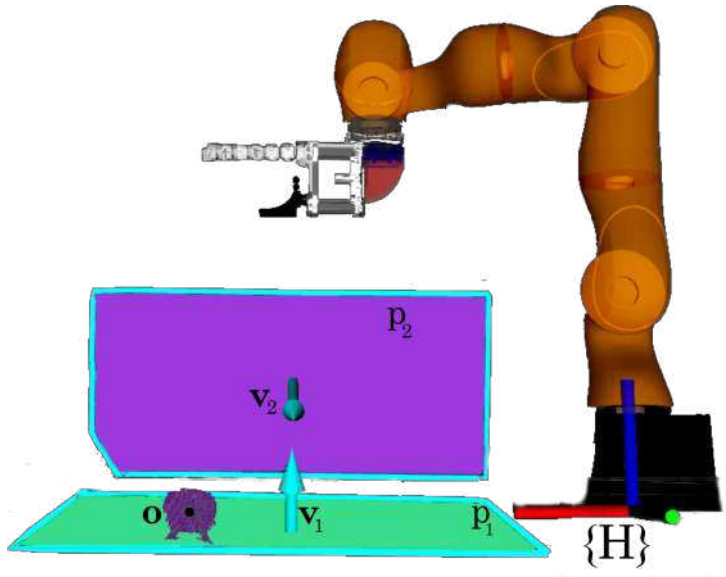


Figure 3.7: Scene detection: planes p_1 and p_2 with normals v_1 and v_2 respectively, and object o.

at a constant speed $\vec{v}$ parallel to the surface, while maintaining a constant force in the direction normal to the surface. During its sliding motion towards the object, the end-effector is rotated with respect to the table surface by an angle α^*. Thus, the angular velocity of the robot end-effector is computed as: $\vec{\omega(t)} = (\vec{v}(t)/d(t)) \cdot \alpha^*$, where $d(t)$ is the distance between the object and the hand. Once the distance between the hand and the object reaches the desired value d^*, the gripper controller flexes the fingers until they touch the object. Then, the scoop is actuated, and the object is pushed against the palm until it is caged.

Two examples of the simulated outcomes of the optimization and their real counterparts are shown in Fig. 3.6. While the cylindrical chips can is grasped from the table, the apple is grasped sliding the scoop over the wall. An interesting result to notice is that in the case of the cylinder the fingers are rotated of different angles in order to hug the object, while for the apple both θ_R and θ_L are set to zero.

3.3 Experimental results

A video of the experimental trials can be found at link.

The experimental setup, shown in Fig. 3.8, included a LBR iiwa 7 robot arm (KUKA AG), a Gamma 6-axis force-torque sensor (ATI Industrial Automation, Inc.), and the Soft ScoopGripper attached to the end-effector [9]. A Kinect One RGB-D camera (Microsoft) was used to detect objects and planes in the scene as described in Sec. 3.2.3.

The adopted dataset of objects is shown in Fig. 3.8 and described in Table 3.3. The objects were chosen to have a wide range of sizes, weights and shapes and most of them come from the YCB Dataset [185]. The apple, the gelatin box, and the chips have paradigmatic shapes (sphere,

cuboid, cylinder); the banana, the spring clamp, and the screwdriver have a small height and diverse shapes; the pasta pack is heavy; the toy dolphin is deformable; the funnel has a complex shape and can rotate when touched; the mug and the bowl are hollow. The cracker box was only used in Experiment 3 as a support for another object.

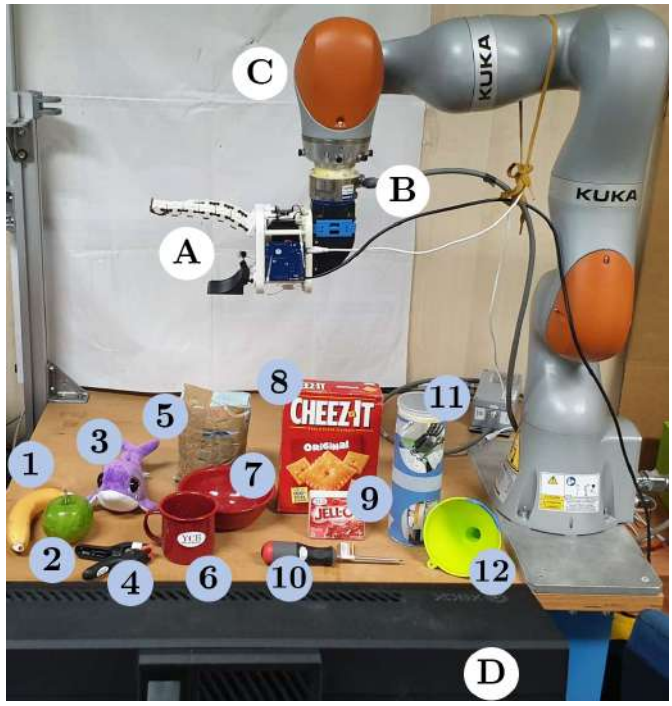


Figure 3.8: Experimental setup and dataset of objects. (A) Soft ScoopGripper, (B) ATI F/T sensor, (C) KUKA LBR iiwa, and (D) Kinect. Objects are indicated with numerical IDs and their properties are reported in Table 3.3.

Three experiments were performed to test the grasp planning algorithm explained in Sec. 3.2. Experiment 1 was meant to test the effectiveness of the grasp planner when dealing with objects in contact with just one surface in the environment, i.e., laying on a table. In this case, the success rate over 11 different objects, picked up 10 times each, was evaluated in correlation to objects' characteristics. Experiment 2 aimed at testing the scoop grasp in situations in which the object is in contact with two different surfaces, i.e., table and wall. Here, the most relevant aspect to study was where to slide the scoop to obtain higher success rates, given objects' features. Experiment 3 had the objective of showing the applicability of the proposed grasping strategies in real world scenarios, including grasping in clutter or inside boxes. In all the experiments, a grasp was considered successful if the object was picked up and moved to the final position without falling. Otherwise, it was considered unsuccessful.

Table 3.3: Dataset of objects.

ID	Object	Weight (g)	Size (mm)
1	banana (YCB)	66	36×190
2	apple (YCB)	68	75
3	toy dolphin	84	$80 \times 200 \times 90$
4	spring clamp (YCB)	59	$90 \times 115 \times 27$
5	pasta pack	510	$65 \times 114 \times 182$
6	metal mug (YCB)	118	80×82
7	metal bowl (YCB)	147	159×53
8	cracker box (YCB)	411	$60 \times 158 \times 210$
9	gelatin box (YCB)	97	$28 \times 85 \times 73$
10	screwdriver (YCB)	98.4	31×215
11	chips can (YCB)	205	75×250
12	plastic funnel	21	125×115

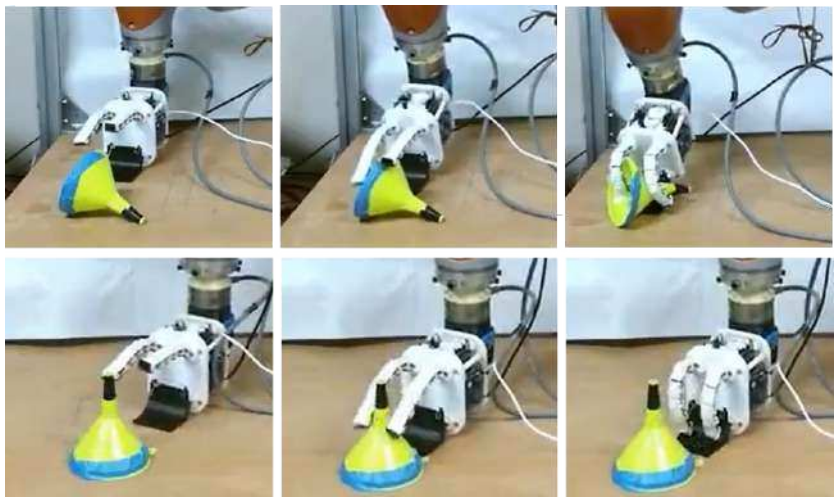


Figure 3.9: Experiment 1: grasp success (top) and failure (bottom) of the funnel.

3.3.1 Experiment 1: scoop grasp exploiting a table

First, we decided to analyze the case where the object is placed on a table, far from any other possible constraint. The scene segmentation algorithm recognizes that no walls or inclined planes are exploitable by the SSG. Then, the optimization problem is solved considering that the scoop will slide on the table and the constrained variables are referred to the reference frame placed on the table itself. The grasp is executed as explained in Sec. 3.2.3 and two trials (a success and a failure) are shown in Fig. 3.9. We carried out 110 trials with this setup, 10 for each object. The third column of Table 3.4 shows the obtained success rates.

Table 3.4: Success rates obtained in Experiment 1 and in the two conditions tested in Experiment 2.

ID	Object	Exp. 1	Exp. 2 (wall)	Exp. 2 (table)
1	banana	10/10	3/6	5/6
2	apple	10/10	3/3	3/3
3	toy dolphin	10/10	6/6	6/6
4	spring clamp	9/10	8/18	16/18
5	pasta pack	4/10	7/18	12/18
6	metal mug	10/10	3/3	3/3
7	metal bowl	10/10	3/3	3/3
8	gelatin box	8/10	6/18	16/18
9	screwdriver	6/10	1/6	3/6
10	chips can	9/10	9/9	9/9
11	plastic funnel	8/10	7/9	9/9
	TOTAL	94/110	56/99	85/99

3.3.2 Experiment 2: scoop grasp exploiting a table or a wall

The second experiment consists in grasping objects which are constrained from two sides. The objects were placed close to a vertical wall and we tested two different types of grasp approaches: *i)* scoop sliding on the wall, towards the object, *ii)* scoop sliding on the table towards the object. In the first case, the optimization algorithm is solved considering that the sliding surface is vertical, thus, in the simulation, the hand is rotated to approach the object from above.

In both approaches, depending on the desired scoop orientation, the fingers might end up touching the constraint perpendicular to the exploited one, during their closure. To ensure a gentle slide and avoid a collapse of the fingers over the surface, we had to properly set the boundaries of the optimization variable d, based on the length of the fingers.

The two conditions were tested with the same 11 objects that were used in Experiment 1. Each object was placed in a certain orientation with respect to the environmental constraints, with one side touching the wall and one touching the table. Then it was grasped with the two different approaches. While in Experiment 1 we performed 10 trials for each object, in Experiment 2 we performed 3 trials for each possible object orientation with respect to the environmental constraints. Objects with a rounded shape (apple, mug, bowl), for example, were tested 3 times per strategy as they have only one possible orientation with respect to the wall, without considering them upside-down since the apple is almost spherical and for the other two objects we want to keep the hollow part up. The box, instead, as well as the pasta pack and the spring clamp were tested in 6 different orientations. Obtained results are summarized in the fourth and fifth columns of Table 3.4. Successful trials obtained letting the scoop sliding on the wall and on the table are shown in Fig. 3.10 and in Fig. 3.11, respectively.

3.3.3 Experiment 3: use cases

Experiment 3 deals with use case scenarios. In particular, we analysed cases where the surface of an object is used to carry out a scoop grasp (Fig. 3.12a) and cases where the scoop is used inside

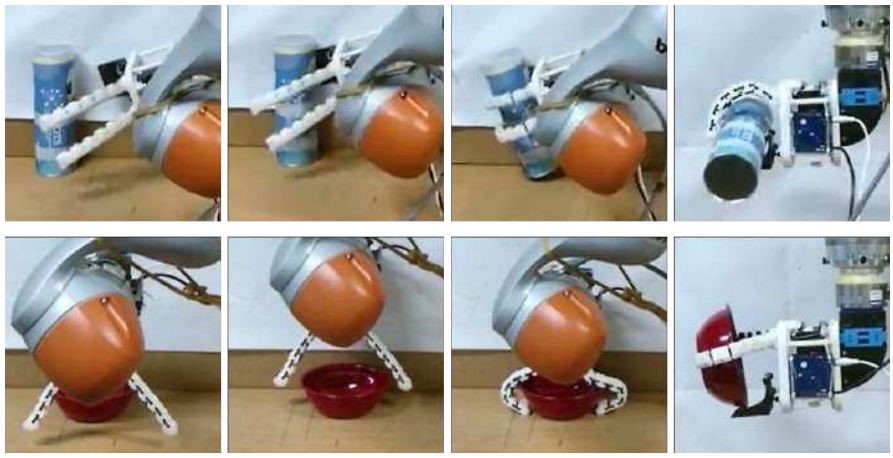


Figure 3.10: Experiment 2: grasps from the wall: only the wall constraint is exploited (top); the scoop slides on the wall and the fingers slide on the table and grasp the object thanks to their compliance (bottom).

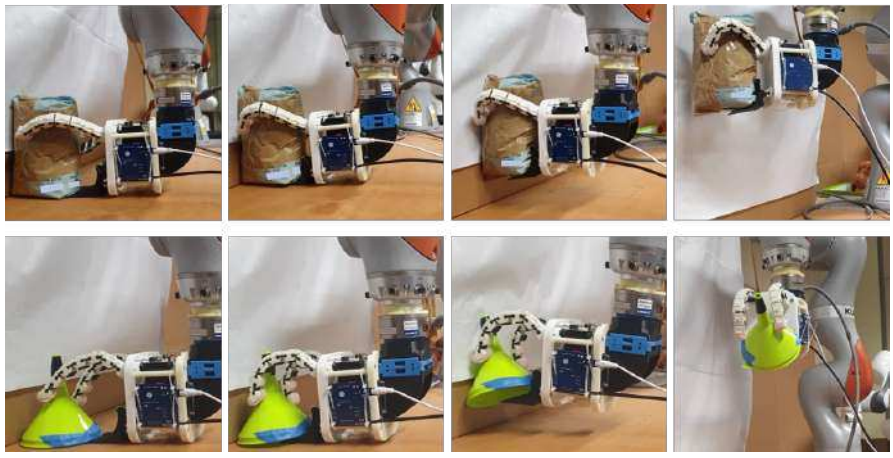


Figure 3.11: Experiment 2. Successful grasp of the pasta pack (top) and of the funnel (bottom) exploiting the table.

a box (Fig. 3.12b). Two additional objects were used: a candy tube (18×228 mm, 12 g) and a box ($74 \times 153 \times 125$ mm, 93 g). A total of 9 experiments were performed for each use case (6 for the box, and 3 for the candy tube), i.e., one test for each possible object orientation. In the first scenario, the surface of the cracker box was used as a sliding plane, and we obtained 8 successes out of 9 grasps. The second scenario involved grasping an object inside a box, exploiting also the internal walls as additional constraints. In this case, we achieved 7 successes out of 9 trials.

(a) (b)

Figure 3.12: Experiment 3. (a) Scoop grasp exploiting another object (cracker box). (b) Scoop grasp inside a box.

3.4 Discussion

Previously presented results indicate that the SSG can grasp a wide range of objects with the same strategy, exploiting hand reconfigurability, and embedded and environmental constraints. Considering all objects grasped with the configuration obtained with our optimization algorithm, 250 out of 326 grasps were successful, with an overall grasp success rate of about 77%. However, the conducted experiments are different in nature and need to be evaluated in detail.

3.4.1 Experiment 1

The overall success rate of Experiment 1 is about 85%. This means that the adopted scoop grasp strategy works fine with different objects in different positions. In the following we will analyse in detail the relation between the success rate, the object features and the optimization variables.

In Fig. 3.13, we report a matrix of scatter plots and histograms of the data related to Experiment 1, which are gathered by dividing each variable into two groups: successful and unsuccessful grasps. Success rates are reported with respect to $\mathbf{x}$ and the objects' features. The selected properties are the object's weight and height; the second one is dependent on the object's pose.

As we can notice, in the first two histograms related to θ_R and θ_L, the workspace of the fingers is spanned in an unbalanced manner. Indeed, 76 out of 110 trials were performed exploiting ranges of the variables θ_R and θ_L belonging respectively to $[-21°, 0°]$ and $[0°, 21°]$. This is mainly related to the objects' height. Indeed, we can notice a correlation between the variation in height and the use of a certain range of θ_R and θ_L. Shorter objects are easily wrapped by caging the fingers towards the scoop, i.e., maintaining the two angles in a small interval near zero degrees. On the other hand, taller objects (such as chips can and pasta pack) are successfully grabbed by rotating the fingers in the ranges $[-120°, -50°]$ and $[50°, 120°]$ to push the object on the scoop towards the palm (see also Fig. 3.6).

In 56.4% of experiments, the optimization algorithm provided an object-hand distance d in the range $[0, 5]$ mm. These low values are mainly due to the effect of the second index introduced in the cost function, which tends to maximize the portion of the scoop area occupied by the

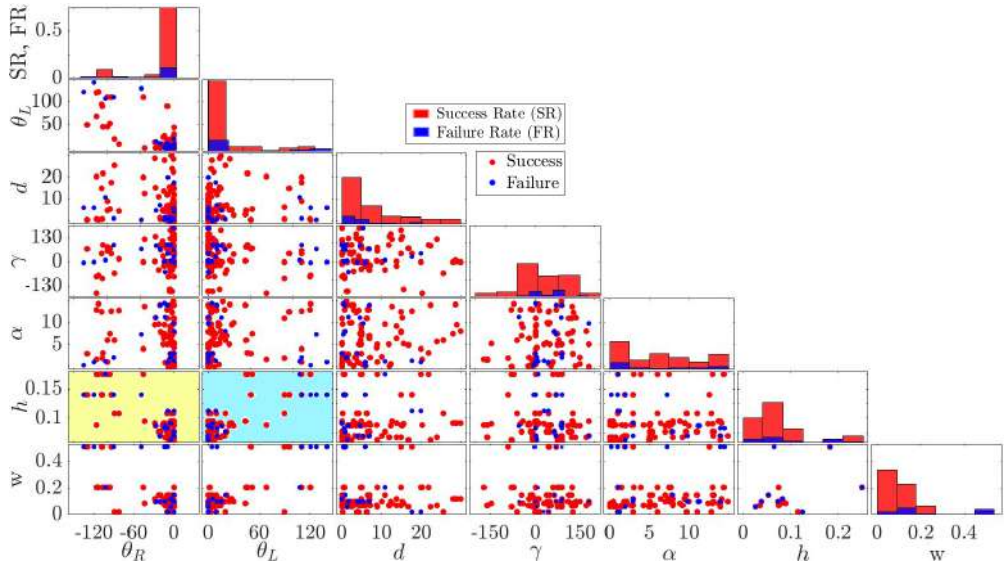


Figure 3.13: Success and failure rates related to pre-grasp pose variables ($[\theta_R, \theta_L, d, \gamma, \alpha]$), and object characteristics (height h and weight w). Notice that the y-axis of the histograms reports the values of success and failures rates, going from 0 to 0.75 (see the y-axis label in the top left panel). All angles are expressed in degrees, the height in m and the weight in kg. The light yellow and light cyan boxes highlight the correlation between the orientations of the fingers (θ_R, θ_L) and the object height (h). The Pearson correlation coefficients $\rho_{\theta_R h}$, $\rho_{\theta_L h}$ are -0.82 and 0.79, respectively: the higher is the object, the greater are the absolute values of the angles.

object. No particular trends link the value of d to a failed grasp.

The variable γ, indicating the direction of the scoop approach towards the object, most of the times varies in the range $[-60°, 150°]$. The reason for this trend is related to the workspace of the robot arm. The failure trend is equally distributed across the range.

There are no remarkable trends for the variable α.

Lastly, as the weight increases, inevitably, the success rate will tend to zero once the payload value supported by the scoop is exceeded. In fact, 37.5% (6 out of 16) of failed grasps are related to the weight of the object (pasta pack).

3.4.2 Experiment 2

In both the conditions tested within Experiment 2, the apple, the bowl, the metal mug, the chips can, and the toy dolphin were successfully grasped in all the orientations.

Short objects such as the banana, the gelatin box, and the screwdriver resulted difficult to be picked from the wall. This is mostly due to the fact that the SSG fingers are rather long and they need to be closed before reaching the object with the scoop to avoid their collapse over the table during the closure motion. A longer scoop (or shorter fingers) would avoid this problem. In general, a trade-off needs to be found. A longer scoop could facilitate the grasp from the wall, but, at the same time, might complicate the one from the table. The screwdriver was the object in the dataset picked up only once after sliding on the wall. The spring clamp was successfully

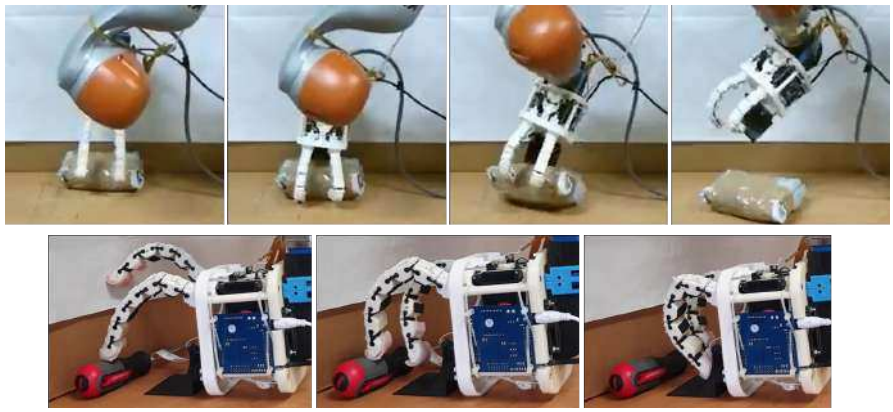


Figure 3.14: Experiment 2: grasp failures. (top) Pasta pack reached from the wall. (bottom) Screwdriver from the table.

grasped in almost half of the cases, and in two of these, the wall grasp succeeded, while the grasp from the table failed. Heavy objects like the pasta pack are very difficult to be picked up from the wall. One of the failed grasps of the pasta pack is shown in the top part of Fig. 3.14.

The grasp from the table failed only with the heavy pasta pack and with short objects (banana, gelatin box, spring clamp, screwdriver). The bottom part of Fig. 3.14 shows the failure obtained with the screwdriver: also here, like in the case of the grasp from the wall, the fingers close before the scoop reaches the object. In general, the results of Experiment 2 show that when it is possible it is better to choose to grasp objects from the table (success rate: 86%) rather than from the wall (success rate: 56%). This is also due to the fact that while in the first case the scoop is perpendicular to the direction of the gravity, in the second, the weight of the object goes entirely on the fingers.

In contrast with Experiment 1, in which the hand grasps objects that are constrained only on one side, in Experiment 2 the second constraint can help to push the object on the scoop. This was shown to be particularly useful when the scoop slides on the table. Remarkable examples are the funnel and pasta pack (see Fig. 3.11). The first one was grasped in the only pose the gripper could not grasp it in Experiment 1 (Fig. 3.9). The pasta pack was grasped 12 times out of 18, with a success rate of almost 67%, instead of 40%.

3.4.3 Experiment 3

Experiment 3 had the objective of testing two possible applications of the scoop grasp. In the first, an object was used as a sliding surface to grab another one attached to it. This simulates a situation in which the object to grasp is so close to another that grasping it with classical grippers and strategies might be ineffective, unless using a very precise positioning of the gripper or using pushing motions to divide the objects. Thanks to the SSG we can use a single motion to achieve the grasp. Sliding between two objects, the scoop manages to separate them and grasp the object of interest while keeping the other in almost static conditions. In the 9 trials, there was only 1 failure, when the candy tube was put with the longest side attached to the cracker box. As discussed above, short objects are difficult to be picked from a surface perpendicular

to the table.

In the second scenario, the object was placed inside a box, simulating a situation in which there are multiple environmental constraints. The task was to slide on the box bottom surface and grab objects by exploiting the box walls. In this case, also the corners can be used (object constrained from three sides). The exploitation of the additional surface is made possible by the reconfigurability of the fingers (see Fig. 3.12b), which allows avoiding collisions with the box wall. The grasp succeeded in 7 out of 9 trials. The 2 failures were recorded with the candy tube. The first one was due to the box size that did not permit the hand to slide under the object. The second one was related to the box inclination that did not allow the candy tube to stand in a vertical position.

3.5 Conclusions

In this chapter, we presented an approach to grasp objects exploiting the embodied constraint (a scoop) and the reconfigurable fingers of a soft gripper. We implemented the so-called "scoop grasp", where the soft fingers are exploited to cage the object on the inherent constraint while it slides under the object itself.

Through the use of an optimization algorithm taking as input the point cloud of the object to be grasped, we searched for the best configuration of the fingers and the best pre-grasp pose for each object. We shaped the cost function of the optimization problem such that the grasp is robust and the embodied constraint is efficiently used. Even though the proposed algorithm is thought to be used with a particular gripper, the SSG, the general approach of combining two terms, one considering the grasp quality and one taking into account the actual exploitation of the embedded constraint, could also be applied to other similar devices.

We performed three different experiments over various objects. First, we tested the scoop grasp strategy over objects constrained only on one side, i.e., laying on a table. Then, we considered a different set-up, where the objects were constrained from two sides, from the table and from the wall. The scoop grasp was performed sliding the scoop over the wall and over the table. Lastly, we performed other experiments in real world scenarios to prove the effectiveness of the SSG in different situations. Experimental results show that reconfigurable soft grippers with embedded constraints represent a viable alternative to more complex soft hands if used with grasping strategies that allow exploiting their features. Reconfigurability allows grasping a variety of objects, whereas the use of the scoop allows exploiting the surfaces in contact with the objects. The intrinsic and passive adaptability of the hand is key to the success of the scoop grasp.

In future work, we will study whether a synergistic approach combining hand design and grasp planning can lead to improved success rates. We will investigate how to adapt the hand design (e.g., scoop length, joints stiffness, finger length, etc.) to the characteristics of the objects, possibly introducing variables related to the hand design in the optimization problem. Future research will also focus on the development of an algorithm able to determine on which plane the SSG should slide on based on the detected scene.

Chapter 4

Learning grasping primitives from humans to exploit embedded constraints

"*Tell me and I forget, teach me and I may remember, involve me and I learn.*"
Benjamin Franklin

Finding effective grasp strategies constitutes one of the main challenges in robotic manipulation, especially when dealing with soft, underactuated, and non-anthropomorphic hands. This work presents a Learning from Demonstration approach to extract grasp primitives using the reconfigurable soft hand, the Soft ScoopGripper (SSG). Starting from human demonstrations, we derived Gaussian models through which we were able to devise different grasping strategies, exploiting the SSG features. As the grasping strategies are tightly related to the characteristics of the object to be grasped, we test two different ways of modeling objects in the training dataset and we comparatively evaluate the resulting primitives. Experimental grasping trials on unknown test objects confirm the effectiveness of the learned primitives and demonstrate that assuming different levels of knowledge about the object representation in the training phase does not significantly influence the grasp success rate.

The chapter is organized as follows. Sec. 4.1 defines the problem of pre-grasp pose computation, deriving its motivation. In Sec. 4.2, we describe the proposed methodology. In Sec. 4.3 Sec. 4.4, we report and discuss the experimental results. Lastly, Sec. 4.5 derives the conclusion of the study, outlining its possible further developments.

4.1 Introduction

One of the main issues in robotic manipulation consists in estimating the grasping strategies of newly designed grippers. If the kinematic structure of the robotic hand is anthropomorphic, it is, at least in principle, possible to analyze how humans grasp and manipulate objects [186] and then transfer the learned skills to the robotic device. For this purpose, devices capable of capturing the human hand kinematics have been developed, e.g., datagloves [187]. Even though reproducing anthropomorphic grasps is not a trivial task in general, still the human hand constitutes the fundamental example from which to take inspiration. When dealing with non-anthropomorphic grippers, instead, the achievable grasping strategies might be completely different from those used by humans, and other methods should be adopted to fully exploit their capabilities. This is particularly true for soft and underactuated grippers which may reach different configurations not only based on their design and actuation inputs, but also due to the physical interaction with the surroundings [49, 4].

In this chapter, we propose to learn new grasping strategies for soft non anthropomorphic grippers from human demonstrations performed through kinesthetic teaching. Instead of defining feasible grasp strategies a priori and only based on the gripper design, we propose first to observe how humans would use the robotic device and then extract grasping primitives for it.

We opted for the kinestetic guiding approach instead of using teleoperation [107, 106], because it is more intuitive for operators [109].

Differently from recent works where human demonstrations are processed through deep classifiers [51], the method we propose here employs the Gaussian Mixture Model (GMM) and non-labeled data.

Figure 4.1: Example of human demonstration. A user handles the robot to grasp an object during the training phase.

Grasp demonstrations were performed over several different objects with a robot manipulator endowed the Soft ScoopGripper [9] (Fig. 4.1). Its peculiar design makes the SSG a good example of versatile, but highly non-anthropomorphic robotic hand. The latter is provided with an embedded constraint that can be used in several different ways, and we believe that studying how humans would use it and transfer this knowledge to autonomous grasping could be an

effective way to fully exploit the SSG capabilities. Even though we applied our method to this specific device, we believe it could easily be generalized to other grippers as it reaches good results with a limited number of demonstrations and no assumptions are made on the gripper structure.

The main contribution of this work is a method that, based on human demonstrations, evaluates the possible pre-grasp poses of a soft robotic gripper. So far, relatively few works in literature have applied learning methods in the control of soft hands [51, 132, 52, 104].

The parameters of the pre-grasp pose are determined by a Gaussian Mixture Regression (GMR) that takes as input the geometrical properties of the object extracted from RGB-D images. The use of regression allows us to find pre-grasp poses of unknown objects without needing labeled data [188, 189] or lookup tables. In addition, we do not rely on grasp candidates as done in other works [190], as this would be ineffective with soft hands, for which it is difficult to precisely predict the contact points on the object.

In [113], the authors propose to build a GMM for each object, to obtain fast responses during real-time grasping. Similarly, we decided to use a model encoded with GMM, even if we have chosen to build the models taking into account the object's properties together with the hand-object relative pose. In this way, our method is composed by a finite number of models and by means of GMR we do not need new demonstrations to retrieve the pre-grasp poses of objects which are not present in the training dataset thanks to the fact that GMR is able to interpolate the missing data in an efficient way [112].

The proposed method also allows to compute the models and the pre-grasp poses faster and using less training data than other approaches, for which the computation time depends proportionally on the number of the problem variables [104]. Indeed, as Calinon *et al.*, pointed out in [112], the strength of GMR lies in the feature of modeling a joint probability density function of the data and then deriving the regression function from the joint density model. This is an advantage since the input and output components are only specified lastly. Given that the density estimation can be learned offline, the regression process can be computed very rapidly.

Once data are acquired they are processed in two different ways, depending on the level of knowledge we assume to have on the object. In particular, we compare a *bounding box*-based method in which each object in the training set is modeled with its bounding box (retrieved from the object point cloud), and a *shape*-based method in which objects are labeled with their shapes (sphere, cylinder, hollow, and cuboid).

The contribution of this work is thus twofold: *i)* we propose a full pipeline to learn grasping strategies for non-anthropomorphic hands from human demonstrations, *ii)* we show advantages and disadvantages of two different approaches to the modeling of objects in the training set, one unsupervised and one supervised.

4.2 Materials and Methods

This chapter presents a method for extracting grasping primitives for soft non-anthropomorphic hands based on a Learning from Demonstrations approach. We compare two different approaches to extract the primitives, one that does not imply any assumption on the object characteristics, and one which is based on the knowledge of the object shape. Fig. 4.2 shows the block diagram

of the two proposed procedures: i) *bounding box*-based and ii) *shape*-based. In the following sections, we detail the proposed methodology.

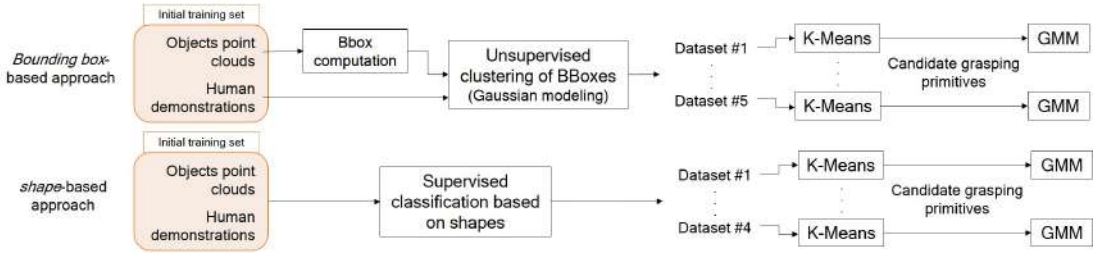


Figure 4.2: Block diagrams of the proposed approaches.

4.2.1 Data acquisition

Six users were asked to control in kinesthetic guidance a collaborative robot arm (UR5, Universal Robots) equipped with the SSG to grasp different objects (see Fig. 4.1). They had to physically move the end-effector towards an object placed on a table and then grasp it with the SSG. The SSG structure is specifically designed for exploiting the environmental constraints thanks to its embedded constraint, i.e., the scoop, which also allows to robustly hold the objects placed over it. On the control side, the fingers can be closed and opened with a simple command, while they can be rotated specifying the desired angle. In this work, we used a newer version of the gripper, in which two additional motors are used to rotate (up to 180^o) the dovetail joints placed at the bases of the fingers. The SSG is shown in Fig. 4.4, where $\{\mathbf{H}\}$ and $\{\mathbf{O}\}$ indicate the hand and the object reference frames, respectively. Notice that, in the next sections, the SSG pose will always be computed with respect to $\{\mathbf{O}\}$.

A total of 360 experimental trials were executed asking users to grasp 12 different objects (see Fig. 4.3 and Table 4.1 for their characteristics). We chose the objects to have a wide range of sizes, and most of them belong to the YCB Dataset [185].

Users performed 5 trials per object and were instructed to test always different directions of approach. Each demonstration started with the robot and the object in a random pose and users had to guide the robot end-effector towards the object. When they deemed the object appropriately graspable by the robotic hand, they pushed a button to close the fingers around the object. Operators could also modify the configuration of the fingers by controlling the rotation of the dovetail joints using a keyboard.

A Kinect One RGB-D camera was used to detect the object in the scene. We assigned the object reference frame $\{\mathbf{O}\}$ such that the z-axis is always perpendicular to the table plane, the y-axis is placed on the longest side of the object, and the x-axis is computed as the cross-product of the other two (see Fig. 4.4). During the demonstrations, we collected the robot Cartesian poses, the point cloud of the object, and the object pose with respect to the robot base frame. Only the trials with successful grasps were recorded and stored.

Figure 4.3: Objects used in the training phase (details in Table 4.1).

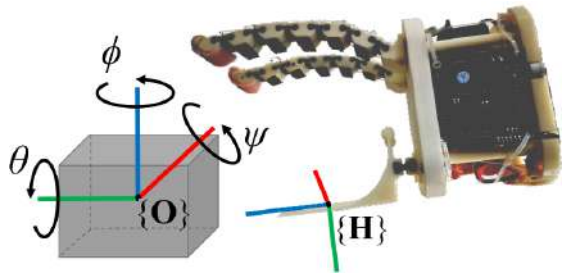


Figure 4.4: The Soft ScoopGripper with its reference frame represented with respect to an object. The rotations are expressed on the object reference frame. The axes are represented in red (x), green (y), and blue (z).

4.2.2 Gaussian modeling for pre-grasp poses extraction

After performing the demonstrations, for each trial we extracted the pre-grasp pose of the hand, i.e., the SSG pose when the closing of the fingers took place. Then, we transformed the extracted pre-grasp poses with respect to the object reference frame to make them independent of the robot base frame. The resulting training dataset included the object-hand relative poses (position $\mathbf{h} = [h_x, h_y, h_z]$ and orientation $\mathbf{o} = [\phi, \theta, \psi]$, described as Euler angles), the fingers orientations ($\boldsymbol{\gamma} = [\gamma_L, \gamma_R], \gamma_L \in [0, \pi], \gamma_R \in [-\pi, 0]$), and the object properties ($\mathbf{d}$).

We decided to make two different assumptions on the level of knowledge gained about the object. This lead to the definition of different $\mathbf{d}$ vectors in the two cases (see the second column of Table 4.2).

In the first approach (*bounding box*-based), the objects are considered as bounding boxes whose dimensions l, w, h along the x, y, z directions of the object frame $\{\mathbf{O}\}$, respectively, are retrieved from the object point-cloud. By doing this, we have neglected the real shapes of the objects by approximating all of them to cuboid profiles. For each trial in the training set, we computed the bounding box of the object and stored its dimensions. Then, using Gaussian modeling, all the obtained bounding boxes were clustered into 5 classes according to their size.

Table 4.1: Properties of training objects.

ID	Object	Size (mm)	Class *Bbox*-based	Class *Shape*-based
1	strawberry (YCB)	⌀35	1	Sphere
2	peach (YCB)	⌀64	5	Sphere
3	apple (YCB)	⌀76	5	Sphere
4	small cylinder	⌀40 × 130	1, 4	Cylinder
5	candy tube	⌀36 × 225	2, 4	Cylinder
6	chips can (YCB)	⌀66 × 230	2, 4	Cylinder
7	metal mug (YCB)	⌀90 × 80	5	Hollow
8	plastic funnel	⌀120 × 130	3	Hollow
9	metal bowl (YCB)	⌀160 × 50	3	Hollow
10	cube	30 × 30 × 30	1	Cuboid
11	jelly box (YCB)	25 × 85 × 70	1, 5	Cuboid
12	brown box	52 × 185 × 135	2, 3, 4	Cuboid

Table 4.2: Object properties $\mathbf{d}$ and number n of extracted primitives for each class of the two different approaches.

Approach	Object properties $\mathbf{d}$	n = # of candidate primitives
Bounding box-based:	$\mathbf{d} = [l, w, h]$	
- Class #1		4
- Class #2		3
- Class #3		2
- Class #4		4
- Class #5		2
Shape-based:		
- Sphere	$\mathbf{d} = [d]$	2
- Cylinder	$\mathbf{d} = [d, h]$	3
- Hollow	$\mathbf{d} = [d, h]$	2
- Cuboid	$\mathbf{d} = [l, w, h]$	4

The number of classes was determined adopting the Bayesian Information Criteria (BIC). As reported in the fourth column of Table 4.1, some of the objects belong to more classes. This is because objects were considered in various poses (e.g., the candy tube can either stand or lay on the table), thus resulting into different values for l, w, h.

In the second approach (*shape*-based), we decided to make an assumption concerning the object characteristics by splitting the demonstrations into 4 datasets classified by the object shapes (sphere, cylinder, hollow, and cuboid). The steps described in the following are performed for both the adopted approaches. On each training dataset, the K-means algorithm is applied

to cluster data based on the similarity of the gripper orientations $\mathbf{o}$ and heights h_z. Different numbers n of clusters were tested and those resulting in the highest silhouette score were chosen. The silhouette analysis measures how similar a data is to its own cluster compared to other clusters [191]. As reported in the third column of Table 4.2, the obtained n varies according to the adopted approach. Each identified cluster corresponds to a candidate grasping primitive. Fig. 4.5 reports the clusters obtained from the K-means analysis for the two approaches.

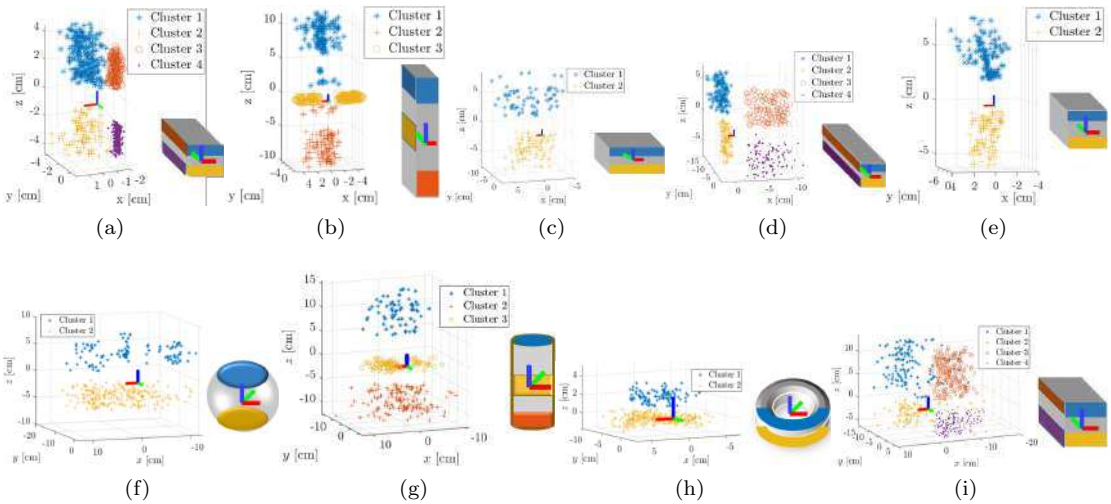


Figure 4.5: Pre-grasp poses clusters obtained from the K-means analysis. Outcomes of the *bounding box*-based approach: a) Class #1; b) Class #2 c) Class #3 d) Class #4 e) Class #5. Outcomes of the *shape*-based approach: f) Spherical objects; g) cylindrical objects; h) hollow objects; i) cuboidal objects. All the poses are expressed with respect to the object reference frame whose axes are: x (red), y (green), and z (blue). For a better understanding, the clustered poses are depicted as a colored area on the surface of each related class (on the right of each graph). The representation of the class has the purpose of identifying the spatial relationship between each object class and the pose of the hand. The correspondence between colors of each cluster on both graphs and classes is maintained.

Note that the K-means step was fundamental, as it allowed us to do a first differentiation in the data, avoiding to mix information, possibly losing some valuable insights. This is particularly important because the Gaussian Mixture Regression, which we employ to find the missing data, fits the data as a whole; therefore, a large amount of completely different data cannot be processed in a proper way [192].

We modeled each candidate primitive using a Gaussian Mixture Model Ω to get a probabilistic encoding of the joint distribution $P(\mathbf{h}, \mathbf{o}, \boldsymbol{\gamma}, \mathbf{d}|\Omega)$. The models are characterized by K Gaussian components.

We determined the number of Gaussians (or states) K by performing the Bayesian Information Criteria (BIC) for each model.

Each model can be represented as

$$P(\mathbf{h}, \mathbf{o}, \boldsymbol{\gamma}, \mathbf{d}|\Omega) = \sum_{k=1}^{K} p_k p(\mathbf{h}, \mathbf{o}, \boldsymbol{\gamma}, \mathbf{d}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{4.1}$$

where p_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ correspond to the priors, the mean, and the covariance of each Gaussian component, respectively.

We initialized the Expectation-Maximization (EM) algorithm using the K-means algorithm. Then, the parameters p_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ are iteratively updated by means of EM algorithm until convergence. Eventually, we used the Gaussian Mixture Regression to reconstruct the missing Gaussian components that led to the pre-grasp pose $\boldsymbol{p}$ specifying the object properties $\mathbf{d}$. The desired pre-grasp pose $\boldsymbol{p}$ is computed as the sum of all the products between each Gaussian's weight $w_k(\mathbf{d})$ and the multivariate Gaussian distribution of the predicted values $\mathbf{h}$, $\mathbf{o}$ and $\boldsymbol{\gamma}$ in the kth Gaussian.

$$\boldsymbol{p} = \sum_{k=1}^{K} w_k(\mathbf{d}) P(\mathbf{h}, \mathbf{o}, \boldsymbol{\gamma}|\mathbf{d}, k) \tag{4.2}$$

The weight of each Gaussian $w_k(\mathbf{d})$ is given by

$$w_k(\mathbf{d}) = \frac{p_k p(\mathbf{d}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} p_j p(\mathbf{d}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{4.3}$$

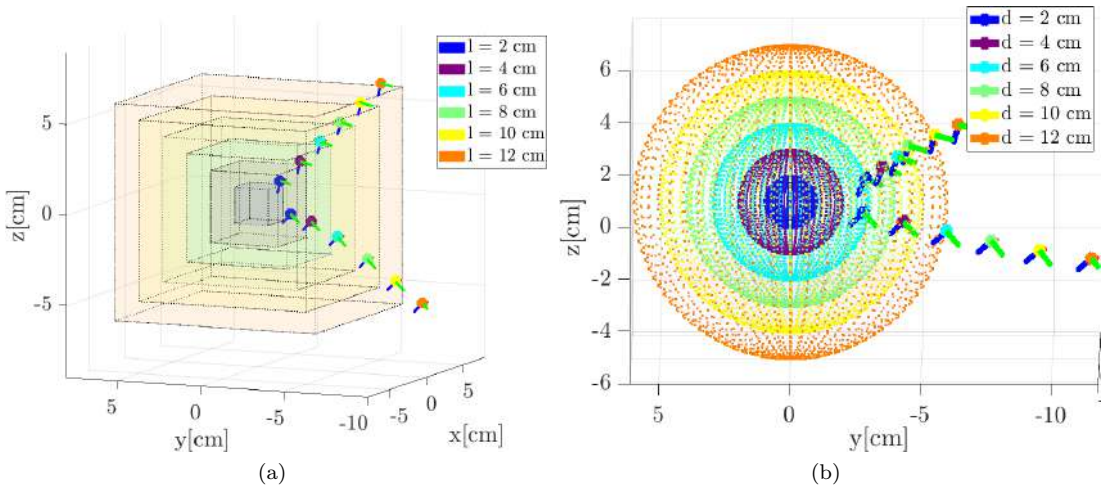
Similarly, it is important to remark that, while in the mixture of Gaussians in Eq. (5.1) the sum of the weights p_k must be equal to 1, that is not the case for the weights $w_k(\mathbf{d})$ in Eq. (5.2).

4.2.3 Model regressions

To compare the two implemented approaches, we evaluated the regressions of the obtained models through the Gaussian Mixture Regression.

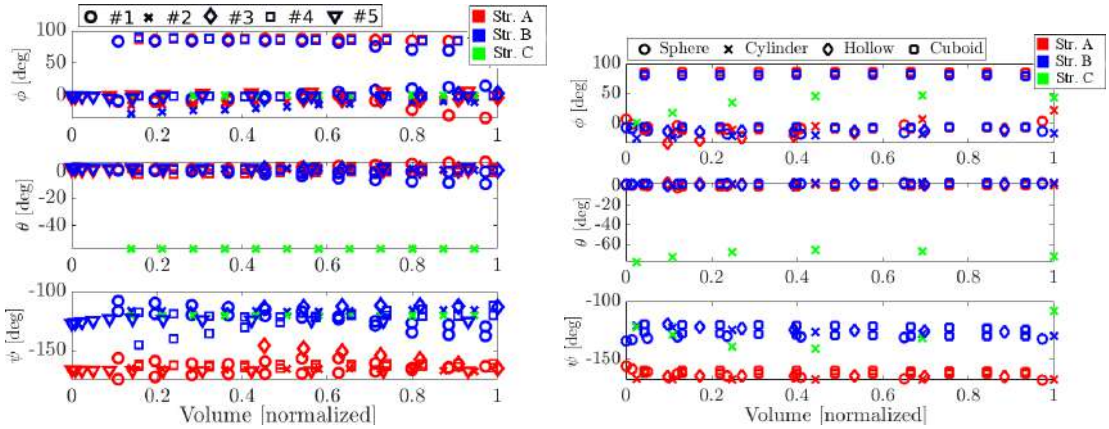
In Fig. 4.6, we reported a comparison between the pre-grasp poses of the spherical objects obtained in the two approaches.. As it can be seen in the figures, in both methods (*bounding box*-based and *shape*-based), the gripper tends to approach the object from two different directions. As the object size increases, the robotic gripper moves accordingly, always maintaining similar relative orientations with respect to the object. Furthermore, in the *bounding box*-based method, the poses are placed along the profile of the edges of the cuboids.

In Fig. 4.7, a comparison among all the orientations of each extracted primitive for each approach is presented. We made a regression through the GMR by varying only one dimension per object while keeping the others constant. Then, we computed the volumes and normalized them in the interval $[0, 1]$. The orientations shown in the figure are expressed through Euler angles and referred to the object reference frame $\{\mathbf{O}\}$. As it was already evident from the data shown in Fig. 4.5, the clusters of pre-grasp poses individuate different approach directions to the objects. By further analyzing the orientations of the interpolated poses (Fig. 4.7), we were able to determine three main strategies which are similar in both approaches and apply to different classes of objects. We will refer to these strategies as Str. **A**, Str. **B**, and Str. **C**. Notice that, although the obtained pre-grasp poses have similarities in their orientations, each pre-grasp outputted by the GMR is "unique", as it is strictly related to the object dimension $\mathbf{d}$ and the



(a) (b)

Figure 4.6: Comparison of the pre-grasp poses obtained using the two approaches. The obtained pre-grasp poses are referred with respect to the object reference frame {**O**}. The regressions are obtained using: (a) the *bounding box*-based approach (Class #5); b) the *shape*-based approach (spherical objects).



(a) (b)

Figure 4.7: Orientations comparison of the retrieved pre-grasp poses. The orientations are described as Euler angles (ϕ,θ,ψ) expressed in the object frame {**O**}. The regressions are obtained using: (a) the *bounding box*-based approach; b) the *shape*-based approach.

object class. We decided to unify the obtained pre-grasp poses under the three aforementioned strategies for the sake of presentation simplicity.

The three strategies can be compared considering the part of the object which is reached by the scoop. In Str. **A**, the object is approached from its upper part, whereas Str. **B** involves the object's lower part. In Str. **C**, instead, the SSG places the scoop on the side part of the object.

The ψ angle is related to the inclination of the scoop with respect to the object. In Fig. 4.7a

and Fig. 4.7b, we can clearly notice two approaching strategies (Str. **A** and Str. **B**) as we have two well-defined inclinations, independently of the object classes. The θ angle, instead, is related to the roll angle of the hand. It is kept around zero degrees for almost all the object classes; the only exception occurs for the cylindrical shapes and objects belonging to Class #2 (Str. **C**). Lastly, the ϕ angle is related to the direction of approach: angles around zero degrees correspond to an approach along the y-axis of the object, while 90 degrees angles are related to an approach along its x-axis. Objects belonging to Classes #1 and #4 (*bounding box*-based method) and cuboids (*shape*-based method) are approached either from the x or from the y direction, from the top or from the bottom, as shown in Fig. 4.5a, 4.5d, 4.5i. The K-means grouped four classes of primitives, but, taking into account the results in Fig. 4.7 we can consider that two of the primitives belong to Str. **A** and two to Str. **B**.

4.2.4 Trajectories reproduction for grasping execution

To devise an automatic grasping procedure, we employed further GMMs. From the previous pre-grasp poses analysis discussed in Sec. 4.2.3, three grasping strategies were found: **A**, **B**, and **C**. Consequently, we decided to group the temporal data of the cartesian poses $\mathbf{c}$ demonstrated by the operators into three clusters depending on the strategy.

As the time of each demonstration was different, we used Dynamic Time Warping [193] to adjust the dataset with a common time interval.

For each grasping strategy, we adopted a model Γ, which can be represented as:

$$P(\mathbf{c}, t|\Gamma) = \sum_{j=1}^{J} p_j p(\mathbf{c}, t|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \tag{4.4}$$

where J is the number of Gaussians.

Before training, we transformed the poses of each trajectory from the manipulator base frame to the pre-grasp pose frame in order to generalize the trajectories.

Once we trained the models, we fed the GMR with the desired time interval t. The retrieved trajectories were then re-transformed with respect to the manipulator base before being sent to the robot.

An example of the obtained trajectories for the cylindrical box is shown in Fig. 4.8. In the three graphs above, it is possible to observe the variation of the hand positions in the three strategies. In the top right graph, it is possible to clearly distinguish the different heights z of the SSG in the case of the three strategies. A marked distinction can also be observed in the orientation graphs, particularly in the trajectories of θ and ψ, which represent the roll and pitch angles of the scoop, respectively. Both in the Str. **A** and **B**, the θ angle is kept almost close to zero radians throughout the trajectory; vice versa, during the Str. **C**, it is possible to appreciate how the scoop rotates, reaching the maximum rotation near the pre-grasp pose. In the case of the ψ angle, we can see how in each strategy the scoop has different inclinations: from the least (Str. **C**) to the most tilted (Str. **A**). Finally, the trajectories of the ϕ angle and of the position components x and y depend on the object pose occupied within the workspace with respect to the manipulator base frame.

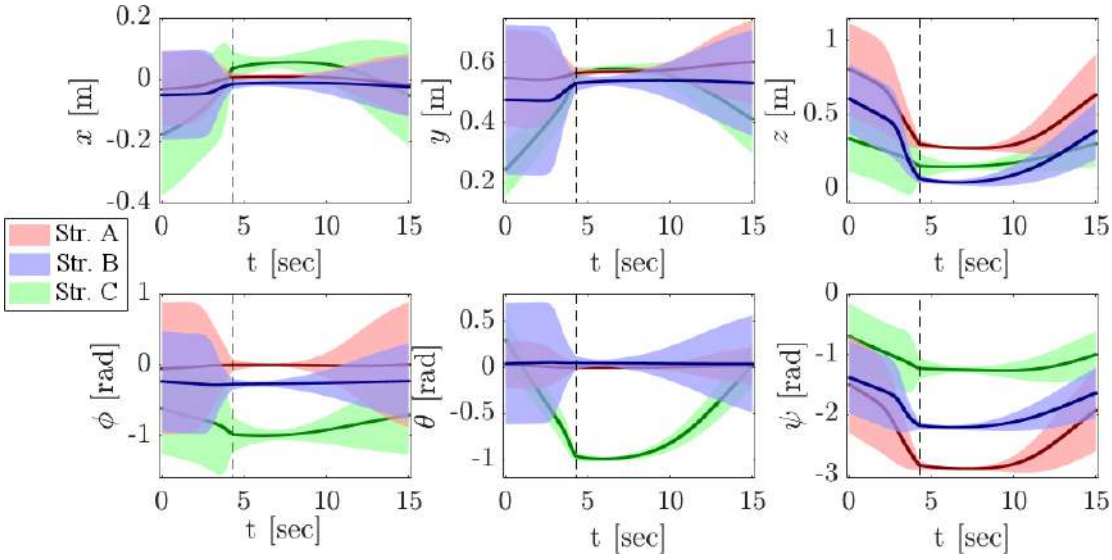


Figure 4.8: Grasping motions obtained for the cylindrical box. The end-effector poses are referred to the manipulator base frame. The red, blue, and green trajectories are related to the Str. **A**, **B**, and **C**, respectively. The dashed lines are relative to the time t when the robot is at the pre-grasp pose.

4.3 Experimental results

4.3.1 Experimental setup

A video of the experimental trials can be found at link.

The same experimental setting used to acquire the demonstrations (UR5 and Soft ScoopGripper) was adopted to exploit the learned skills in the autonomous grasp of unknown objects. The main aim of the performed experiments was to test the effectiveness of the retrieved strategies and compare their grasp success rates for the two different approaches described in Sec. 4.2.2. The 6-DoF manipulator was controlled in velocity and the Levenberg-Marquardt inverse kinematics solver available within the Kinematics and Dynamics Library (KDL) [194] was used to let the robot reach a certain pre-grasp pose. A Kinect One RGB-D camera was employed to detect the objects in the workspace and to retrieve their properties, i.e., bounding boxes or shapes.

In the *bounding box*-based approach, we exploited the pre-trained GMM to identify the bounding box class of tested objects. Whereas, for the *shape*-based approach, we trained a Convolutional Neural Network (CNN) to estimate the shape of the objects. For the training of the network, we provided as input the voxels of the objects used during the demonstrations phase (Fig. 4.3) and as output the labels of the corresponding shapes. The architecture we adopted for the network is composed by a 3D voxel grid ($32 \times 32 \times 32$) as input layer, two convolution layers composed of 32 filters (respectively $5 \times 5 \times 5$ and $3 \times 3 \times 3$), and a max pooling layer ($2 \times 2 \times 2$) followed by two dense layers (128 and 4). In principle, we could have used other techniques (e.g., K-nearest neighbor) to classify object shapes, however choosing the

Figure 4.9: Test objects (details in Table 4.3).

Table 4.3: Properties of test objects.

ID	Object	Size (mm)	Mass (g)
1	mango	$75 \times 95 \times 65$	253
2	plum (YCB)	$\varnothing 50$	28
3	plush	$140 \times 70 \times 140$	102
4	cylindrical box	$\varnothing 130 \times 250$	124
5	spray can	$\varnothing 50 \times 170$	118
6	banana	$\varnothing 35 \times 175$	66
7	scotch tape	$\varnothing 100 \times 40$	75
8	plastic mug (YCB)	$\varnothing 80 \times 70$	28
9	hollow box	$115 \times 130 \times 105$	21
10	mouse box	$40 \times 70 \times 105$	27
11	metal box	$70 \times 95 \times 70$	44
12	blue box	$75 \times 120 \times 150$	91

best classifier is beyond the scope of this work. The adopted CNN was derived from structures already used in literature [195] and worked for our setup.

To perform the experiments, we chose 12 novel test objects (Fig. 4.9, Table 4.3) and for each of them we tested all the possible applicable strategies. We performed 5 grasping trials for each strategy. Each trial consisted of the following steps: *i)* given the RGB-D image of the scene, we generated the 3D point cloud of the object, and the related voxel grid; *ii)* we fed the GMM/CNN to identify the object box/shape; *iii)* the object size $\boldsymbol{d}$ is inserted in the corresponding model Ω to retrieve the pre-grasp pose $\boldsymbol{p}$ (see Sec. 4.2.2); *iv)* the pre-grasp pose is inserted into the trajectory retrieved from the model Γ; *v)* the trajectory is sent to the robotic arm; *vi)* when the robot Cartesian pose coincides with the pre-grasp pose, a controller flexes the fingers until they touch the object and the scoop is actuated.

After the achievement of a grasp, we controlled the orientation of the gripper such that the surface of the scoop was always perpendicular to the gravity vector. In this way, the scoop could sustain most of the weight of the object, resulting in a more stable final configuration.

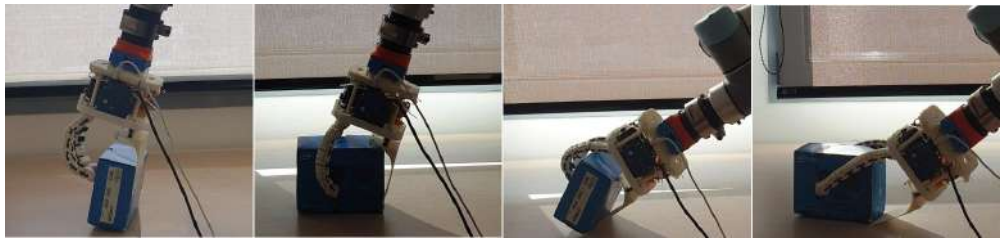


Figure 4.10: Grasps of the cuboids. From left to right: Str. **A** approaching along the object x-axis; Str. **A** approaching along the object y-axis; Str. **B** approaching along the object x-axis; Str. **B** approaching along the object y-axis;

Table 4.4: Grasp success rates.

		Bounding box-based				**Shape-based**			
ID	Object	Predicted class	Str. **A**	Str. **B**	Str. **C**	Predicted class	Str. **A**	Str. **B**	Str. **C**
1	mango	Class #5	2/5	3/5	–	Sphere	3/5	4/5	–
2	plum	Class #5	1/5	3/5	–	Sphere	2/5	4/5	–
3	plush	Class #4	10/10	10/10	–	Sphere	5/5	4/5	–
4	cylindrical box	Class #2	0/5	5/5	5/5	Cylinder	4/5	5/5	4/5
5	spray can	Class #2	3/5	5/5	4/5	Cylinder	3/5	5/5	4/5
6	banana	Class #4	4/10	4/10	–	Cylinder	2/5	1/5	4/5
7	scotch tape	Class #5	4/5	4/5	–	Hollow	4/5	4/5	–
8	plastic mug	Class #5	4/5	3/5	–	Hollow	4/5	5/5	–
9	hollow box	Class #3	4/5	5/5	–	Hollow	4/5	5/5	–
10	mouse box	Class #1	9/10	7/10	–	Cuboid	9/10	8/10	–
11	metal box	Class #5	4/5	5/5	–	Cuboid	9/10	7/10	–
12	blue box	Class #4	10/10	10/10	–	Cuboid	9/10	10/10	–
	TOTAL		55/80	64/80	9/10		58/75	62/75	12/15

We decided to test objects changing their orientation for each trial, but always placing the same face in contact with the table. We did this to avoid considering different classes for the same object (e.g., the standing cylindrical box belongs to Class #2, but lying down would be Class #4). A grasp was considered successful only if the object was carried to the final position without falling. Notice that since objects belonging to Classes #1 and #4 and cuboid shapes are characterized by 4 primitives, two belonging to Str. **A** and two to Str. **B**, (Sec. 4.2.3), we tested both approaching directions for each strategy. Examples are shown in Fig. 4.10.

4.3.2 Results

Obtained results in terms of grasp success rates are reported in Table 4.4 for the two approaches. The *bounding box*-based models have an effectiveness of about 75% (128 out of 170 tests). In particular, Str. **A** has an overall success rate of about 73% (55 out of 80), Str. **B** of 80% (64 out of 80), and Str. **C** of 90% (9 out of 10). However, Str. **C** can only be applied to Class #2. For the shape-based models, 132 out of 165 grasps were successful, with an overall grasp success rate of about 80%. Also in this case, Str. **B** (62 out of 75, 83%) is more effective than Str. **A** (58 out of 75, 77%), whereas Str. **C** (applied to cylindrical objects only) has a success rate of 80% (12 out of 15).

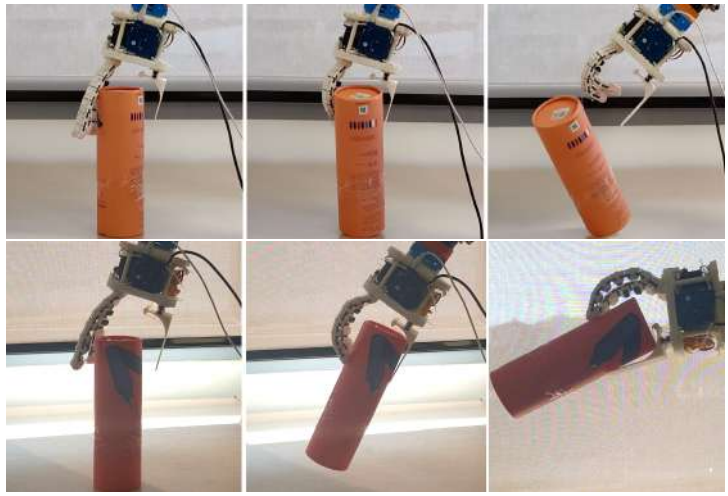


Figure 4.11: Str. **A** for the cylindrical box using (top) *bounding box*-based approach and (bottom) *shape*-based approach. Here the orientation of the fingers plays a fundamental role.

4.4 Discussion

Overall, classifying objects in basic shapes allows to obtain higher success rates than using simpler models like bounding boxes. However, in both approaches, extracted strategies present similar grasping success rates for the majority of the objects and the *bounding box*-based approach still presents a rather good performance. This confirms that, independently of the adopted object representation, the proposed extraction of primitives with GMM and GMR is effective, as long as the loss of information is minimal.

Analyzing experimental results, we derived advantages and limitations of the two approaches.

In the *bounding box*-based approach, the various objects classes were obtained through an unsupervised classification where we entirely neglected the shapes of the objects. Consequently, some objects (such as the cylindrical box and the plastic mug) have a lower success rate than with the *shape*-based approach. The cylindrical box is the only object the SSG could not grasp with the Str. **A** in the *bounding box*-based approach. The reason is mainly due to a retrieved incorrect fingers configuration. Fig. 4.11 shows a comparison of sequences of a successful and failed grasp in the case of the cylindrical box of *shape*-based and *bounding box*-based approach, respectively. Indeed, the ability to reorient the fingers represents a crucial factor for tall objects, which can be easily grasped in both approaches using Str. **B** because the fingers wrap around the object, resulting in more stable grasps (see Fig. 4.12-top). Alternatively, a solution can be represented by Str. **C** (Fig. 4.12-bottom).

In the case of the plastic mug, Str. **B** of the *shape*-based approach tended to insert the fingers in the object's cavity, reaching rather stable grasps, whereas Str. **B** in the *bounding box*-based approach did not exploit this feature (see Fig. 4.13).

The banana turned out to be a relatively hard to grasp object, mainly due to its size. Overall, the *shape*-based approach performed better than the *bounding box*-based one (46% against 40%). However, Str. **A** and Str. **B** of the *bounding box*-based approach proved to be more effective.

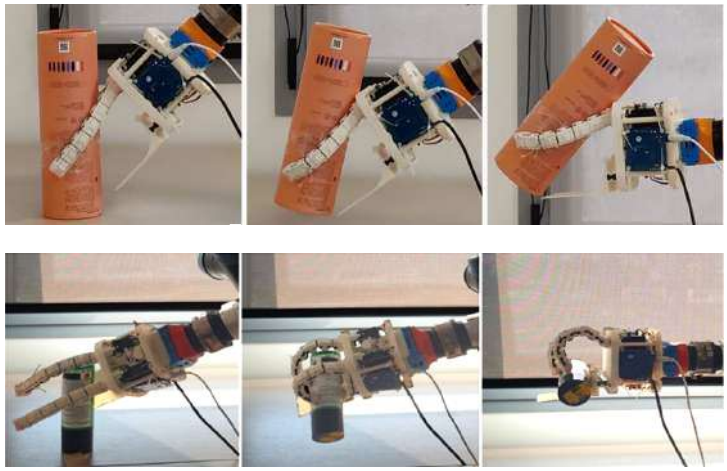


Figure 4.12: Str. **B** (top) and Str. **C** (bottom) used for cylindrical and Class #2 objects.

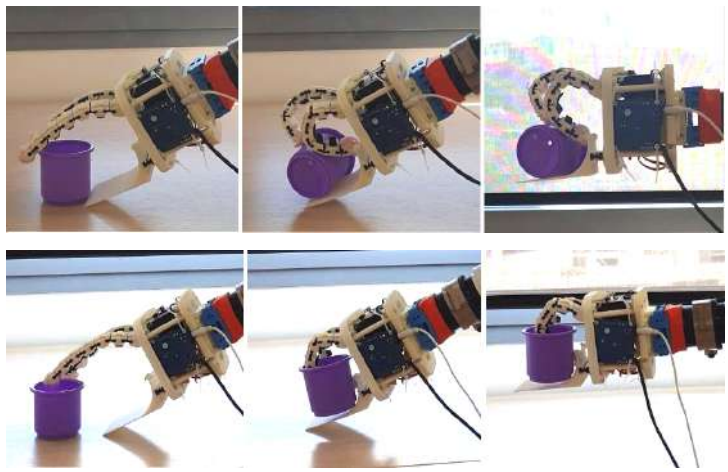
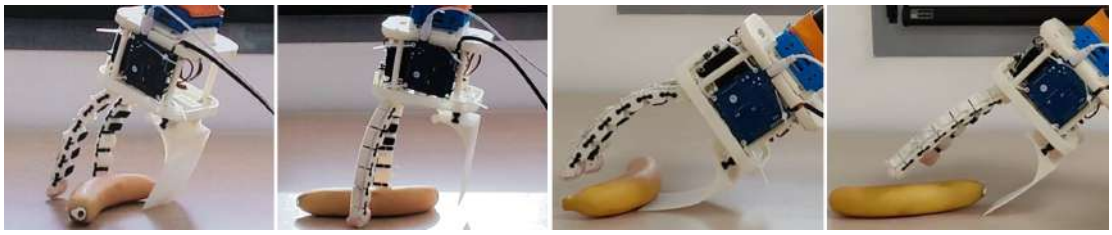


Figure 4.13: Str. **B** for the plastic mug using (top) the *bounding box*-based approach, and (bottom) the *shape*-based approach.

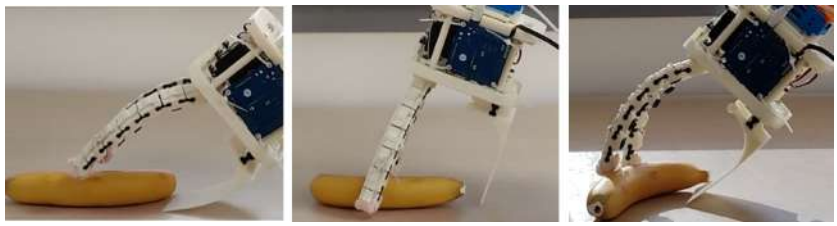
Note that, in the *shape*-based approach, the banana is recognized as a lying cylinder, and the pre-grasp poses are computed consequently. This can be considered as a strength of this approach, as the objects are classified independently of their pose. Fig. 4.14 reports a visual comparison of the strategies in the two methods.

Other two objects that resulted challenging to grasp in both approaches with Str. **A** were the plum (3 out of 10) and the mango (5 out of 10). Both objects require high precision in the positioning of the fingers, and when touched, they can easily incur involuntary rotations resulting in failures. Fig. 4.15 shows two examples of failed grasps. Nonetheless, we found that Str. **B** can mitigate the problem. In fact, in these specific cases, the success rates increase (7 out of 10 for both objects).

As a final consideration, notice that here we tested all the possible strategies for each object,



(a)



(b)

Figure 4.14: Grasps of the banana: (a) *bounding box*-based approach. From left to right: Str. **A** and **B** approaching along x and y directions. (b) *shape*-based approach. From left to right: Str. **A**, **B**, and **C**.

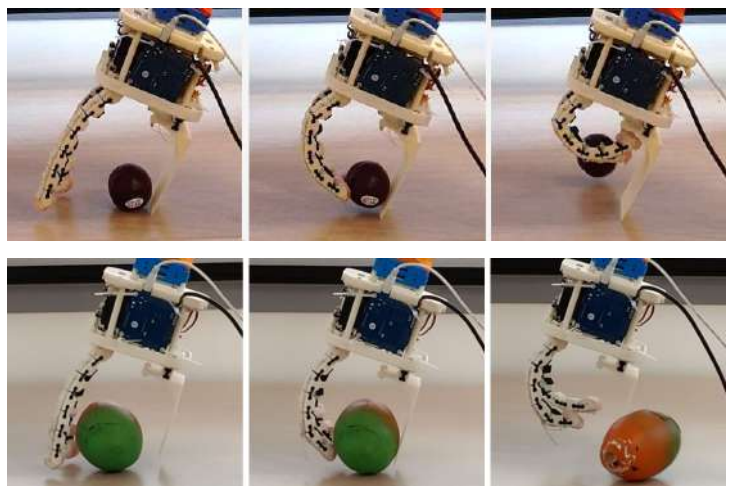


Figure 4.15: Str. **A** failures for plum (top) and mango (bottom).

but, in general, one could choose the most suitable strategy to adopt, based, for example, on the arm workspace limitations or on task-oriented considerations.

4.5 Conclusions

In this chapter, we proposed a method to obtain grasping strategies for non-anthropomorphic soft-rigid robotic grippers. First, we gathered data from human demonstrations asking to 6 users

to grasp multiple times 12 objects using kinesthetic guidance. Obtained data were processed in 2 different ways, assuming different levels of knowledge about the objects properties. We built and trained Gaussian Mixture Models with the aim of retrieving hand pre-grasp poses through Gaussian Mixture Regressions.

To validate our approach, we computed the regressions of the models over 12 novel test objects, and evaluated the grasp success rate of the learned primitives to compare the proposed approaches. Although they obtained similar success rates, experimental results suggest that it is preferable to search for grasp primitives taking the object shape into account. Leaving aside the fact that the *shape*-based approach requires the execution of an additional network for shape estimation, we can assert that the two approaches present a similar computational complexity. Indeed, the reason is that GMM and GMR complexity depends on the number of Gaussians chosen for the model representation [112], and, in our case, even if a different number of inputs characterizes the two approaches, the number of states is comparable for each primitive.

The presented methodology can be generalized to other soft grippers provided that suitable pre-grasp parameters are individuated. In this work, since the fingers of the chosen gripper are reconfigurable, we considered their orientation when training the models, but simpler grippers (e.g., with fixed fingers which only close in one direction), would not require additional parameters. In addition, we believe that simpler grippers could obtain satisfactory results even with not too specific object models (e.g., using the *bounding box*-based approach).

While in this work we showed the effectiveness of each primitive in terms of success rate, in future work, we plan to develop policies, possibly still based on learning from humans, to properly select the optimal primitive for a given object in a specific scene (e.g., in the presence of obstacles or in a cluttered environment). Future research will also evaluate the use of objects with more complex shapes, which could be approximated by the combination elementary shapes [196, 197].

Chapter 5

Grasping in clutter by exploiting embedded constraints using reinforcement learning

"*Anyone who stops learning is old, whether at 20 or 80.*
Anyone who keeps learning stays young. "
Henry Ford

Learning-based approaches applied to robotic manipulation have obtained relevant results, especially in object detection and grasp planning in unstructured environments, including cluttered and narrow spaces.

Adopting a design perspective aiming at embedding the robot "intelligence" more in the gripper hardware than in the software may represent another approach to tackle manipulation in complex scenarios. Several studies demonstrated that compliant elements in the hand structure enable advanced grasping strategies exploiting the hand-object-environment interactions.

In this chapter, we study how a deep reinforcement learning-based planner may exploit the mechanical intelligence of a soft-rigid gripper endowed with a scoop-shaped part in a decluttering task. Performed tests show that the proposed system can successfully empty previously unseen cluttered scenes, improving completion and grasp success rate with respect to a classic parallel-jaw gripper using the same planner. In addition, while rigid parallel grippers usually perform preparatory non-prehensile actions *before* grasping the target object, here, the used gripper can exploit environmental constraints and perform non-prehensile motions *during* the grasp action. This allows to significantly simplify the learning phase.

The chapter is organized as follows. Sec. 5.1 defines the problem of grasping in clutter, deriving its motivation. In Sec. 5.2, we describe the proposed methodology. In Sec. 5.3 Sec. 5.4, we report and discuss the experimental results. Lastly, Sec. 5.5 derives the conclusion of the study, outlining its possible further developments.

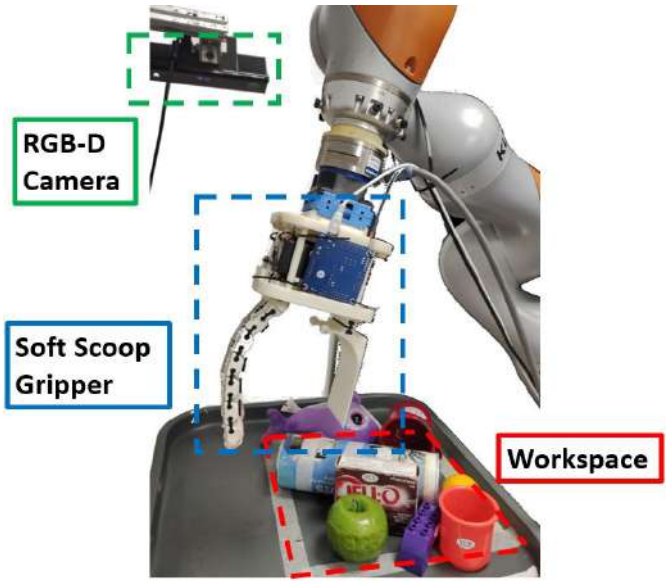


Figure 5.1: Experimental setup for decluttering. Red: workspace where the clutter is placed, green: RGB-D camera for object detection, blue: adopted soft-rigid gripper.

5.1 Introduction

Data-driven and learning-based approaches have lead to important results in robotic grasping and manipulation, above all when it comes to the proficient use of rigid parallel-jaw grippers and suction cups in unstructured environments [92, 198, 122]. At the same time, the development of a growing number of grippers with different structures, materials, and capabilities has shown that the robot design features can fundamentally help the grasping process [199, 10]. Compliant structures, for example, show superior capabilities in adapting to objects with different shapes, and can safely interact with their surroundings [4]. However, the interplay between learning algorithms and gripper design has rarely been studied since most of the works presenting learning-based approaches rarely employ more complex, possibly soft, devices.

In this chapter, we aim at answering one main research question: Can the gripper design "help" the learning process when dealing with complex grasping problems? In other words, can we leverage the gripper "embedded mechanical intelligence" to simplify the learning process?

To study this problem we decided to focus on a specific task, i.e., removing all objects from a cluttered scene (decluttering), and to use a specific gripper, i.e., the Soft ScoopGripper (SSG) [9]. The SSG has two main features which differentiate it from a standard parallel gripper: it has passively compliant joints and a rigid embedded scoop-like structure which allows the proficient exploitation of environmental constraints. The decluttering plan is obtained adapting a Reinforcement Learning (RL) framework for parallel grippers [8] to the SSG. Decluttering is achieved by exploiting prehensile and non-prehensile actions and using environmental constraints exploitation strategies. The scenario tackled in this chapter is shown in Fig. 5.1.

Results obtained on real cluttered test scenes show that, thanks to the gripper design, non-

prehensile actions can be performed *during* grasping, and not as pre-grasp preparatory adjustments as in previous works [8, 127]. This allows to use a simpler learned policy based on a single action, instead of two (or more) actions. In addition, from a qualitative comparison between the results obtained in [8] and those obtained with the chosen soft-rigid gripper, it emerges that the gripper design impacts the performance of the learning process and can lead to an improvement in terms of decluttering completion rate.

5.2 Materials and Methods

5.2.1 Problem Formulation and State Representation

The problem of decluttering a scene is formulated as a Markov decision process (MDP). At time t, at any state s_t, the agent chooses and executes an action a_t according to a policy $\pi(s_t)$. Once reaching a new state s_{t+1}, the agent receives a corresponding reward $R_{a_t}(s_t, s_{t+1})$. The aim is to find an optimal policy π^* able to maximize the sum of the future rewards over an infinite-horizon, given by $R_t = \sum_{i=t}^{\infty} \gamma R_{\alpha_i}(s_i, s_{i+1})$, where γ is the future discount.

An off-policy Q-learning was adopted to train a greedy policy $\pi(s_t)$. This policy chooses the actions by maximizing the Q-function $Q_\pi(s_t, a_t)$, i.e., the measurement of expected reward for selecting an action a_t in a certain state s_t at time t.

Let us define the temporal difference error δ_t between $Q_\pi(s_t, a_t)$ and a fixed target value y_t:

$$\delta_t = |Q(s_t, a_t) - y_t|$$
$$y_t = R_{a_t}(s_t, s_{t+1}) + \gamma Q(s_{t+1}, \arg\max_{a'}(Q(s_{t+1}, a')))$$

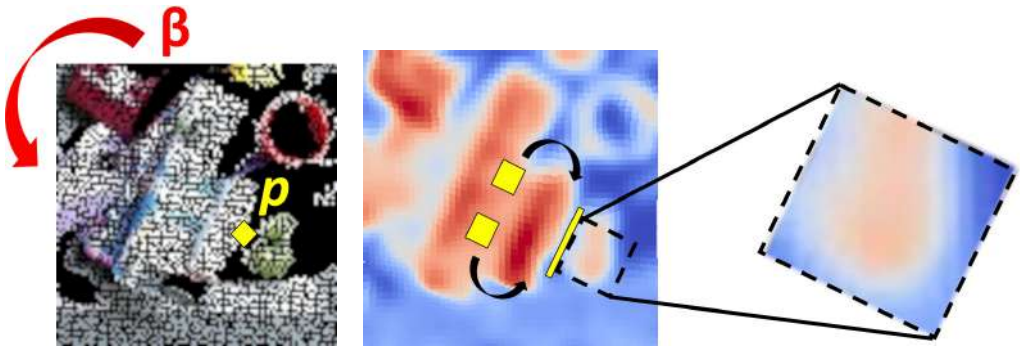
where a' is the set of all available actions. The training phase is based on iteratively minimizing δ_t.

Similarly to what is done in [8], we modeled each state s_t as a heightmap of the cluttered scene at time t. RGB-D images which come from a fixed-mount camera were firstly projected onto a 3D point cloud, and then re-projected to give rise to two heightmap images representated in both channels (RGB and depth). We determined the size of the heightmap taking into account the robot reachability to execute the actions. In our case, we constrained the workspace to an area of 0.3×0.3 m^2, leading to heightmaps of 100×100 pixels. Hence, each pixel spatially represents a 3D space of 3×3 mm^2 in the agent's workspace. The workspace is shown in Fig. 5.1.

5.2.2 Action Definition

Starting from the state representation s_t (i.e., the heightmap of the scene), we defined the actions for the SSG as motion primitives μ. We back-projected a given pixel p selected by the network from the heightmap into a 3D location q, where the primitives are then executed. An example of a registered heightmap is shown in Fig. 5.2.

We defined two actions, i.e., pushing and grasping (Fig. 5.3 and Fig. 5.4, respectively). In both actions the robot positions the gripper at the 3D location q with an orientation angle β



(a) (b)

Figure 5.2: Examples of registered heightmaps. (a) Color heightmap that contains the RGB channels and is fed to the network. The best pixel p is in yellow and the angle to rotate the image β is in red. (b) Depth heightmap (as cloned depth channels DDD) together with the hand mask (in yellow) to determine the fingers rotation (black arrows); the area behind the scoop is delimited by a dashed-line square. The developed model-based algorithm evaluates the density of clutter in that square to determine the hand inclination during grasping.

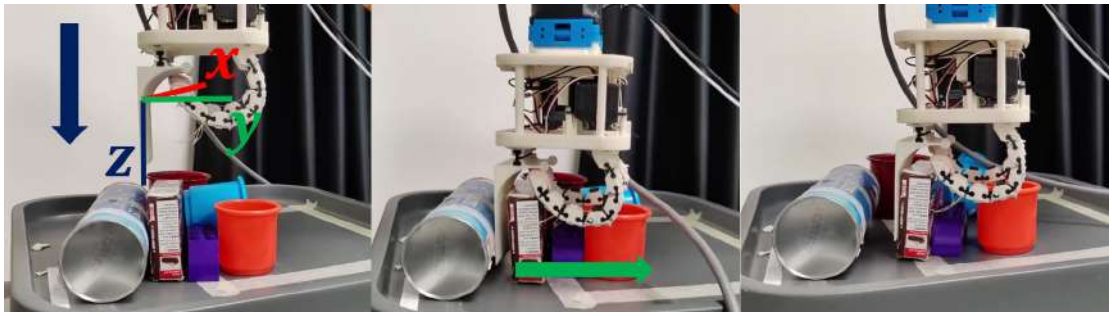


Figure 5.3: Push action: in the first frame we represented the scoop reference frame with x, y and z in red, green and blue, respectively. The thick blue and green arrows represent the motion of the gripper necessary to perform the push.

determined by the network (Fig. 5.2a). The angle β expresses a rotation of the heightmap and is discretized in intervals of 22.5°, resulting in 16 different orientations.

To achieve a push, the SSG fingers are initially closed to avoid possible hindrance during the action. Then, a hybrid force-velocity controller which mantains a constant force on the scoop z-axis is used, while the scoop follows a straight trajectory along its y-axis, as shown in Fig. 5.3.

To perform a grasp, the gripper is first positioned above the grasping location. Hence, the scoop starts descending following a straight trajectory. The fingers are closed when the gripper reaches the table surface and the force sensed at the robot wrist exceeds a certain threshold. The scoop inclination α with respect to the surface where the objects lie is controlled based on the scene state. In particular, a model-based algorithm that analyzes the heightmap behind the scoop to determine if it is obstacle-free or cluttered was developed. In the first case, the

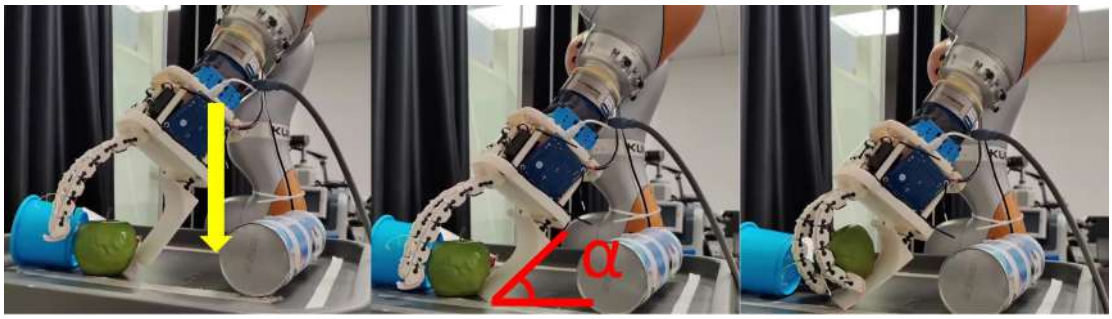


Figure 5.4: Grasp action: the yellow arrow represents the motion of the gripper towards the table after having reached the 3D position q, while α (in red) represents the inclination of the SSG if the area is obstacle-free.

scoop is tilted with a predefined angle[1] that improves the grasping, otherwise the scoop is kept orthogonal to the table.

Additionally, the algorithm selects the orientation of the fingers. To do that, we defined a 2D mask of the gripper, which is positioned at the grasp location on the heightmap (Fig. 5.2b). If a contact between the tip of a finger and an object is detected, the finger is rotated incrementally of a fixed angle until the collision is avoided.

5.2.3 Learning Algorithm

The learning algorithm has been implemented as in [8].

To train the policy $\pi(s_t)$, we defined a fully convolutional network (FCN) for each action (ϕ_g for grasping and ϕ_p for pushing). Then, we modeled the Q-function using the two networks.

Both FCNs take as input the heightmap images representation of the state s_t and generate a pixel-dense map of Q values with the same size and image resolution of the input image. Each Q value prediction at a pixel p represents the expected future reward of executing primitive μ at the 3D position q, where $p \in s_t$.

Each FCN consists of two 121-layer DenseNet [200] pre-trained on ImageNet [201], followed by channel-wise concatenation and 2 additional 1×1 convolutional layers interleaved with nonlinear activation functions (ReLU) and spatial batch normalization. The first DenseNet takes as input the RGB heightmap, whereas the other one takes the normalized depth heightmap. For each FCN, the input heightmaps are rotated in 16 different orientations, resulting in pixel-wise maps Q values as output (16 for pushing and 16 for grasping). The executed action a'_t is the primitive μ which maximizes the Q-function at a pixel p corresponding to the highest Q value among all pixel-wise maps. For what concerns rewards, a reward $R_g(s_t, s_{t+1}) = 1$ is assigned if a grasp is successful. To evaluate the success or failure of a grasp, we kept track of the torque value at the fingers' motor. If a certain threshold was overcome, the grasp was deemed successful, as it meant that the object was firmly held until the end of the grasping action. In pushing, we assigned a reward $R_p(s_t, s_{t+1}) = 0.5$ if the algorithm detected significant differences between the states at time instants t and $t + 1$, making a comparison between two consecutive heightmaps.

[1] In this case, we chose $\alpha = 45°$ based on the experience gained in the previous chapter.

The FCNs were trained using the Huber loss function.

$$\mathcal{L}_i = \begin{cases} \frac{1}{2}(Q^{\theta_i}(s_i, a_i) - y_i^{\theta_i^-})^2, \text{ for } |Q^{\theta_i}(s_i, a_i) - y_i^{\theta_i^-}| < 1 \\ |Q^{\theta_i}(s_i, a_i) - y_i^{\theta_i^-}| - \frac{1}{2}, \text{ otherwise} \end{cases}$$

where θ_i are the parameters of the neural network at iteration i, and the target network parameters θ_i^- are held fixed between individual updates. Only if the pixel p and the network ϕ are responsible for the executed action a_i the gradients pass through them. Otherwise, the pixels backpropagate with 0 loss. We adopted stochastic gradient descent with momentum to train the FCNs, with fixed learning rates of 10^{-4}, momentum of 0.9, and weight decay 2^{-5}. We used stochastic rank-based prioritization, approximated with a power-law distribution, to exploit prioritized experience replay [202]. The exploration strategy we adopted is considered ϵ-greedy, where ϵ is initialized at 0.5 and annealed to 0.1, while the future discount γ is constant at 0.5.

The models are trained in PyTorch with an NVIDIA GeForce 3060 on an Intel CORE i7-11800H clocked at 2.30 GHz.

The FCNs were trained with data gathered with a real experimental setup, not with simulation. The motivation for this choice lies in the complexity of accurately reproduce in a dynamic simulator the physical interaction between a soft underactuated gripper and its surroundings, especially in cluttered environments. Details on the training setup and results are reported in Sec. 5.3.1 and Sec. 5.3.2, respectively.

5.3 Experimental results

5.3.1 Setup

A video of the experimental trials can be found at link.

We executed several experimental trials first to train and then to test two policies, one using only the grasping primitive (grasp-only) and one with both grasping and pushing actions (push-grasp). We then analyzed the obtained results to investigate the performance of a soft-rigid gripper i) in the training phase, and ii) in the test phase.

To collect training data and then to test the trained policies, we adopted the same experimental setup including an LBR iiwa 7-DOF robot arm (KUKA AG) equipped with an ATI Gamma 6-axis force/torque sensor (ATI Industrial Automation, Inc.) at the wrist and the Soft ScoopGripper. A Kinect One RGB-D camera (Microsoft), statically mounted on a fixed shaft, was used to overlook the tabletop scenario and capture images of resolution 960×540. The setup is shown in Fig. 5.1.

A total of 28 objects were used for training and testing, most of them coming from the YCB dataset [185]. Their properties are listed in Table 5.1. They were arranged in different ways to compose training and testing scenes.

Table 5.1: Dataset of objects.

	Object	Weight (g)	Size (mm)
Training objects	apple (YCB)	68	⌀75
	peach (YCB)	33	⌀64
	orange (YCB)	47	⌀63
	strawberry (YCB)	18	⌀44 × 55
	tennis ball (YCB)	58	⌀65
	baseball ball (YCB)	191	⌀96
	dolphin plush	84	80 × 200 × 90
	batman plush	102	140 × 70 × 140
	small cylinder	35	⌀40 × 130
	candy tube	12	⌀36 × 225
	scotch tape	75	⌀100 × 40
	spray can	118	⌀50 × 170
	chips can (YCB)	205	⌀75 × 250
	blue cup (YCB)	31	⌀90 × 74
	red cup (YCB)	21	⌀75 × 68
	metal mug (YCB)	90	⌀90 × 80
	pudding box (YCB)	187	35 × 110 × 89
	gelatin box (YCB)	97	28 × 85 × 73
	spring clamp (YCB)	59	90 × 115 × 27
	lego brick (YCB)	2.5	95 × 35 × 40
	plastic funnel	120	⌀120 × 130
Test objects	banana (YCB)	66	⌀35 × 175
	tuna fish can (YCB)	171	⌀85 × 33
	body spray	34	⌀45 × 175
	green cup (YCB)	17	⌀65 × 64
	mouse box	27	40 × 70 × 105
	wooden block	374	140 × 63 × 43
	mouse box	146	90 × 175 × 60

5.3.2 Results from Training

In the training phase, we let the robot declutter several different scenes, each including 12 out of the 21 training objects listed in Table 5.1. The objects in the scene were randomly chosen and unsystematically placed in the robot workspace. Examples of training scenes are shown in Fig. 5.1, Fig. 5.3, and Fig. 5.4. The scene was changed either when all objects were successfully removed, or when the robot actions failed 10 times in a row.

The training curve of the two adopted strategies (grasp-only and push-grasp) is shown in Fig. 5.5. For comparison, we also report the final success rates extracted from the learning curves obtained in [8].

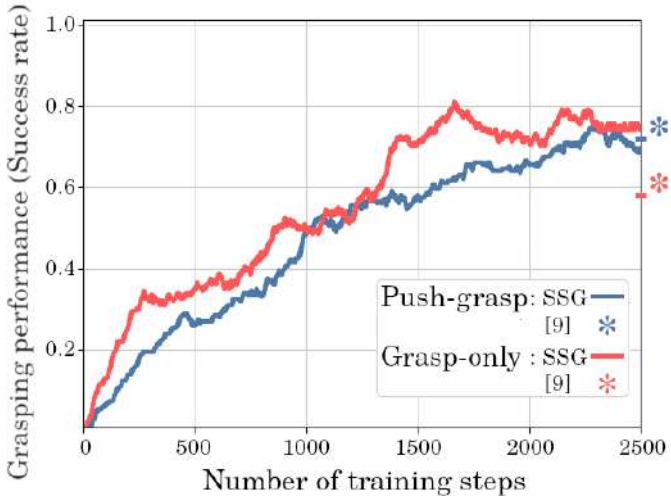


Figure 5.5: Solid lines: learning curves obtained with the SSG in the training phase considering only the grasping action (red) or both push and grasping actions (blue). Stars: final success rate obtained in the learning curves reported in Fig. 5 of [8] for grasp-only (red) and for push-grasp (blue).

5.3.3 Results from Testing

To test the trained networks we used the 4 cluttered scenes reported in Fig. 5.6. Similarly to [8], test scenes were composed either of completely new objects (Fig. 5.6a, Fig. 5.6b) or of objects used also in the training phase (Fig. 5.6c, Fig. 5.6d). It should be noted that since the number and the arrangement of the objects are completely different between training and testing, and since the whole learning algorithm is not based on the recognition of the single object but on the overall heightmap, this choice did not bias the results (see also Sec. 5.4).

Each test scene was tackled 10 times with the grasp-only network and 10 times with the push-grasp network, reaching a total of 80 test trials. A trial ended either when the scene was completely decluttered or when there were 10 failures in a row[2].

Similarly to [8], the performance of the two networks was evaluated with 3 metrics: completion rate, action efficiency, and grasp success rate. The first one is computed as the ratio between the number of objects grasped before the end of the trial and the total number of objects. The second one is the number of grasped test objects over the number of actions performed until the end of the trial. The third one consists of the number of successful grasps over the number of performed grasp actions. Thus, in grasp-only policies, action efficiency and grasp success rate are equivalent. Obtained results are shown in Table 5.2. The average values across different trials for the same scene are reported in rows 2-5 of Table 5.2, whereas the average values among all trials for all scenes are in the last row.

[2]A certain action fails according to the criteria described in Sec. 5.2.3.

(a) Test Scene 1

(b) Test Scene 2

(c) Test Scene 3

(d) Test Scene 4

Figure 5.6: Cluttered scenes adopted during the testing phase. Top: clusters made of new objects, bottom: clusters made of objects used also in training scenes.

5.4 Discussion

Looking at the obtained learning curves for the two networks (Fig. 5.5), it is possible to study whether introducing soft grippers in cluttered scenarios can influence the learning process. Although the objects used in this chapter differ from those used in [8], we believe a comparison with this previous work is anyway interesting, also considering that we used the same network structure and objects with more complex shapes. From Fig. 5.5, several insights can be gained. Firstly, by considering the final success rates of the two strategies proposed in this chapter, we noticed that both networks (blue and red lines) exhibit success rates around 70-75%, similarly to the Visual Pushing-Grasping (VPG) strategy proposed in [8] (blue star). Conversely, the grasp-only strategy by Zeng et al. [8] (red star), reaches a final success rate of less than 60%. Even though absolute values cannot be directly compared, there is a clear trend showing that while with the SSG both strategies perform similarly, when using a parallel jaw gripper the addition of the push action significantly helps changing the environment state to free the constrained

Table 5.2: Results of the testing phase. All results are presented in percentage. Rows 2-5 represent the results obtained for each scene of Fig. 5.6, whereas the last row reports the average values.

Test Scene	Strategy	Completion rate	Action efficiency	Grasp success
1	Push-grasp	75.3	69.1	82.8
	Grasp-only	**82.0**	**85.1**	**85.1**
2	Push-grasp	68.9	65.4	78.5
	Grasp-only	**79.0**	**84.6**	**84.6**
3	Push-grasp	74.5	68.3	82.4
	Grasp-only	**81.4**	**85.4**	**85.4**
4	Push-grasp	74.9	66.0	81.5
	Grasp-only	**80.4**	**87.3**	**87.3**
All	Push-grasp	73.4	67.2	81.3
	Grasp-only	**80.7**	**85.6**	**85.6**

objects. The additional action can be avoided when the gripper embeds "intelligent" components in its design. Specifically, in this chapter, a rigid, passive element, the scoop, attached to the hand via a flexible joint allows the robot to easily move the objects and/or adapt to them directly during the grasping action.

Another insight we get from Fig. 5.5 is related to the learning rate of the two networks. Comparing our two strategies, we noticed that the network trained using the grasp-only action learns faster than the one trained with push-grasp: the success rate settles in a range of 10% with respect to the final value at around 1500 and 2300 episodes, respectively. The reason is that the hand can grasp objects directly inside the cluttered scene, thanks to the reconfigurability of the fingers and its capability of separating objects easily during grasping. This peculiarity allows the network to use a single effective strategy and, consequently, to learn faster than a two-actions network.

Concerning the results of the test trials, as it can be seen from Table 5.2, all scenes present similar results. However, test scene #2 records the lowest values of the three metrics. We hypothesize that in this particular scene, it was harder for the network to discriminate the edges of the objects as they have similar heights. Also, the wooden block is rather heavy: so it was very difficult to grasp.

Overall, the grasp-only strategy outperforms the push-grasp network in all the adopted metrics. Especially the action efficiency disparity between the two strategies is remarkable. The number of necessary actions in the grasp-only trials is lower with respect to the push-grasp trials. This happens because the hand can directly grasp the objects, even in rather cluttered scenarios, without using non-prehensile actions. Conversely, the push-grasp strategy necessitates of more actions to declutter scenes completely. A similar reasoning holds when comparing the completion rates, where it is possible to see that the grasp-only policy leads to the complete clearing of the scene more frequently with respect to the push-grasp strategy, which tends to

fail more often because of the inefficiency of the push action. We also noticed that, differently from [8], where the grasp-only performed worse than the push-grasp in real-world experiments, here the grasp success rate is comparable for both strategies. This is due to the embedded intelligence of the hand, which, thanks to its reconfigurable soft fingers and its scoop, can grasp objects even without a preparatory push action.

Lastly, it can be noticed that the results obtained by the push-grasp strategy are comparable to the ones obtained with the VPG strategy in the real-world tests performed in [8], even if the objects used here have more complex shapes.

5.5 Conclusions

In this chapter, we integrated a soft-rigid gripper able to safely and effectively interact with the environment, the Soft ScoopGripper, with an active exploration algorithm to empty a cluttered scene. We trained two reinforcement learning policies. The first was characterized by pushing and grasping actions, while the second relied only on the grasping action. Then, we tested the obtained policies in challenging cluttered unknown scenes.

Comparing the results obtained in [8] and those obtained using the SSG, we show that integrating features like softness, embedded constraints, and model-based algorithms can affect the learning process, influencing the performance of the implemented strategies. We also show that, with the SSG, a simpler policy (i.e., grasp-only) learns faster and performs better in test scenes than the two-actions policy (i.e., push-grasp).

Considering the work presented in Chapter 6 proposing to add scoop-like structures to commercial soft grippers, we will work on developing a RL framework that can be applied to any gripper that embeds a scoop, making the scoop the only necessary element for transfer learning. Future research will also focus on developing policies to fully exploit environmental constraints when non-prehensile strategies (e.g., pushing) are not applicable, as in restricted scenarios. Thus, we will also include multimodal perception during the learning phase to understand to which extent the gripper can interact with its surroundings.

Part II

Optimal Design Techniques for Embedded Constraints

Chapter 6

Automated design of embedded constraints for soft robotic hands

"*A man who asks is a fool for five minutes.*
A man who never asks is a fool for life"
Chinese proverb

Soft robotic hands allow to fully exploit hand-object-environment interactions to complete grasping tasks. However, their usability can still be limited in some scenarios (e.g., restricted or cluttered spaces). In this chapter, we propose to enhance the versatility of soft grippers by adding special passive components to their structure, without completely altering their design, nor their control. A method for the automated design of soft-rigid scoop-shaped add-ons acting as "embedded constraints" is presented. Given a certain gripper and a large set of objects, the design parameters of the optimal scoop for each object are derived by solving an optimization problem. Also the object-environment relative pose is considered in the optimization. The obtained "optimal scoops" are clustered to get a limited set of representative scoop designs which can be prototyped and used in grasping tasks. In this work, we also introduce a data-driven method allowing a grasp planner to select the most suitable scoop to be added to the used hand, given a certain object and its configuration with respect to the surrounding environment. Experiments with two different hands validate the proposed approach.

The chapter is organized as follows. Sec. 6.1 defines and motivates the study. Sec. 6.2 describes the implemented methodology, whereas Sec. 6.3 and Sec. 6.4 report and discuss the experimental results. Lastly, Sec. 6.5 derives the conclusion of the study, outlining the future works.

6.1 Introduction

Designing robotic grasping systems capable of handling several different objects in unstructured and possibly cluttered environments still remains an open challenge. The *soft manipulation* approach has proposed to tackle it by adopting "an integrated, interdisciplinary view of manipulation" [10], where the hardware design and the software to use it are co-developed to fully unlock the potential of soft hands.

Thanks to passively compliant elements included in their structure, soft robotic grippers can safely comply with objects and environmental constraints, allowing the implementation of novel, human-inspired, grasping strategies which deliberately and effectively exploit the hand-object-environment interactions [53, 50]. The hand design can be enriched by adding soft-rigid structures which facilitate these interactions. Salvietti et al. [9], for example, proposed to attach a sort of scoop to the hand palm to easily slide between surfaces to grasp the desired item, whereas Gafer et al. [5] attached magnetic nails to soft fingers to better slide underneath deformable fragile objects.

In this chapter, we propose an automated procedure for designing soft-rigid scoop-shaped passive inclusions that can be added to commercial or custom grippers to enable new grasping strategies and enlarge the set of graspable objects (see Fig. 6.1). The add-ons could be particularly useful in cluttered or restricted environments, where the exploitation of environmental constraints is a crucial, but rarely considered, component [23].

Figure 6.1: Adding a scoop-shaped add-on to a soft hand: pre-grasp configuration, and final grasp. The object cannot be grasped from the top without the scoop, as it is too large. The positioning and geometrical features of the scoop are the result of an automated design approach.

Usually, works on grasping in cluttered and narrow spaces focus on vision problems and present algorithms to derive feasible grasps from the image of the scene [203, 204], or to effectively separate the target object from the clutter [205, 127]. Close to the vision and planning software, also the robot design can play a fundamental role in achieving successful grasps in challenging environments [206, 66]. In this work, we do not propose to fully re-design the hand, but to

modify the structure of existing grippers to achieve a grasp strategy that uses the environment as an help for the grasping process. In this way, not only the design, but also the planning phase is simplified, as just by changing a part of the hand, several different objects can be grasped using the same scoop grasp strategy.

In Chapter 3, we proposed a new strategy, called *scoop grasp*, to simplify the grasping procedure when objects are constrained from one or two sides. This strategy can be effectively applied only to hands that embed a suitably designed constraint in their structure. Here we introduce a procedure for the automated design of interchangeable scoops that allow to perform scoop grasps with any kind of grippers.

Given a certain robotic hand, the proposed approach follows three main steps. First, the optimal design parameters of the scoop-like structures to be added to the chosen initial hand are derived through an optimization algorithm. The latter is based on quasi-static grasp analysis and computes an "optimal scoop" given the object to be grasped and its relative position with respect to environmental constraints (e.g., wall, table). Thus, if the optimization is conducted over several different objects, the number of obtained solutions is the same as that of objects, and each of them would correspond to a different scoop.

As second step, to reduce the number of possible scoops for a certain gripper, we used an approach based on Gaussian Mixture Model (GMM) [110] to cluster scoops with similar characteristics. Lastly, the final extracted scoops were tested in several scoop grasp experiments aimed at collecting data to train a Convolutional Neural Network (CNN) able to evaluate which scoop should be mounted for a given object.

The obtained CNN was validated with two different soft hands, the Soft ScoopGripper [9] and the Pisa/IIT SoftHand [43].

We demonstrated that even robotic hands which do not have embedded constraints in their initial design (e.g., the Pisa/IIT SoftHand) can benefit from these additions, enlarging the set of graspable objects and of possible grasping strategies. The proposed design method is applicable to any robotic hand, since the scoop plays the role of an extension of the palm, that can be easily removed and does not add any requirement from the control point of view. Differently from previous work on automated design, we explicitly consider environmental constraints in the design optimization process, as we embrace the idea that hand-object-environment interactions should be fully exploited to achieve robust grasps [207, 23].

6.2 Materials and Methods

Fig. 6.2 shows the block diagram of the automated design process leading to the prototyping of a set of scoop-like embedded constraints for the two adopted hands. After a brief description of the used robotic hands, we detail each step of this process in Sections 6.2.2, 6.2.3, 6.2.4.

To apply the obtained scoops in grasping tasks, we trained a CNN for each adopted hand. Given the RGB-D image of the object and the environmental constraint it is in contact with, the CNN outputs the estimated grasp success-rate for each scoop associated to the chosen hand. This process is shown in Fig. 6.3 and detailed in Sec. 6.2.5.

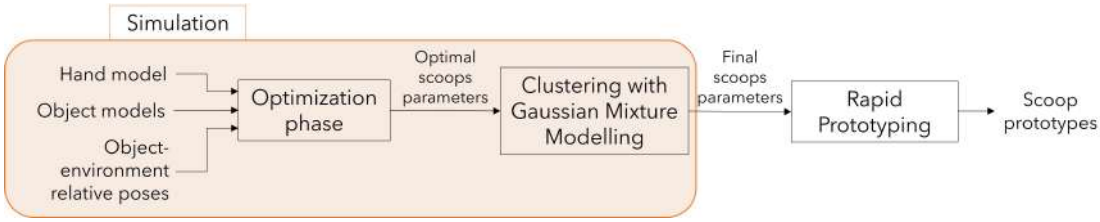


Figure 6.2: Block diagram showing the automated design process for embedded constraints (scoops). Given a certain hand, a set of objects and their relative poses with respect to an environmental constraint, a set of optimal scoops is obtained. To reduce the number of scoops we cluster them with a GMM-based approach.

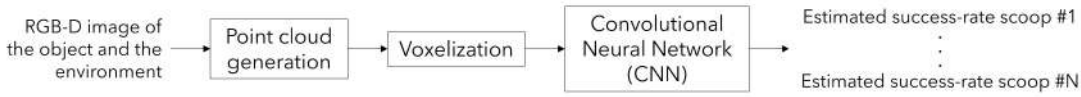


Figure 6.3: Block diagram showing the working flow of the CNN which estimates the grasp success-rate of the designed scoops for a given input object/environment pair. A different CNN has been trained for each hand.

6.2.1 The Soft ScoopGripper and the Pisa/IIT SoftHand

We chose these two hands to show the effect of adding the newly designed scoops to both, a hand that is natively designed to incorporate an embedded constraint, and a hand that is not. While the SSG structure is specifically designed for exploiting environmental constraints thanks to its embedded constraint, the Pisa/IIT SoftHand is an anthropomorphic soft hand that is capable of interacting with the environment thanks to its compliance. This hand is designed to be able to perform robust, human-like power grasps.

When applied to the SSG, the newly designed scoops allow to extend the range of graspable objects. Even if the scoop can be closed towards the fingers through a tendon-driven mechanism actuated by another motor [9], in this case, we did not use this actuation because our purpose is to add a passive extension of the palm.

When applied to the Pisa/IIT SoftHand, the scoops not only enlarge the space of graspable objects, but also allow to use a simple top-grasp approach even in the presence of obstacles or when objects cannot be reoriented to exploit other strategies (e.g., slide-to-edge grasps).

6.2.2 Optimization phase

The first step of our method consisted in finding the optimal parameters to design the scoops and position them into the structure of the chosen hands. To this aim we developed a simulation framework using the SynGrasp MATLAB Toolbox [48]. We modelled the kinematics of the two adopted hands and we implemented the simulation of quasi-static grasps of 50 different objects from the YCB Dataset [185]. A top-grasp was iteratively generated by closing the simulated hands over the object model until the optimal combination of decision variables was selected by the solver. The object model was obtained from the object point cloud using the Crust

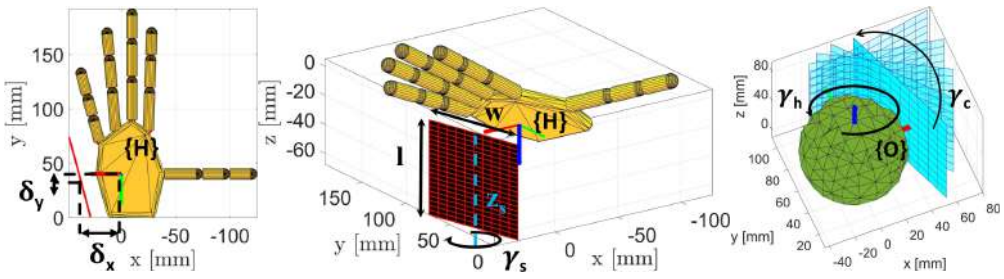


Figure 6.4: Pisa/IIT SoftHand (yellow), scoop (red), object (green), and constraint (cyan) models in SynGrasp. Optimization parameters: (a) planar displacement between the hand reference frame $\{\mathbf{H}\}$ and the scoop, (b) length l, width w, and orientation γ_s of the scoop, (c) hand rotation γ_h and constraint rotation γ_c around the z-axis of the object. The axes are represented in red (x), green (y), and blue (z).

algorithm [184].

Scoop-like embedded constraints are particularly useful to slide over surfaces which constraint objects and to design them to achieve this grasp strategy, we decided to include a model of the envisioned environment into the optimization process. In particular, we added a planar surface representing a wall-like environmental constraint to each object model. The surface was positioned in 6 different random orientations with respect to the object, obtaining 300 object-environment pairs per hand.

In the proposed approach, the vector $\mathbf{x}$ of decision variables of the optimization problem contains the planar displacements δ_x and δ_y between the origin of the hand reference frame $\{\mathbf{H}\}$ and the center of the scoop, the length l and width w of the scoop, and the angles γ_s and γ_h: $\mathbf{x} = [\delta_x, \delta_y, l, w, \gamma_s, \gamma_h]$. The angle γ_s expresses the orientation of the scoop with respect to its z-axis (z_s), whereas γ_h is the rotation of the hand frame $\{\mathbf{H}\}$ around the z-axis of the object. Note that we consider only the rotation around the z-axis because we are assuming to implement top-grasps. The decision variables are depicted in Fig. 6.4.

The optimization problem is formulated as follows:

$$\max_{\mathbf{x}} GQI(\mathbf{x}) + \frac{1}{1 + \Delta_\gamma(\mathbf{x})}. \tag{6.1}$$

The two components of the cost function are a grasp quality index (GQI) and a term which considers the scoop, object and constraint orientations.

In the simulations, we chose the Grasp Isotropy Index (GII) as GQI [72]. In principle, however, it is possible to choose any other grasp quality index that can be computed based on the knowledge of the quasi-static model of the grasp [177].

In the second term of the cost function, $\Delta_\gamma(\mathbf{x})$ is defined as: $\Delta_\gamma(\mathbf{x}) = |\gamma_h + \gamma_s - \gamma_c|$, where γ_c is a constant angle expressing the orientation of the environmental constraint with respect to the z-axis of the object reference frame. To ensure that a variety of conditions is considered in the optimization phase, γ_c is randomly set to a value belonging to the interval $[0, 2\pi]$ at the beginning of each optimization.

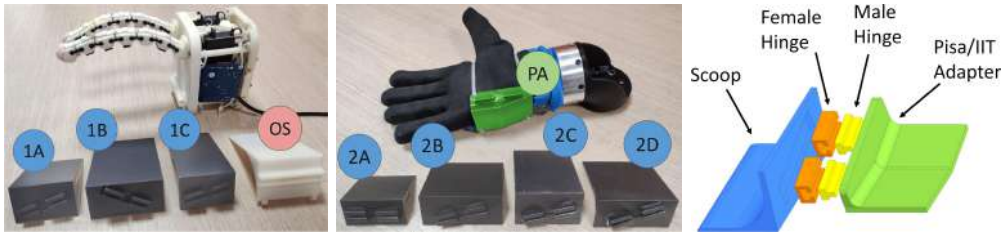
The second term of the cost function is meant to give more importance to the solutions that allow to align the scoop to the constraint, and it approaches 1 when the scoop is parallel to the constraining surface.

To solve the optimization, we adopted a genetic algorithm [178]. We set an initial population of 100 individuals, a non-linear feasible creation function, and a selection tournament function, obtaining a mean number of generations of 6.7 and an average execution time of 170.49 s per object.

In the optimization problem formulation, we accounted for a control parameter (γ_h) together with the design variables of the scoop (δ_x, δ_y, l, w, and γ_s). The first is fundamental to ensure that we can evaluate several different grasps over the object, without constraining a-priori the hand-object relative pose. This allows to really optimize the grasp quality index, suitably aligning the hand to the object, while enforcing the scoop to slide between the object and the environmental constraint thanks to the second term of the cost function. The obtained scoops are thus optimized to perform scoop grasps from the top exploiting wall-like constraints and their design features are fed into the second step of the automated design procedure (see Fig. 6.2).

Note that in this work, we only consider vertical wall-like constraints, but the optimization approach could be extended to differently inclined surfaces, which could also pertain to surrounding objects and not necessarily to fixed environmental constraints.

6.2.3 Gaussian modelling for scoop clustering



(a) (b) (c)

Figure 6.5: Scoops printed after the design phase. (a) The three scoops obtained for the SSG (blue labels from 1A to 1C) and the scoop proposed in [9] (OS red label) whose design variables are $\delta_x = 50$ mm, $\delta_y = 0$ mm, $l = 101$ mm, $w = 70$ mm, and $\gamma_s = 0°$; (b) the four scoops (blue labels from 2A to 2D) printed for the Pisa/IIT SoftHand and the scoop adapter (PA green label); (c) CAD model of the attaching mechanism of the scoops for the Pisa/IIT SoftHand. From the left, the scoop is represented in blue, the female part of the hinge in orange, the male part in yellow and the adapter for the soft hand in green. As it can be noticed, the inclination of the scoop is achieved thanks to the pocket on the back of the scoop.

After running the simulations, we obtained an optimal scoop for each object-constraint pair. As it would be unfeasible to prototype as many different scoops, we decided to reduce the number of scoops using a clustering algorithm.

The training set is indicated with $\hat{\mathbf{x}} = [\delta_x, \delta_y, l, w, \gamma_s]$. On this set, we modelled each candidate scoop using a Gaussian Mixture Model Ω to get a probabilistic encoding of the joint

Table 6.1: Embedded constraints obtained from the Gaussian Mixture Model. We assigned a label to each scoop, from 1A to 1C for the SSG and from 2A to 2D for the Pisa/IIT SoftHand.

Hand	Scoop	Design Variables				
		δ_x [mm]	δ_y [mm]	l [mm]	w [mm]	γ_s [deg]
Soft ScoopGripper	1A	50	5	65	55	−6
	1B	55	−10	110	80	15
	1C	70	−15	110	60	−25
Pisa/IIT SoftHand	2A	35	15	45	60	0
	2B	40	0	60	80	15
	2C	35	−10	75	65	15
	2D	50	−5	65	75	25

distribution $P(\hat{\mathbf{x}}|\Omega)$. The models are characterized by K Gaussian components. We determined the number of Gaussians (or states) K by performing the Bayesian Information Criteria (BIC). The model is represented as

$$P(\hat{\mathbf{x}}|\Omega) = \sum_{k=1}^{K} p_k p(\hat{\mathbf{x}}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{6.2}$$

where p_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ correspond to the priors, the mean, and the covariance of each Gaussian component, respectively.

We initialized the Expectation-Maximization (EM) algorithm using the K-means algorithm. Then, the parameters p_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ are iteratively updated by means of EM algorithm until convergence. We obtained three and four scoops for the SSG and the Pisa/IIT SoftHand, respectively. Table 6.1 reports their features.

6.2.4 Design of the embedded constraints

We developed the scoops using the same procedure followed in [9]. However, we used ABS-M30 to print the scoops and the flexible hinge. We preferred this choice because a less flexible material makes the hinge more robust and hard to be deformed. In the first version of the SSG, the flexible hinge was needed to be more compliant because of the presence of a motor to actuate the scoop. On the other side, the scoops we design should be attached to the hand robustly to be capable to help the grasping.

The printed scoops are shown in Fig. 6.5 with blue labels. In Fig. 6.5a, the red label indicates the original scoop (OS) of the SSG [9], while in Fig. 6.5b the green label indicates the adapter for the Pisa/IIT SoftHand (PA).

The mechanism to attach the scoops to the hands has to be designed according to the optimization results and the adopted hand. We decided to follow this procedure: *i)* we extracted the coordinates of where the adapter should be placed from the outcomes of the GMM; *ii)* we checked the profile of the hand in the neighborhood of these coordinates and we designed an "enveloping" support able to adapt and be adherent to the hand in the given point; *iii)* after having designed the support, we printed the scoops with an inclined pocket on their back made

to attach the female adapter of the hinge reproducing the optimization parameters. The model of these parts can be seen in Fig. 6.5c.

6.2.5 CNN for scoop choice

To make it possible to use the prototyped scoops in grasping tasks, we developed a 3D Convolutional Neural Network capable of selecting the most suitable scoop for grasping a certain object with a certain hand (see Fig. 6.3).

Similarly to what has been proposed in [195], the architecture of our network is composed by a 3D voxel grid ($32 \times 32 \times 32$) as input layer, derived from the voxelization of the 3D point cloud, two convolution layers where the first and second layers have 32 filters (respectively $5 \times 5 \times 5$ and $3 \times 3 \times 3$), and a max pooling layer ($2 \times 2 \times 2$) followed by two dense layers, 128 and N each. In our work, N represents the number of scoops obtained from the GMM clustering. Given that we wanted to predict the grasping success rate of each scoop for novel objects, we modelled the activation of the output layer as a sigmoid function, and consequently our loss function as a binary cross-entropy.

To build the dataset, we chose 10 training objects (Fig. 6.6). The dataset was composed by a voxel grid converted from a partial point cloud, labelled by the grasping success rate of each scoop. The voxel grid included voxels belonging to the constraint itself to discern the same object in different poses with respect to the constraint. Fig. 6.7 shows the voxelization process.

Figure 6.6: Training objects for the CNN. We chose two objects per paradigmatic shape (sphere, cylinder, cuboid, hollow) and a more complex object (the plush toy). The blue cylinder was used in two different poses: lying and standing.

We collected 50 labelled data entries for each scoop (150 for the three scoops of the SSG and 200 for the four scoops of the Pisa/IIT) for the training objects. We also augmented the training dataset by rotating and translating the initial point clouds. The network was trained with the Adam optimizer. All the experimental trials were conducted employing a KUKA LBR iiwa and a Kinect One RGB-D camera. We performed these grasps with the scoop grasp strategy (presented in Chapter 3), sliding on a box that constrains the object. The scoop was always placed parallel to the constraining surface of the box.

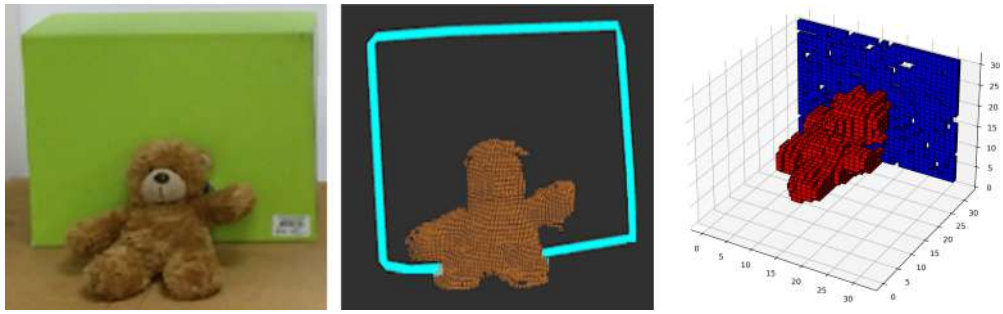


Figure 6.7: RGB image of an object and the environment, corresponding point cloud and voxelized version.

6.3 Experimental results

A video of the experimental trials can be found at link.

The experimental phase consisted in grasping 13 different objects (Fig. 6.8) placed close to an environmental constraint (a large box), exploiting the scoop grasp strategy presented in Chapter 3. Most of the objects we tested belong to the YCB Dataset [185]. We conducted all the experiments employing a KUKA LBR iiwa 7, a Gamma 6-axis force-torque sensor (ATI Industrial Automation, Inc.), and a Kinect One RGB-D camera to detect the objects. In all the trials, the object was kept in the same pose with respect to the constraint since otherwise it would have appeared as a different object-environment pair to the CNN. We executed three batches of experimental trials to demonstrate different hypotheses.

First, we tested each scoop of the SSG on three objects with prototypical shapes (spherical, cylindrical, and cuboidal) to prove that the scoop having the highest success probability according to the CNN actually performs better than the others on previously unseen objects. The second experiment aimed at testing the optimal scoops of the SSG suggested by the CNN and at comparing their effectiveness with respect to the scoop originally designed in [9]. In the third experiment, we tested the scoops designed for the Pisa/IIT SoftHand. We compared obtained results by repeating the same experimental trials removing the designed add-ons. The aim was to study whether embedded constraints can represent an aid in case of restrained scenarios, where the hand alone cannot proficiently use the environment.

In all the experimental trials involving the use of a scoop, we carried out the following steps. The object to be grasped was placed close to a vertical surface. The vision algorithm recognized the planes in the scene (captured through an RGB-D camera) and identified a cluster of 3D points belonging to the object and the constraint. The 3D point cloud was then converted into a 3D voxel grid which was fed to the CNN. Lastly, the outcome of the network provided us with the optimal scoop to be mounted on the hand. Hence, the robot was moved towards the vertical surface placing the scoop plate parallel to it. A hybrid force-velocity controller was used to ensure that the robot arm stopped its motion as soon as the scoop touched the surface. Once this condition was met, the scoop started sliding on the surface towards the object at a constant speed parallel to the surface while maintaining a constant force in the direction normal to the scoop plate. Once the hand reached the object, the fingers started their closure motion. A grasp

Figure 6.8: Tested objects. Yellow labels indicate the objects used in Experiment 1 (Table 6.2). Blue labels indicate the objects used in Experiments 2 and 3 (Table 6.3).

Table 6.2: Objects size, output of the CNN and grasp success rates obtained from Experiment 1. We reported in bold the CNN highest predictions and the highest success rate for each object.

				Output CNN			**Success Rate**		
ID	Object	Size [mm]	Mass [g]	Scoop 1A	Scoop 1B	Scoop 1C	Scoop 1A	Scoop 1B	Scoop 1C
1	tennis ball	⌀65	59.4	**0.56**	0.14	0.16	**6/10**	2/10	2/10
2	body spray	⌀45 × 175	34	0.34	**0.67**	0.17	4/10	**6/10**	0/10
3	white box	90 × 175 × 60	146	0.29	0.32	**0.47**	2/10	4/10	**6/10**

was considered successful if the object was picked up and moved to the final position without falling. Otherwise, it was deemed to be unsuccessful.

6.3.1 Experiment 1: validation of the algorithm with the SSG

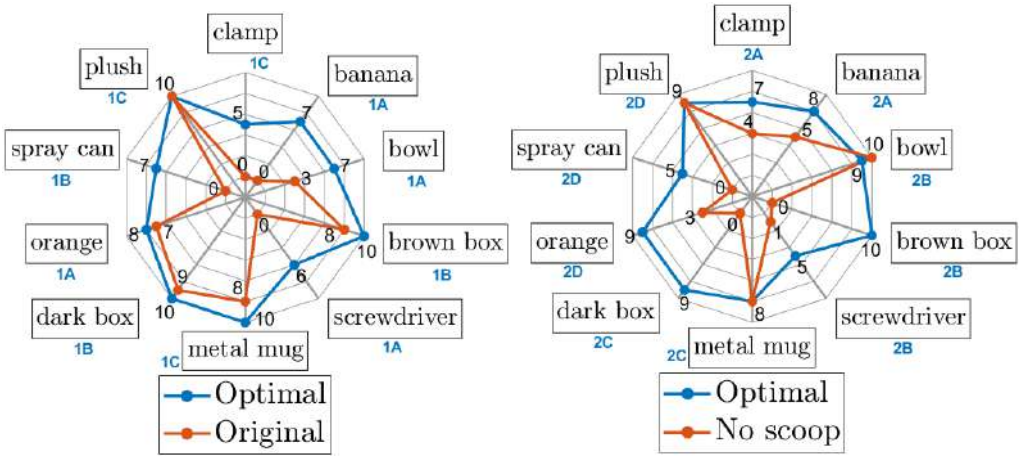
Experiment 1 aimed at validating the output of the network. Note that the reported results concern the CNN trained for the SSG. We decided to focus on just one hand since the neural networks have been trained with the same method for both hands, but with different data. Three objects (see the yellow labels in Fig. 6.8) whose characteristics are reported in the first three columns of Table 6.2 were picked up with the SSG, 10 times per scoop, collecting a total amount of 90 trials. The network outputs and the experimental results are reported in the last two columns of Table 6.2.

6.3.2 Experiment 2: optimal vs original scoop for the SSG

The goal of the second batch of experimental trials was to test if the optimal scoops provided by our architecture could outperform the standard one originally mounted on the SSG. Ten novel objects (shown with the blue labels in Fig. 6.8 and described in Table 6.3) were grasped in two

Table 6.3: Objects tested in Experiment 2 and Experiment 3.

ID	Object	Size [mm]	Mass [g]
1	clamp	$85 \times 110 \times 30$	59
2	banana	$\varnothing 35 \times 160$	66
3	bowl	$\varnothing 155 \times 55$	147
4	brown box	$100 \times 145 \times 65$	76
5	screwdriver	$\varnothing 35 \times 210$	98.4
6	metal mug	$\varnothing 90 \times 80$	118
7	dark box	$115 \times 155 \times 4.5$	28
8	orange	$\varnothing 70$	50
9	spray can	$\varnothing 50 \times 160$	118
10	plush	$100 \times 140 \times 130$	102



(a) (b)

Figure 6.9: Success rates for (a) Experiment 2 and (b) Experiment 3. The blue line represents the success rate obtained with the optimal scoops in both cases, while the orange one shows the success rate obtained using (a) the SSG original scoop, and (b) the Pisa/IIT SoftHand without any scoop. Next to each object name, we reported the name of the used optimal scoop in blue (*cf.* Fig. 6.5).

different conditions by using the original scoop and the one provided by the CNN. A total of 200 experimental trials was collected by performing 10 trials per object in each condition. Fig. 6.9a reports the comparison between the success rates obtained in the two cases.

6.3.3 Experiment 3: Pisa/IIT SoftHand with and without scoop

Experiment 3 was meant to demonstrate that a generic commercial hand equipped with optimal embedded constraints outperforms the same hand without any additional part. We adopted the

Pisa/IIT SoftHand to carry out the grasping procedure by testing 10 objects (the same used in Experiment 2) in two conditions: with and without the scoop. We executed 10 trials per object per condition, collecting a total of 200 trials. To retrieve the optimal hand pose for grasping a certain object without the scoop, we decided to solve an optimization problem similar to the one presented in Chapter 3. The used optimal vector $\mathbf{x}^* = [x^*, y^*, z^*, \gamma^*]$, consisted of the relative position between the hand and the object, and the optimal angle γ^* which represents the hand rotation around the z-axis of the object reference frame. The solutions resulting in compenetration between the hand and the environmental constraint were discarded. Obtained success rates are reported in Fig. 6.9b.

6.4 Discussion

Experiment 1 is fundamental to evaluate the validity of the proposed method.

The results shown in the Table 6.2 provide us with two main insights: *i)* the scoop suggested by the network coincides with the one providing the highest success rate in the experiments, and *ii)* overall, the values of the CNN predicted success rates are comparable with the experimentally obtained ones. In Experiment 2, the overall success rate of the trials carried out with the optimal scoop is 80% against 45% of those performed considering the original scoop. This means that the newly designed scoops outperform the original ones in restricted scenarios. Experiment 2 also gives us interesting insights regarding the application of the scoops designed for the SSG. As pointed out in Sec. 3.4.2, small objects were challenging to be picked up performing the scoop grasp sliding on a vertical surface. However, designing a proper scoop for a given object-constraint pair compensates for this problem. Remarkable examples are the clamp (Fig. 6.10b), the banana, the screwdriver, and the spray can which, although having a relatively low overall success rate of 62% (25 out of 40 trials) with the optimal scoop, were never picked up using the original scoop (Fig. 6.10a). The remaining objects, instead, present similar or better success rates (55 out of 60 trials, 92%) when grasped with respect to the original scoop (45 out of 60 trials, 75%).

Experiment 3 aimed at demonstrating that the presented methodology can be applied to hands designed without foreseeing any additional parts to enhance their performance. Overall, the experimental trials performed with the hand equipped with the optimal scoop present higher success rates (79%) than without it (40%). In particular, each object grasped with the customised hand presents equal or higher success rates than the default hand. The only exception is the bowl (9 out of 10 trials). In this case, the main reason is that the object is wide enough that the hand can reach an optimal pre-grasp pose without interfering with the environmental constraint. However, this kind of behavior strongly depends on the characteristics of the object. Indeed, hollow (e.g., bowl) or deformable (e.g., plush toy) objects are more prone to be easily grasped even without the add-on.

On the contrary, large and rigid objects (such as the box in Fig. 6.10d) turned out to be impossible to grasp without the scoop (Fig. 6.10c). The presented experimental results show that the exploitation of optimal embedded constraints is generally the best choice.

(a)

(b)

(c)

(d)

Figure 6.10: Examples of grasps from Experiment 2 and 3. On the top, (a) the SSG fails to grasp the clamp because of the distance between the fingers and the scoop, while (b) the SSG equipped with the suggested scoop is achieving the grasp. On the bottom, (c) the Pisa/IIT SoftHand is not equipped with the scoop and it fails to grasp wide undeformable objects (e.g., brown box), while (d) the hand achieves more successful grasps with the help of the scoop.

6.5 Conclusions

In this chapter, we proposed a method to design optimal scoop-like embedded add-ons to fully exploit the presence of constraints and obstacles in unstructured environments. We applied our methodology to two different soft robotic hands, the Soft ScoopGripper and the Pisa/IIT SoftHand. Running 300 grasp simulations per hand, we collected optimal design variables by solving an optimization problem. To obtain a reasonable number of add-ons, we clustered the outcomes of the optimization by means of Gaussian modeling, obtaining a total amount of seven scoops. Hence, we used a data-driven approach to correctly select the optimal scoop with respect to a given object in a restricted scenario. We tested the designed scoops in three different experiments to validate our method. Experimental results show that the developed architecture always provides the optimal scoop. Additionally, properly designed add-ons allow augmenting the range of graspable objects and the exploitation of new grasping strategies even with hands that are not initially designed to embed a scoop. Still, the scoop can be easily detached when considered a hindrance for the execution of other grasping strategies. In future work, we plan to exploit this technology in highly cluttered environments where the scoop can also be exploited for non-prehensile actions.

Future research will also focus on improving the presented scoops, considering different concavities, which can easily comply with the object surface. In addition, we will devise methods to optimize the overall structure of the scoop taking into account its structural mechanics properties and we will explore the design of soft scoops in which the compliance is a functional element for task accomplishment.

Chapter 7

Data-driven topology optimization framework for robotic grippers

"*You can have data without information,*
but you cannot have information without data.."
Daniel Keys Moran

A widespread methodology to enhance the design of robotic devices is represented by topology optimization. Typically, the optimization aims at designing a certain part of the robot to satisfy a priori, user-defined mechanical properties while minimizing the used material for building the structure. In this chapter, we apply topology optimization to robotic grippers, and we propose to define the requirements for the optimization in a data-driven way based on simulated experiments of grasping tasks. Specifically, the architecture we propose is composed of three sequential phases. The input of the architecture includes the initial model of the gripper, the specific gripper component to be optimized, and a set of parameters. The first part of the architecture acquires force signals from the gripper component that are sensed during the grasping simulations. Hence, these signals are fed into the second phase, which analyzes the forces through pixel connectivity and Dynamic Time Warping algorithms and provides the instructions for the topology optimization. Ultimately, the third block performs the optimization. The method is tested by optimizing a specific part of a soft-rigid gripper. Results from simulation confirm that the proposed architecture provides an improved version of the original gripper, not only in terms of optimized use of materials but also in terms of grasp success rate.

The rest of the chapter is organized as follows. Sec. 7.1 defines and motivates the study. In Sec. 7.2, we describe the different blocks composing the architecture of our toolbox. In Sec. 7.3 we report the analysis of the tests, while Sec. 7.4 contains a discussion of the results. Finally, Sec. 7.5 derives the conclusion of the study, outlining the future works.

7.1 Introduction

Over the last decade, optimal design methods have been exploited in robotic manipulation to build grippers capable of grasping different objects or performing specific in-hand manipulation motions [208, 209, 210]. One of the adopted techniques is Topology Optimization, which aims to maximize a system's performance while reducing the material layout [159]. Thanks to their versatility, topology optimization algorithms allow developing grippers capable of complying with specific requirements [160] or actuation technologies [161, 162, 163]. The contacts between the gripper and the objects are typically modeled in the form of pressure loading and friction tractions [165] or as constraints [167]. The positioning of contacts is defined a priori based on the designer's intuition of where the gripper will most probably grasp the objects (e.g., fingertips [211]. While this is a reasonable choice for relatively simple grippers (e.g., 2-fingered devices [163]), it might not be realistic with more complex, possibly soft, robots.

In this work, we propose to topologically optimize the design of soft-rigid robotic grippers based on contact force data retrieved from preliminary dynamic grasp simulations. The newly proposed framework consists of the pipeline sketched in Fig. 7.1. Our approach bases the gripper design on data-driven considerations, similarly to what is done in *co-design* methods in which the structure and the control strategy of the gripper are simultaneously optimized verifying the feasibility of the solutions in simulation [171]. As shown also in other previous works, simulation data can provide useful information for the design of robotic grippers [172].

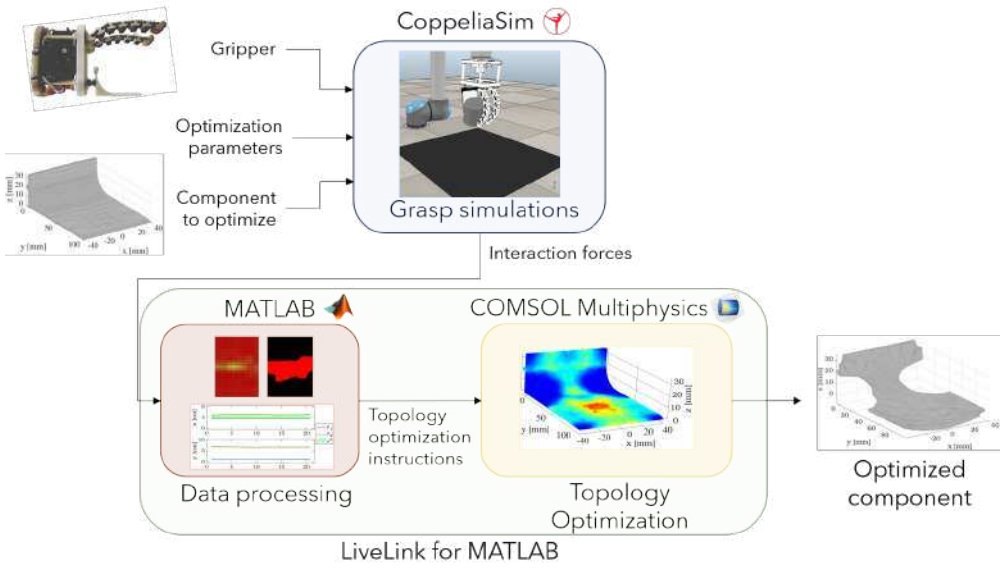


Figure 7.1: Schematic of the proposed framework. In the first block several grasps are simulated and force signals from sensor plates located in the part(s) of the gripper to be optimized are recorded. The data are then converted by the second block into heightmaps, processed and then clustered. Lastly, the obtained information is translated into loadings, tractions, and constraints for the topology optimization analysis.

In [172], the authors proposed to exploit simulation data with a previously developed gripper

finger design and optimization methods for generating a finger cut-out for an asymmetrical object used in industrial assembly tasks. Moreover, as shown in Chapter 6, we used simulations for the initial steps of a data-driven automated design procedure, as they can be used to generate several data, without performing real experiments, which would be time-consuming and possibly unsafe.

The framework proposed in this chapter consists of different steps. We assume to have an initial gripper design that we want to optimize topologically. In the first phase, the simulated model of the initial gripper is loaded in CoppeliaSim [212], and several grasps are simulated with it. During the simulations, force signals from sensor plates located in the part(s) of the gripper to be optimized are recorded. The retrieved data are then converted into heightmaps and processed using the pixel-connectivity algorithm and Dynamic Time Warping clustering [193]. Lastly, the obtained information is translated into loadings, tractions, and constraints for the topology optimization analysis conducted in COMSOL Multiphysics [213]. The proposed methodology was tested with the Soft ScoopGripper, a tendon-driven soft gripper embedding a rigid flat part [9].

7.2 Materials and Methods

The framework proposed in this work allows to topologically optimize grippers based on accurate a priori information on their grasping capabilities. As shown in Fig. 7.1, the framework is composed of three main phases. The first one is devoted to the dynamic simulation of several grasps with the selected gripper and accepts inputs from the user, including the model of the gripper, the component of the gripper to optimize, and a set of parameters. The data gathered from the 600 simulated grasps are stored and those related to the forces applied onto the gripper are used as inputs for the second phase. Here they are processed to obtain a set of instructions for the topology optimization that is carried out in the last phase. The final output consists of the STereoLithography (STL) file of the optimized component.

In our implementation of the framework, the gripper simulations are conducted in CoppeliaSim, the force data are analyzed in MATLAB (MathWorks, Inc., USA), and the Topology Optimization is carried out in COMSOL Multiphysics (Comsol Inc., USA) through the LiveLink for MATLAB. However, in principle, other software with similar features could be exploited as well.

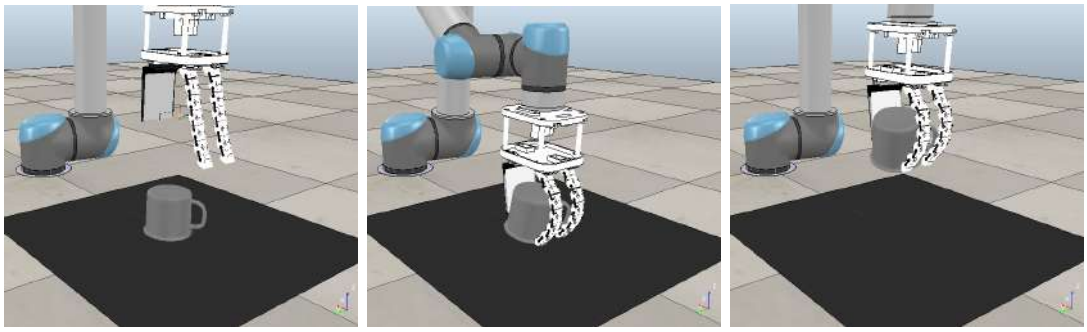
In the following, we detail the steps of the proposed framework applied to a specific gripper, i.e., the Soft ScoopGripper [9], and a specific part of it, i.e., its embedded scoop-like flat structure. Nonetheless, the proposed procedure is general enough to be applicable to different hands and different parts of them (e.g., phalanges, nails, palm).

7.2.1 Data Collection from Grasp Simulations

To conduct dynamic simulations in CoppeliaSim, the first step to perform is to import the URDF file of the gripper. We built the URDF of the Soft ScoopGripper considering that it is an underactuated soft-rigid hand that embeds a flat, scoop-like surface that can be used to proficiently exploit environmental constraints during grasping tasks, as discussed in Chapter 3. The scoop is actuated by two tendons, and it is attached to the gripper palm through a flexible

hinge to better adapt to the features of the environment. In addition, the gripper has two fingers composed of rigid phalanges and soft joints and driven through tendons by a differential mechanism. To have a more efficient simulation, in CoppeliaSim, we modeled the gripper fingers as lumped parameters kinematic chains, where each flexible joint is described by a rotational active joint. We empirically determined in [9] the closing motion, controlling in velocity each joint and making the fingers independent, similarly to the effect generated by the differential mechanism present on the real gripper. In this way, we obtained a realistic "soft" behavior of the fingers, without explicitly modeling the dynamics of the tendons.

The part of the gripper that we chose to optimize was the scoop, as it plays a fundamental role in the grasps performed by the gripper, and can even be adapted to different hands, as demonstrated in Chapter 6. In the simulator, the surface of the selected part is divided into a grid whose size is a parameter that can be defined by the user. In each element of the grid we placed a simulated force/torque sensor to characterize the hand-object interactions arising during grasping tasks. In our setup, we recorded only the forces normal to the sensors surface and the grid was made of 7×9 elements to cover the entire surface of the component and considering that there should be a minimal distance between the sensors.



(a) (b) (c)

Figure 7.2: Example of a grasping sequence in simulation on CoppeliaSim. (a) the robotic arm approaches the object, (b) the record of the forces is started after a contact is detected, (c) the gripper brings the object to a final position.

To conduct the simulations, the gripper has to be attached to the model of a robot arm. We chose an UR5 collaborative robot (Universal Robots) and we performed a total of 600 top-grasps, i.e., 12 trials for each of the 50 different objects that we selected from the YCB dataset [185].[1] The objects were placed within the robot workspace to make the planned grasps feasible. The initial object orientation was set to a value equal to $\kappa\pi/6$ radians, with κ=1, .., 12. To understand if a grasp was successful, we placed a force sensor on the surface where the objects lay: if no force was read after 3 seconds from the end of the grasp execution, a grasp was considered successful. We adopted the Newton physical engine to perform the simulations. Fig. 7.2 shows an example of a simulated grasp.

During each simulation, we acquired the force signals from the sensors placed on the surface

[1]Note that the number of trials and objects, as well as the type of objects, can be adapted to the required accuracy and to the envisaged application.

Table 7.1: Parameters to be provided to the framework and the values adopted for our case-study.

Parameters	Values
Gripper model (URDF)	Soft ScoopGripper
Component model (STL)	scoop
Grid size	7×9
# of trials	600
# of objects	50
# of object orientations	12
Sampling frequency	10 Hz

of the scoop. The sensors started recording when the first non-zero force was detected, whereas the opening command of the gripper after the grasp triggered the stop. The sampling rate of the signals we chose was 10 Hz.

The parameters we chose in this first phase of the framework are summarized in Table 7.1.

7.2.2 Data processing

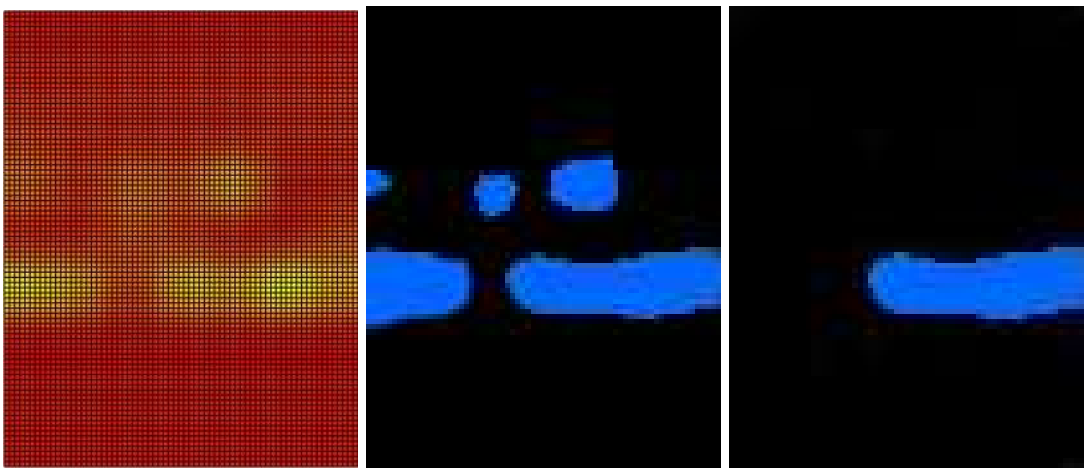
The force signals sensed on the gripper component were analyzed in MATLAB to translate them into helpful information to perform a topology optimization. First, we separated the successful grasps from the failed ones, creating two different datasets. Then, we applied the following four steps to them.

Step 1. For each time instant of each grasping trial, we created a 2D image in which each color indicates a different magnitude of the force. To get a higher resolution, we up-sampled the data using Gridded Interpolation. The up-sampling is useful to increase the resolution because sensors have limited dimensions even if they are simulated (neglectable mass and inertia). An example of final 2D image grid is shown in Fig. 7.3a.

Step 2. Starting from the obtained grids we created a binary image using a threshold value for the force magnitude and coloring only the pixels corresponding to force values above the threshold. We determined the threshold to distinguish an object contact from a finger one. Thus, we considered the maximum force applied by the fingers directly touching the sensors as threshold. To make the information more homogeneous we used a pixel-connectivity algorithm to create uniform blobs of pixels (see Fig. 7.3b), similarly to what was done in [214] for a different application. Each blob is described through a limited set of parameters, including its center, its area, etc. [214].

Step 3. For each image we kept the largest blob and deleted the other ones, thus selecting only the area in which most of the hand-object interactions arise, as shown in Fig. 7.3c.

Step 4. After having obtained a blob for each time instant of each grasping trial, we used the Dynamic Time Warping (DTW) algorithm [193] to measure the similarity between the different grasping trials in terms of evolution of the parameters of the blobs. This information was then used to cluster grasping trials in which the blobs evolved in similar ways. We adopted the K-medoid as clustering algorithm, and not the more widely adopted K-means, as its results are reliable since they effectively belong to the original signals.



(a) (b) (c)

Figure 7.3: Analysis of a single time instant of a single grasping trial: (a) 2D grid of the recorded forces, (b) image of the blobs obtained from the pixel-connectivity algorithm, and (c) image of the blob with the largest area. As it can be noticed, the blobs in (b) correspond to the yellow areas in (a), i.e., areas of the studied surface where the forces were higher.

To determine the optimal number of clusters k, we employed the Calinski-Harabasz Criterion or Variance Ratio Criterion (VRC). The Calinski-Harabasz index is defined as:

$$VRC_k = \frac{\sigma_B^2}{\sigma_W^2} \frac{N-k}{k-1} \tag{7.1}$$

where σ_B^2 is the overall between-cluster variance, σ_W^2 is the overall within-cluster variance, k is the number of clusters[2], and N is the number of observations. Through this analysis, for each cluster, we were able to extract the centroid parameters describing a resulting blob evolving in time, as shown in Fig. 7.4. To translate this evolution into instructions for the topology optimization, we considered the bounding box of the blob and its center at each time instant.

7.2.3 Topology Optimization

Once we obtained the results from the data processing phase, we translated the force signal distributions into the information necessary to perform a classic topology optimization of the examined piece. We conducted the topology study with the density method on COMSOL Multiphysics because it can be easily integrated with MATLAB thanks to LiveLink for MATLAB.

First, we inserted the scoop model establishing as material the ABS with Young's modulus equal to 3.2E+9 Pa and Poisson Ratio of 0.35. Then, we specified the loads necessary to actuate it through the cable-driven system and the fixed boundary of the hinge joint linking the scoop to the rest of the gripper. Therefore, we applied the results evaluated in the data processing. From the data of the successful grasps, we obtained two force distributions that, at steady state,

[2]In our case, $k = 2$ for both datasets containing the successful and the failed grasps.

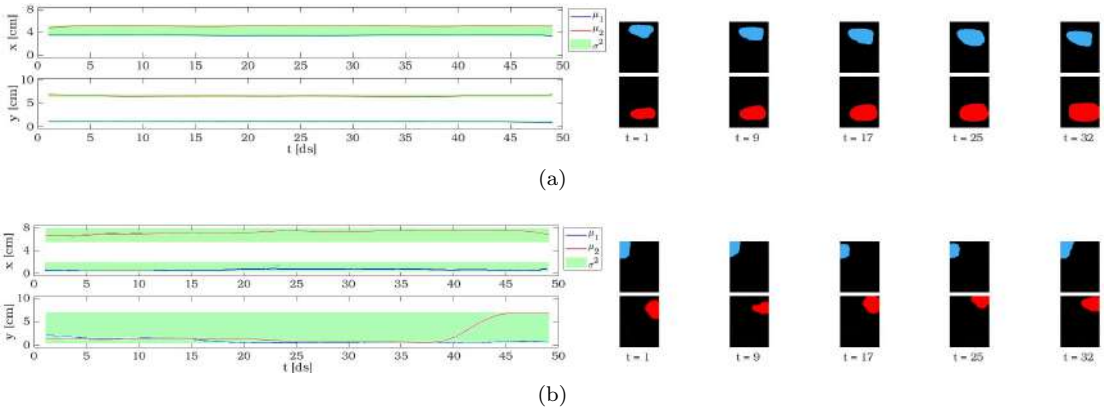


Figure 7.4: Final distributions of the successful (a) and failed (b) grasps. From the left: the position assumed by the clusters on the x-axis of the grid, the position assumed by the clusters on the y-axis of the grid, five time instants of the positioning of cluster 1 (in blue), and five time instants of the positioning of cluster 2 (in red).

are located in the center of the scoop along the x-axis (x=4.27 $\pm$ 0.95 cm) and mainly in two regions on the y-axis (y_1=6.75 $\pm$ 0.27 cm and y_2 = 1.14 $\pm$ 0.19 cm, respectively). Thus, we set two boundary loads in these areas.

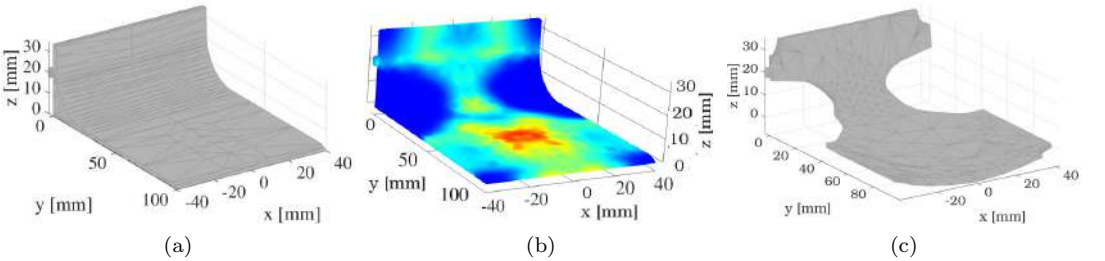


Figure 7.5: Steps of the design of the scoop. (a) initial design, (b) Von Mises stress distribution during the topology study, and (c) final optimized design .

On the other side, we extracted two other force distributions from the analysis of the failed grasps close to the boundaries of the scoop along the x-axis (with y=1.23 $\pm$ 0.42 cm). We interpreted this information as the objects tended to slip from the center of the scoop towards its boundaries. For this reason, we set the areas of these distributions as fixed parts, so they do not undergo any deformations during the grasping execution, keeping the object in the middle of the mechanical piece.

To solve our topology optimization problem, we adopted the widely used modified SIMP method [159], where the design domain is discretized into N finite elements, each characterized by its density $\rho \in [0, 1]$. Given that the resulting design can depend on the used mesh, we adopted a minimum length scale via a filter radius in a Helmholtz filter. We used the moving asymptotes algorithm (MMA) [215] to solve the optimization problem. We assumed a volume

fraction V^* of 0.5, a maximum number of iterations of 150, and a relative tolerance stopping criterion of 0.001. Finally, we performed a shape optimization to make the structure more robust, plotted the optimal structure and exported it as a STL format file for testing. The original and the optimized scoops are shown in Fig. 7.5.

7.3 Experimental results

After obtaining the STL of the optimized scoop, we carried out an analysis to prove its efficacy and usefulness.

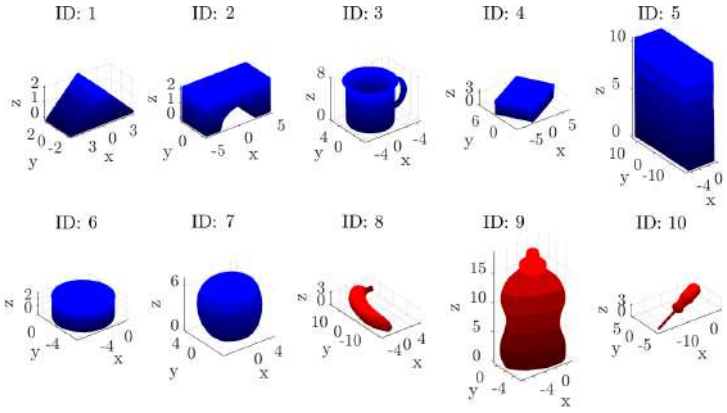


Figure 7.6: Tested objects. We colored in blue the 7 tested objects used also during the analysis, whereas we colored in red the 3 novel objects. The unit of measurement is in cm.

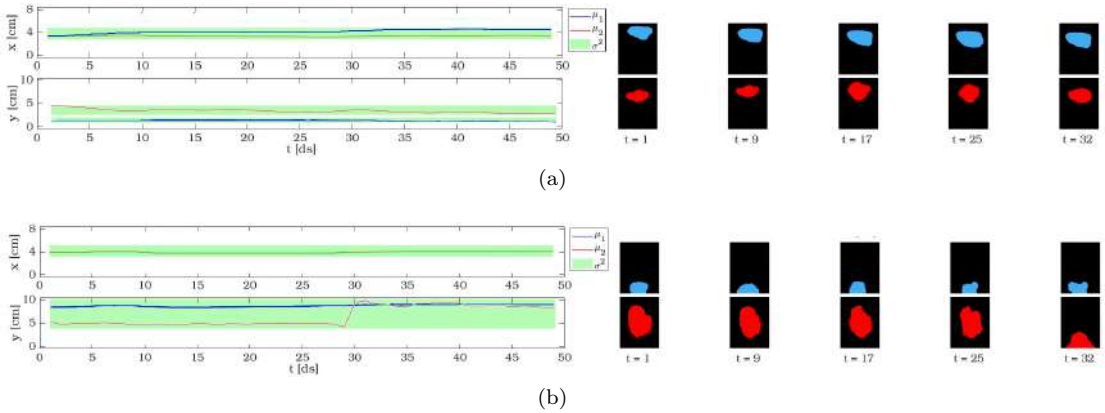
The test was meant to verify the newly designed scoop's success rate and force distributions compared with the original one. Thus, we executed 300 grasps per scoop in simulation, as previously done in Sec. 7.2.1. We included in the test dataset 7 objects belonging to the ones on which the analysis has been performed and 3 new ones to prove that our method is generalizable. The dataset is shown in Fig. 7.6.

We evaluated the experiments in terms of grasp success rate and of the intensity of the forces applied on the scoop.

Regarding the first aspect, we obtained a total number of successful grasps of 189/300 and 232/300 for the original and the optimal scoop, respectively. The grasp success rates for each object are reported in Table 7.2. With respect to the second aspect, we evaluated the force signals recorded on the surface of the original and optimal scoops with the analysis proposed in Sec. 7.2.2. Here we compared the values assumed by the clusters over time resulting from the DTW analysis and the difference in their standard deviations. The signals obtained from the optimal scoop are shown in Fig. 7.7.

7.4 Discussion

Evaluating the testing phase results, we can affirm that the proposed framework can improve the gripper performance. Indeed, overall, the grasping success rate increases when using the optim-



(a)

(b)

Figure 7.7: Test analysis of the successful (a) and failed (b) grasps with the optimal scoop. From the left: the position assumed by the clusters on the x-axis of the grid, the position assumed by the clusters on the y-axis of the grid, five time instants of the positioning of cluster 1 (in blue), and five time instants of the positioning of cluster 2 (in red).

Table 7.2: Success rates of the testing phase for both the original and optimal designed scoops.

Objects	Obj. ID	Original Scoop	Optimal Scoop
Triangular Toy	1	24/30	25/30
Bridge Toy	2	19/30	18/30
Metal Mug	3	14/30	27/30
Sugar Box	4	23/30	23/30
Cracker Box	5	23/30	25/30
Tuna Can	6	18/30	20/30
Apple	7	26/30	30/30
Banana	8	10/30	20/30
Mustard	9	20/30	25/30
Screwdriver	10	12/30	21/30
	Total	189/300	232/300

ized component (77.33% against 63.67%). Specifically, the newly designed component is more effective with objects that were difficult to be picked up with the original scoop. Remarkable examples are the metal mug, the banana, and the screwdriver (a total of 68/90 against 36/90 successful grasps). The explanation lies in the fact that being the original scoop flat, the objects tended to roll outside boundaries along its x-axis. Constraining the scoop sections during the design phase (see Sec. 7.2.3), we achieved a concavity in the center of the scoop that opposes the rolling. The analysis of the sensed forces fully confirms the previous statement. Indeed, as it can be noticed from the forces recorded during the successful grasps (see Fig. 7.7a), the optimal scoop helps the gripper to restrain the objects within the same region, preventing involuntary movements of the objects.

Not only are the forces on the x-axis more concentrated in the scoop center (x=3.9185 $\pm$

0.0715 cm), but also on the y-axis we noticed a desired trend, considering that the signals are closer to the origin of the component (y_1=3.74±0.7278 and y_2=1.02±0.2159 cm). This behavior means that the objects are moved towards the palm of the gripper during the grasping execution, making the grasp more robust.

Nonetheless, when using the optimal component, the gripper lost some capabilities in grasping box-shaped objects, e.g., the bridge toy and the cracker box were equally or less caught. Moreover, evaluating the force analysis, we can affirm that the failing trials are mainly related to a slipping motion along the y-axis (Fig. 7.7b). This slippery behavior is probably due to unpredictable interactions between the fingers and the objects. Thus, the objects are not properly grasped because the fingers fail to keep them on the scoop during the grasping motion. To corroborate this assumption, it can also be noticed from Fig. 7.7b that the red blob present oscillating trends on the y direction, where it assumes a value of almost 10 cm at time instant $t = 30$ starting from around 5 cm.

7.5 Conclusions

In this chapter, we propose a framework to enhance the topology optimization of grippers, considering a data analysis on previously conducted simulations of grasps. We conceived our framework as three distinct sequential blocks. Given different parameters from the user, the first block is responsible for recording the force signal acting on a gripper component during grasping execution performed in simulations on CoppeliaSim. The architecture's second part consists of a MATLAB algorithm that analyzes the forces through pixel connectivity and Dynamic Time Warping algorithms. Lastly, the framework translates these distributions into information necessary to perform the topology optimization on COMSOL, resulting in an optimized STL of the given component. We tested our method by optimizing a link of the Soft ScoopGripper, a soft-rigid gripper. From the results, we noticed that the proposed framework produces an analysis that effectively improves the given component and prevents unwanted behaviors of the gripper.

In future works, we plan to improve the framework with other mechanical analyses further and consider composite optimizations of different components. As the framework is now, it can be considered an open loop, given that no feedback is provided during the optimization. Thus, another possible development consists of developing the closed-loop version of the architecture, where the components are optimized and updated after a certain number of iterations.

Another possible development of the proposed framework will consist in the evaluation and design of compliant yielding structures, possibly considering an analysis for hyperelastic materials.

Future research will also focus on the transfer of this method from simulated to real robotic grippers, evaluating the possibility of acquiring the training data through experiments using ad-hoc sensors.

Part III

Application to Real World Scenario

Chapter 8

Design and Control of a Soft-Rigid End-Effector for Food Handling

"*Aimless extension of knowledge is merely inefficiency. I am designed to avoid inefficiency.* "
Isaac Asimov

Food handling is a very challenging task for robotic grippers. It deals with the manipulation of highly deformable and fragile items that can be easily damaged and consequently no longer be attractive. Moreover, ingredients for the preparation of the different dishes are usually stored in small containers that are often not easily accessible from the top in real scenarios.

This chapter introduces an innovative soft-rigid, tendon-driven gripper: the Double-Scoop Gripper (DSG). Its two-fingered design exploits a specialized structure to cope with constrained spaces (e.g., containers in narrow shelves). The DSG can delicately grasp objects of various shapes by employing two scoop-shaped fingertips that can form a single plate when fingers are flexed. Data obtained from an on-board camera are used to detect the food item features and plan the grasping strategy that better exploits the possible environmental constraints regulating the opening of the two fingers and the approaching direction of the gripper.

Experiments were conducted using real food ingredients within a pick-and-place setup to evaluate both the grasping and the realising capability of the gripper. Obtained results are promising and suggest that this approach could be particularly advantageous in the context of automated food serving.

The chapter is organized as follows. Sec. 8.1 defines the problem of grasping in clutter, deriving its motivation. In Sec. 8.2, we describe the proposed methodology. In Sec. 8.3 Sec. 8.4, we report and discuss the experimental results. Lastly, Sec. 8.5 derives the conclusion of the study, outlining its possible further developments.

8.1 Introduction

Food manufacturing has witnessed significant advancements in automation and robotics, revolutionizing food handling processes. In this context, industrial food service robots play a pivotal role in enhancing efficiency, conserving space, and elevating cleanliness and safety standards [216]. Over the past few decades, robotics found widespread usage in tasks like dispensing ingredients, executing precise cuts, packaging or casing food items, as well as skillfully picking and placing products into containers while also facilitating the sorting process [15, 217]. Food handling is an important task also in assistive applications in which robot manipulators can be used to help people in the kitchen or feed and serve patients with upper limb impairments [218, 219].

When dealing with pick-and-place tasks, robotics must face three main challenges [220, 221]. To begin with, the robot's end-effector must delicately handle the fragile nature of food items susceptible to damage. Secondly, the robot's capability to adjust to environmental changes is crucial, ideally without necessitating a reconfiguration of the entire robotic setup. Lastly, vision algorithms are vital in detecting and recognizing the targeted food items for picking. However, complexities like obstacles within the scene often obstruct a clear view of the workspace.

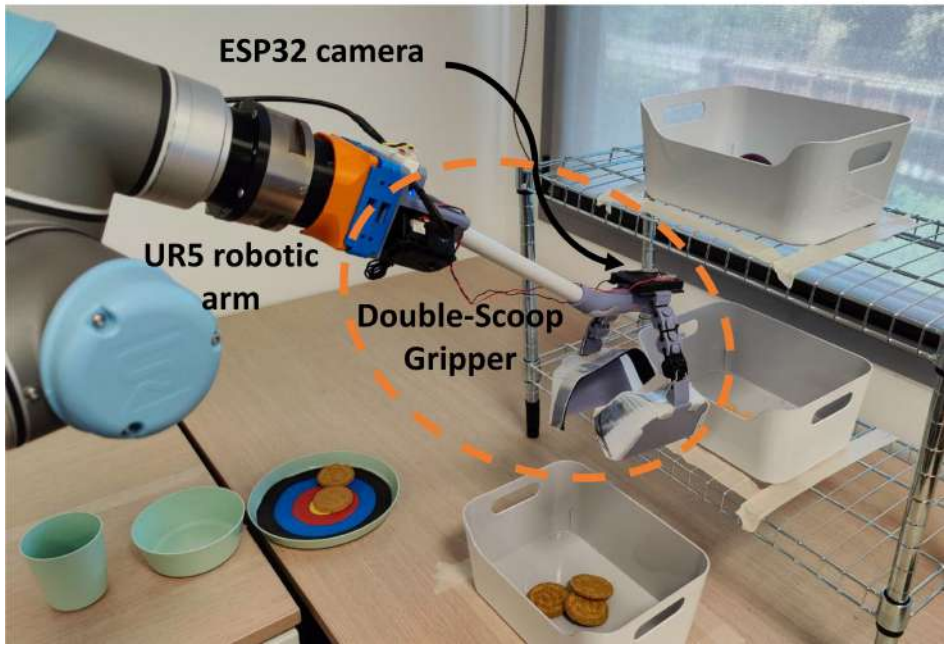


Figure 8.1: Experimental setup for food handling. Pick area: 3 containers at different heights (table, within and above the shelves). Place area: 3 containers of varying sizes (plate, bowl, glass). The proposed Double-Scoop Gripper, embedding an on-board camera, is indicated with an orange circle.

In this chapter, we propose the development and application of a novel soft-rigid gripper, the Double-Scoop Gripper, specifically designed for pick-and-place scenarios (as shown in Fig. 8.1). The gripper was devised to tackle the food tray assembly scenario of the *2023 IEEE Robosoft*

Competition ("food handling for trays preparation)[1] ", where food was stored in containers placed in restricted spaces (i.e., containers in narrow shelves) and needed to be released into small containers like bowls and plates. The gripper structure comprises two opposing tendon-driven soft fingers. Exploiting the idea of "side and bottom grasp" defined in [220] and introduced in [5], we positioned at the tips of these fingers a flat, rigid component able to slide below the food items with additional flexible borders that create a scoop (Fig. 4.9). This unique configuration offers a threefold advantage: firstly, it allows the gripper to cage objects rather than applying conventional pinching; secondly, it improves the releasing of the food items with respect to other soft grippers; lastly, it enables the gripper to grasp the food exploiting the constraints present in the environment (e.g., other food items or container walls). To cope with narrow environments, the gripper's palm is attached to the robot arm end-effector through a rigid support element. This feature allows us to manoeuvre the gripper into constrained spaces where it is not possible to perform top grasps. The two actuators of the fingers and the electronics for low-level control are positioned at the robotic arm end-effector making completely washable the gripper part that may be in contact with the food. The repositioning of the motors, also decreases the bulkiness of the gripper which represents an important feature when grasping in narrow spaces.

An RGB camera is embedded in the palm of the gripper. This placement ensures the identification of objects within containers, even when obstructive elements like shelves or other containers occlude the visual field of an external camera. To determine the objects' features, we employed a vision algorithm that first acquired the camera images and consequently planned the grasping strategy that better exploits the possible environmental constraints.Thus, depending on the position and the size of the food items, the opening of the fingers and the gripper orientation were regulated to adapt to the exploitable environmental constraint.

We tested our device evaluating the grasping and releasing success rate in a scenario with real ingredients in multiple containers and a tray, considering single and multiple objects pick-and-place.

8.2 Materials and Methods

8.2.1 Gripper Design

The design of our gripper complies with the two following guidelines: *i*) prevent damage to delicate food and *ii*) grasp items placed in narrow places. We developed this gripper accounting for the food tray assembly scenario considered in the *2023 IEEE Robosoft Competition*. One of the core components of the DSG consists of the two scoops placed at the fingertips. When closed, these parts form a flat base that can be slid under food items, becoming extremely useful to avoid any damage. The two scoops have a total surface area of 120 cm^2 (60 cm^2 each). We also added soft protection walls to the three sides of the scoop to force the food to stay on the gripper base.

Another important aspect of our design is the use of two motors (one per finger) for the tendon driven actuation. This solution helped in connecting the two scoops to the motor housing using a rigid plastic pipe characterized by a length of 20 cm. To actuate the soft-rigid structures

[1] https://robosoft2023.org/2023-competitions/, Accessed 15/09/2023

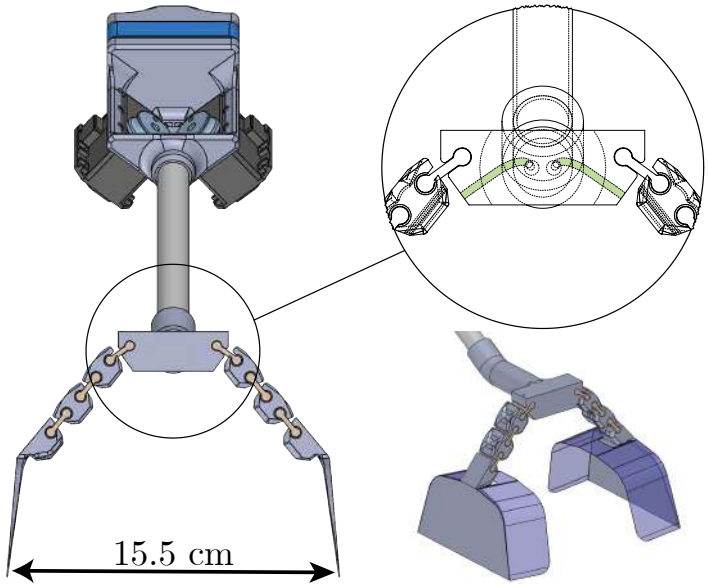


Figure 8.2: Structure of the Double-Scoop Gripper.

that drive the scoops' motion and to redirect the tendon's path, we design a unique structure, highlighted in Fig. 8.2. This part of the gripper was designed to minimize the friction on the tendon by having a smooth curved channel with a large curvature radius, thus avoiding the addition of two pulleys. All these design features allowed us to significantly reduce the gripper's bulkiness, making it possible to grasp objects in narrow spaces. The position of the two actuators and the electronics at the robotic arm end-effector makes the gripper's parts washable.

All the rigid parts, consisting of the motor housing, the motor pulleys, the tendon router part, the rigid modules of the scoops and the scoop base, are made with Elegoo's ABS-like resin, except for the tube, which is a commercial PVC tube. All the flexible parts, the flexible hinges and the food protector around the base are made of 85A shore TPU. We used two Dynamixel MX-28AT motors controlled by an ESP32 microcontroller connected to a custom transistor-transistor logic adapter board. To measure the payload of the gripper, we closed it in such a way to form a flat surface with the two scoops and progressively added weight over them. We stopped as soon as a gap between the scoops started to appear. The resulting gripper payload is 650 g.

8.2.2 Object Detection with ESP32-Camera

Another key feature of the DSG lies in the RGB small camera placed above the fingers. This camera allows to detect objects inside food containers even if the hand or other obstacles are occluding the field of view of an external camera. We employed the camera mainly for the following reasons: i) to detect how many objects are present and ii) where are they with respect to the food container's center. Once we obtained these pieces of information, we were able to adopt different grasping strategies depending on the objects position and quantity.

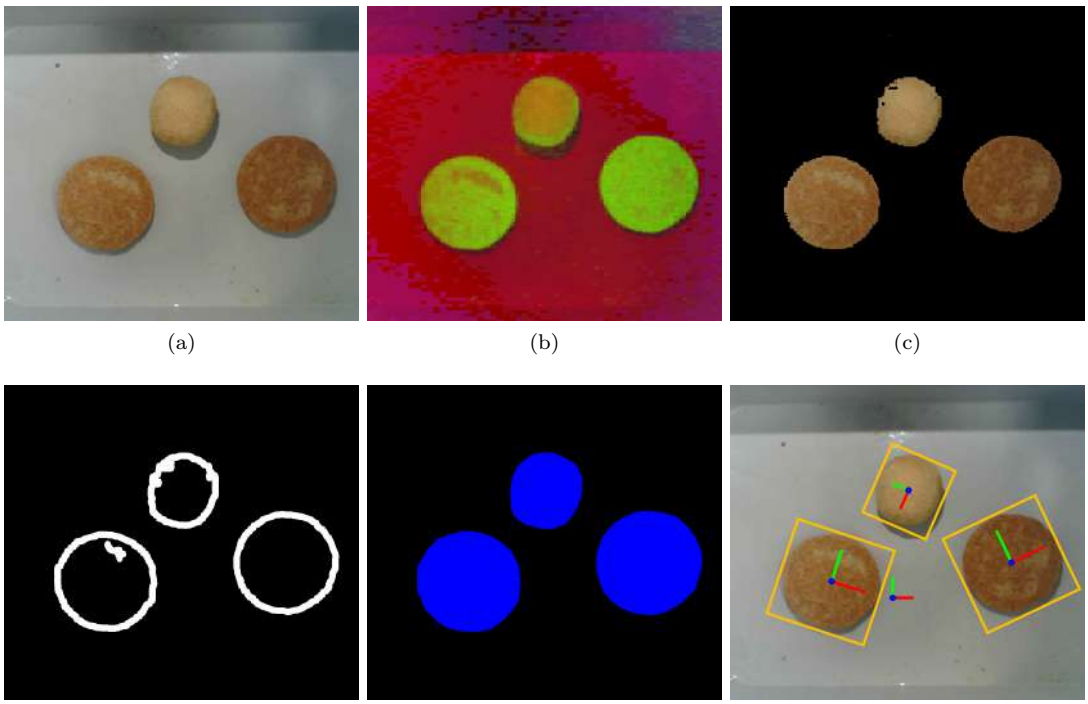


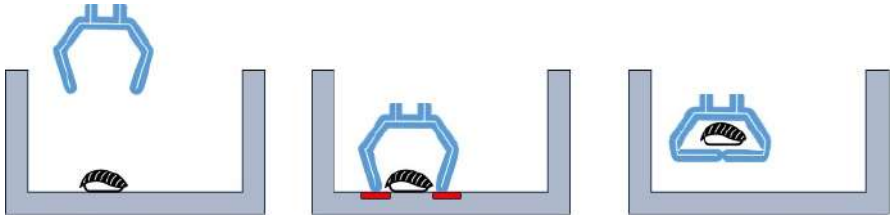
Figure 8.3: Detection of food items in containers: a) RGB streaming, b) HSV conversion, c) background removal, d) extracted contours, e) detected blobs, and f) final image with the bounding boxes and their respective reference frames.

We employed a ESP32-Camera with a resolution of 1280×1024, connected through WiFi to our network. To acquire the images from the ESP32 we used a Real Time Streaming Protocol (RTSP) and we integrated it in ROS with the OpenCV VideoCapture Module [222]. After having obtained the camera image topic (Fig. 8.3a), we calibrated the camera evaluating the camera intrinsic parameters, fundamental in our case to determine the objects position. Then, we applied the following steps to detect and evaluate the position of the objects in the scene: ***Step 1.*** We removed the image background converting the RGB image into the HSV colorspace (Fig. 8.3b) and thresholding with a filter (Fig. 8.3c). ***Step 2.*** Objects contours were found applying a Canny Edge Detection algorithm to find the pixels belonging to the objects edges and an algorithm to retrieve the contours from them (Fig. 8.3d). To improve the algorithm performance we first removed the noise in the image with a 7×7 Gaussian filter. ***Step 3.*** We extracted the image blobs by labeling the different contours, we evaluated their convex hull and we kept the ones which area was higher than an empirical threshold, meaning that they were effectively objects and not noise, as shown in Fig. 8.3e. ***Step 4.*** We determined the objects minimum bounding box (Fig. 8.3f) using the principal component analysis (PCA). ***Step 5.*** Lastly, we converted the objects position from pixels to meters given the intrinsic parameters and the height of the camera.

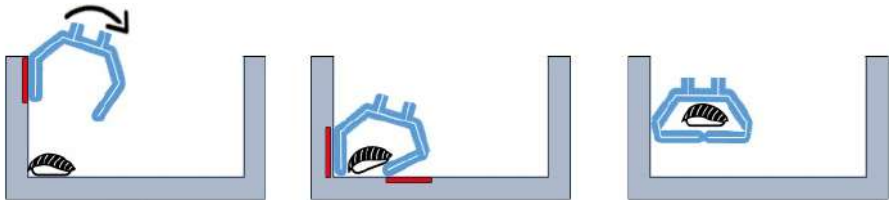
After having performed all the steps, we based our grasping strategy considering the number of detected objects, their dimensions (through the bounding boxes) and their pose with respect to center of the image.

8.2.3 Grasp and release strategies implementation

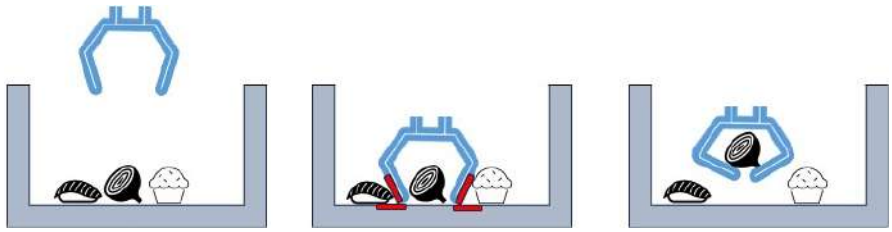
Different grasping strategies have been developed depending on where the food is placed inside a container.



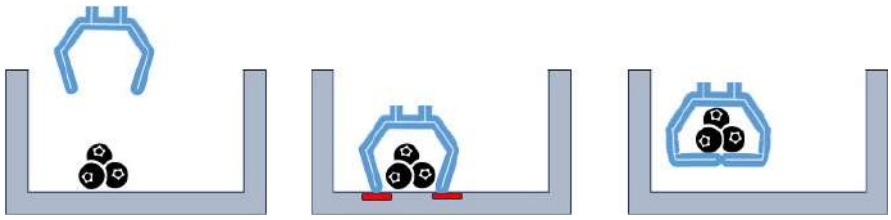
(a) Single and free



(b) Single and wall-constrained



(c) Single between multiple



(d) Multiple

Figure 8.4: Picking strategies depending on the placement and amount of food items. Grey: container walls, blue: gripper, red: areas where the gripper exploits the environment.

The approaches have been determined also considering the part of the container exploited by the scoops. We identified 4 strategies: *i*) unconstrained (i.e., not in contact with the container's

walls) single quantity of food; *ii*) single food item close to the walls of the container; *iii*) single target food among multiple food items; *iv*) multiple target food items close to each other. Fig. 8.4 shows a sketch of the strategies.

Strategies *i)* and *ii)* aim to demonstrate the gripper's effectiveness in exploiting the environment around the single item.

(a) Large with short walls

(b) Large enough to let the gripper open partially

(c) Small enough to impede fingers opening

Figure 8.5: Placing strategies depending on the size of the target container. Grey: container walls, blue: gripper, red: areas where the gripper exploits the environment.

To differentiate them, we evaluated whether the position of the object with respect to the center of the container was within an empirically predetermined threshold. The threshold was calculated considering the longest side of the container and the maximum opening of the gripper. If the distance between the object and the center of the container is less than the threshold, the gripper performs a top grasp taking advantage of the bottom surface of the environment (e.g., as shown in Fig. 8.4a); otherwise, the grasp is made, exploiting the container's wall closer to the scoops (Fig. 8.4b). Strategy *iii*) aims to demonstrate the ability of the DSG to pick a target food when placed among multiple objects by exploiting the scoops to separate them and, at the same time, preventing damage(Fig. 8.4c). Conversely, strategy *iv)* is executed similarly to strategy *i)*, but it deals with multiple objects simultaneously (Fig. 8.4d), suitable for tasks where speed is fundamental). In all the strategies, the opening of the scoops is adjusted according to the size of the food item. Besides, when exploiting the walls, the gripper is tilted to place the scoop

closer to the lateral surface to adapt to it, as shown in the first sketch of Fig. 8.4b.

Three placing strategies were implemented, considering containers of different sizes (as shown in Fig. 8.5). The identified strategies highlight the gripper's ability to adapt to the containers by exploiting their different characteristics. If the release occurs in a large short-walled container, we exploit the environment on the bottom surface of the container to ensure a precise place by the two scoops (Fig. 8.5a). Otherwise, if the container walls are tight enough not to allow the gripper to move on the bottom surface of the container, the scoops are laid on the container's lateral walls to gently release the food item. We considered two possible scenarios: one where the container is large enough to exploit both the scoops (Fig. 8.5b) and another where using both walls is not feasible because the opening of the hand would not be enough to release the food items (Fig. 8.5c).

8.3 Experimental results

8.3.1 Experimental setup

A video of the experimental trials can be found at link.

During the experimental phase, the Double-Scoop Gripper was mounted on a UR5 robotic arm (Fig. 8.1).

To test the performance of the DSG, we conducted a pick-and-place procedure, evaluating separately the results of the two tasks. The aim is to demonstrate that the proposed gripper can grasp and release single and multiple objects by applying the strategies described in Sec. 8.2.3. The picking scenario shown in Fig. 8.1 consists of grabbing food from 3 containers placed at different heights (on a table, between 2 shelves, and above the highest shelf). In the placing scenario, instead, the task is to release the food in containers of different sizes, such as a plate, a bowl, and a glass. For both tasks, we employed 5 real food items of different sizes, shapes, and softness (see Fig. 8.6). We tested each scenario with all the objects 5 times each, collecting 100 and 150 trials for pick and place, respectively[2].

To perform the pick-and-place procedure, we implemented a Finite State Machine (FSM), which consists of the following phases: *i*) the robot is moved to a home pose; *ii*) the gripper is placed above a food container; *iii*) the RGB camera performs a scan searching for the food location and, once detected, estimates its size (see Sec. 8.2.2); *iv*) the gripper moves accordingly above the food item(s), regulating also the opening of the scoops; *v*) the gripper starts descending towards the food item and closes the scoops once it reaches the pre-grasp pose; *vi*) the gripper goes up and then is moved towards the release container; *vii*) the scoops open, delivering the food and accomplishing the task. The positions of the pick and place containers were assumed to be known a priori, although an external camera could be used to detect them.

8.3.2 Results for pick and place

Regarding the picking phase, we evaluated four conditions for each object. A grasp was considered successful if the target food was held inside the gripper until it arrived above the releas-

[2]Note that, although the number of trials differs, we applied the whole procedure described in Sec. 5.3.1 for each pick and place attempt. Thus, to execute a place, a successful grasp is needed.

Figure 8.6: Food items adopted in the experimental trials: sausages, meatballs, carrots, cookies, and zucchini.

Table 8.1: Pick-and-place results. Columns 2-6 contain the results from the picking strategies, while columns 7-13 show the results from the placements. n indicates the number of target food items. The overall results of the two experimental phases are represented in bold.

	Pick					**Place**						
Food items	**Free** $n=1$	**Constrained** $n=1$	**Target** $n=1$	**Free** $n=3$	All (item)	**Plate** $n=1$	**Plate** $n=3$	**Bowl** $n=1$	**Bowl** $n=3$	**Glass** $n=1$	**Glass** $n=3$	All (item)
Meatballs	5/5	5/5	5/5	15/15	30/30	5/5	15/15	5/5	15/15	5/5	13/15	58/60
Cookies	4/5	3/5	3/5	13/15	23/30	5/5	15/15	5/5	15/15	5/5	13/15	58/60
Carrots	5/5	4/5	5/5	15/15	28/30	5/5	15/15	5/5	15/15	5/5	15/15	60/60
Sausages	5/5	2/5	5/5	14/15	26/30	5/5	15/15	5/5	15/15	5/5	15/15	60/60
Zucchinis	5/5	3/5	4/5	14/15	26/30	5/5	15/15	5/5	15/15	3/5	12/15	55/60
All (strategy)	24/25	17/25	22/25	71/75	**133/150**	25/25	75/75	25/25	75/75	23/25	68/75	**291/300**

ing container. Table 8.1 reports the results of the picking phase (from the second to the sixth column). Overall, the DSG successfully grasped 133 out of 150 objects, obtaining a success rate of about 88.6%. In particular, a single quantity of food placed away from the container's walls was picked-up in 24 out of 25 trials, whereas, in constrained cases (i.e., in the proximity of the walls), it was caught in 68% of trials (17 out of 25). Instead, when the target food was placed among others, 22 out of 25 grasps were successful. Lastly, multiple quantity of the same object ($n = 3$) placed away from the walls were grasped in the 94.6% (71 out of 75) of cases.

The placing phase was characterized by more experimental trials with respect to the picking phase because we tested single ($n = 1$) and multiple ($n = 3$) objects in all three placing scenarios. Overall, 291/300 (97%) objects were successfully placed, as shown in the last column of Table 3.4. In more detail, the DSG released all the food items in the plate with a success rate of 100% in both single and multiple cases. The same result was obtained when the bowl was the release container (see ninth and tenth column of Table 3.4). On the other hand, releasing foods into the glass was the most challenging. The gripper successfully placed 23 out of 25 (92%) and 68 out of 75 (90.6%) food items into the glass, in single and multiple quantities of objects, respectively.

(a)

(b)

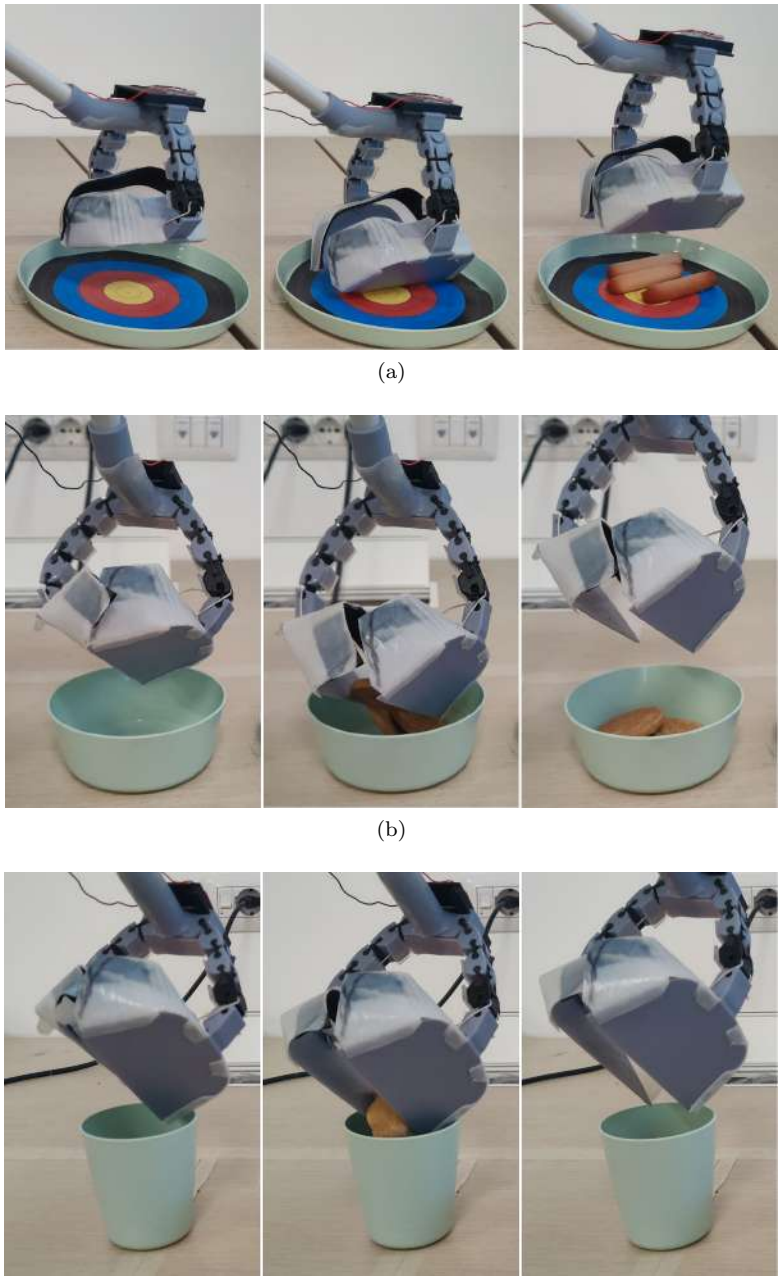
(c)

(d)

Figure 8.7: Sequence of picks. a) Picking of multiple zuchini; b) target picking of a meatball; a) successful wall-pick of a meatball; b) failed wall-pick of a carrot.

8.4 Discussion

Previously presented results indicate that the DSG can successfully grasp and release single and multiple food items of different sizes and shapes. Thanks to the softness of the fingers,



(a)

(b)

(c)

Figure 8.8: Sequence of multiple placement. a) Sausages in the plate; b) cookies in the bowl; c) meatballs in the glass.

it can effectively manipulate irregular, delicate and deformable objects like the ones used in the experimental setup. Besides, the developed strategies fully exploit the gripper features in combination with the environment around the manipulandum (e.g., exploiting the containers

surfaces to help the grasp/release). The delocalization of the gripper from the robotic end-effector was demonstrated to be functional for reaching narrow environments (i.e., the container between the two shelves).

The devised picking strategies work effectively with various objects under different conditions. Exploiting the embedded constraints (i.e., the two scoops), the DSG can delicately handle food items sliding underneath them (Fig. 8.7a) to pick them up without causing any damage. Almost all objects were easily grasped when placed away from the constraints of the container. The only exception is cookies, which were grabbed 4 times out of 5 (80%) and 13 times out of 15 (86.6%) in the single and multiple cases, respectively. The reason of the failures is due to the fact that the scoops were not always able to slide under the object because of the stiffness of the cookie.

On the other hand, grasping becomes quite challenging when the container's wall constrains an object. Meatballs are the only items the DSG was able to grab in all the trials (see Fig. 8.7c)), proving the efficacy of the developed strategy and the ability of the gripper to exploit the constraint, coping with narrow environments. Conversely, the failures are mostly due to the fact that the scoop sometimes tends to lose contact with the wall while sliding, causing involuntary deflection (see Fig. 8.7d). This led to failure, especially with short objects (cookies, zucchini, and sausages). Note that, currently, the sliding of the scoop is controlled in open loop. Thus, a more accurate control of the scoop descent along the wall (e.g., implementing a force control strategy) could increase the effectiveness of the strategy.

Another ability of the DSG is to exploit the scoops to perform precision grasps, as shown in Fig. 8.7b. In this strategy, the role of visual feedback is crucial in detecting the targeted food item. Once located, the gripper adjusts the closing of the scoops to easily separate and catch the target food among multiple objects.

Overall, the experimental trials validated the developed strategies for placing. As it can be noticed from columns 7-13 of Table 3.4, most of the releases were successful for all the objects. We achieved a perfect score for the sausages and carrots, releasing the objects correctly every time. Indeed, their thin shape was helpful, but even with round items like the meatballs and the cookies, we still obtained 96.6% of success rate. Fig. 8.8 reports a successful placement in the three containers.

The most challenging food item was the zucchini, which presented 55 out of 60 successful placements. Moreover, the zucchini was the only item that showed a failure in the single placement. This failure was mainly due to the sticking property of the zucchini's surface, which inhibited the sliding on the scoops when tilting the gripper. Unable to slide, these objects tend to stay in the middle of the scoops, falling outside of the glass during the opening of the fingers. This behavior does not happen when dealing with the other selected food items, resulting in successful placements.

The glass was also the only container where multiple objects failed. Indeed, the issue in this case was caused by the tightness of the container itself, which hindered the placement of more than two items when their size was notable. As can be seen from Table 8.1, carrots and sausages were easily put into the glass three at a time, whereas dealing with meatballs, cookies, and zucchinis, the gripper lost one of the three picked items during the opening of the fingers.

8.5 Conclusions

In this chapter, we proposed the design and application of a novel soft-rigid gripper, the Double-Scoop Gripper. This gripper is specifically tailored to tackle food tray assembly scenarios, where food is placed in restricted spaces. The gripper structure comprises two opposing tendon-driven soft-rigid structures with a rigid component at their tips. To deal with narrow environments, the gripper is situated at the end of a pipe and is equipped with an RGB camera.

We developed four picking and three placing strategies to fully exploit the gripper's characteristics, based on vision inputs. We tested the DSG in a pick-and-place experiment to validate our design. The experimental trials showed that the gripper is able to comply with the fragility of the food items without damaging them and to reach containers in narrow environments. The results demonstrated that the gripper achieves high success rate in both grasp and release exploiting the environment in different conditions.

In some cases, we noticed some limitations of the proposed design, but these can be overcome by modifying the fingers' structure to obtain an improved control over the opening motion to better adapt the scoops to the environment. Future research will also focus on implementing a prismatic joint to substitute the actual fixed pipe to increase the range of motion of the robotic arm. This solution should also improve the reachability of the robot, allowing the planning algorithm to chose configurations that previously were not possible.

Chapter 9

Conclusions

"*The ending is nearer than you think, and it is already written. All that we have left to choose is the correct moment to begin.*"
Alan Moore

Soft manipulation empowers robotic systems to engage with delicate objects and their surroundings, particularly in unstructured and dynamic scenarios. This capability extends the application of robotic manipulators beyond industrial contexts, where interaction with the environment is essential and must avoid causing damage. Consequently, various grasping strategies have been developed to fully leverage the constraints presented by the environment. Nonetheless, these strategies face the challenge of identifying an exploitable environment, and there is a possibility that this may not be feasible during grasp planning. Hence, the control methods and design of embedded constraints outlined in this Thesis offer a solution by eliminating the need to identify the environment separately, as it is inherently integrated into the end-effector itself.

Chapter 3 presents an innovative grasping strategy called scoop grasp, allowing for the grasping of diverse objects using the embedded constraint. The proposed approach is rather versatile as it can determine a suitable pre-grasp pose based solely on the point cloud of the object and the normals to the planes in contact with it. The scoop's capacity to adapt to unstructured environments allows it to slide on surfaces with different inclinations. Thanks to its adaptability and embedded intelligence, the Soft ScoopGripper proves effective in grasping objects in unstructured scenarios. It can successfully employ traditional grasping strategies, such as top grasps or side grasps, without relying on environmental constraints, as its scoop can function as an additional embedded constraint supporting the grasp. Our goal was to demonstrate how this gripper surpasses existing devices by utilizing its unique features to execute innovative grasping strategies where others fall short. For instance, when objects are confined by other objects or environmental features, traditional strategies may struggle, but the scoop makes the task considerably easier. Experimental results support the claim that the Soft ScoopGripper can grasp objects in complex scenarios. The devised grasping strategy proves effective, yet it demands significant computational power and time due to the numerous potential hand configurations. Conversely, a more straightforward and efficient method involves generating grasps based on human demonstrations.

Chapter 4 introduces an approach to extract grasping strategies for soft, non-anthropomorphic hands. This method enables the learning of primitives from human demonstrations without relying on labeled data or lookup tables. The applied methodology is particularly advantageous for soft underactuated hands as it replaces the conventional process of pinpointing and replicating exact contact points between fingers and objects. Human demonstrations are analyzed across various levels of object knowledge, yielding similar grasping strategies in terms of success rates. Considering the object's shape is preferable, and it would be interesting to explore the

generalization of these strategies to more complex objects by decomposing them into simpler shapes, ensuring the applicability of our method. A potential solution involves testing complex-shaped objects by approximating them to the closest shape within the training dataset and then assessing the system's performance – determining if the extracted primitives can be effectively applied to such objects. Further enhancements to the current system could include developing an algorithm for selecting primitives while considering various constraints, such as the presence of obstacles or manipulator reachability. Future developments may also explore the integration of force data into grasping motion, allowing for the full exploitation of the compliance of soft hands and mitigating unpredictable effects of hand interaction with the environment. Alternatively, an effective solution for understanding how to leverage embedded constraints could be the implementation of a fully automated algorithm employing a trial-and-error approach.

Chapter 5 incorporates the Soft ScoopGripper (SSG) with an active exploration Reinforcement Learning algorithm to clear a cluttered scene. Two reinforcement learning policies are trained, the first involving pushing and grasping actions, and the second relying solely on grasping actions. Subsequently, these policies are tested in challenging, cluttered, and unknown scenes. By comparing the outcomes achieved with simpler rigid grippers to those obtained with the SSG, it is demonstrated that the integration of features such as softness, embedded constraints, and model-based algorithms can impact the learning process, influencing the performance of implemented strategies. Additionally, it can be noticed that, with the SSG, a simpler policy learns more quickly and performs better in test scenes than the two-action policy. This outcome results from the inherent exploitation of the environment by the SSG, allowing the hand to insert between two objects to execute a grasp without the need for rearrangement. Future research will delve into developing policies to fully exploit environmental constraints in scenarios where non-prehensile strategies (e.g., pushing) are impractical, such as in restricted scenarios. Multimodal perception during the learning phase will also be incorporated to ascertain the extent to which the gripper can interact with its surroundings. Considering the potential incorporation of embedded constraints into other hands, future work will aim to develop a Reinforcement Learning framework applicable to any gripper featuring a scoop, making the scoop the sole necessary element for transfer learning. Hence, the incorporation of these passive components into the structure of soft grippers is investigated to introduce the specific characteristic of embedded constraints into other hands, without necessitating a complete overhaul of their design or control systems.

In Chapter 6, a methodology to optimize the design parameters of scoop-shaped add-ons is proposed through the utilization of an automated design approach. In contrast to previous efforts in automated design, this approach explicitly considers environmental constraints in the design optimization process. This belief is grounded in the idea that fully leveraging interactions among the hand, object, and environment is crucial for achieving robust grasps, thereby expanding the capacity to execute scoop grasps with diverse gripper types. In testing this approach, it can be observed that optimally designed scoops enhanced the success rate on the original hand, i.e., the SSG. Additionally, they enabled the Pisa/IIT SoftHand to grasp objects that were previously challenging or impossible to grasp. Still, the scoop can be easily detached from the hand when considered a hindrance for the execution of other grasping strategies. In upcoming research endeavors, the aim will be to implement the optimal design to assess and test various hands in cluttered environments using the methodology developed in Chapter 5. Additionally, methods

for designing soft scoops where compliance serves as a functional element will be explored for accomplishing specific tasks. While all the scoops presented feature a flat design, well-suited for insertion between two surfaces, considering different concavities would enhance the scoop's ability to conform to the object's surface. Furthermore, the methods presented focused on improving the grasp success rate and versatility of the scoops, but did not address their structural robustness. Hence, there is a need for a method to optimize the overall structure of the scoop, taking into account its structural mechanics properties.

Chapter 7 introduces a framework designed to enhance the topology optimization of grippers by leveraging data analysis from previously conducted simulations of grasps. The framework comprises three sequential blocks. In the first block, given user-defined parameters, the force signals acting on a gripper component during simulated grasping executions in CoppeliaSim are recorded. The second part of the architecture involves a MATLAB algorithm that analyzes these forces using pixel connectivity and Dynamic Time Warping algorithms. Subsequently, the framework translates these distributions into information required for topology optimization on COMSOL, resulting in an optimized STL of the specified component. The approach is validated by applying it to optimize the embedded constraint. The results demonstrate that the proposed framework effectively enhances the specified component and mitigates undesired behaviors of the gripper. As the current framework operates as an open loop without feedback during optimization, a potential development is the creation of a closed-loop version, where components are optimized and updated after a specific number of iterations. Future research will also center on transitioning this method from simulated to real robotic grippers, exploring the feasibility of acquiring training data through experiments utilizing ad-hoc sensors.

Finally, Chapter 8 presents the design and application of a novel soft-rigid gripper called Double-Scoop Gripper, built upon the control and design methods outlined in this Thesis. This gripper is specifically crafted for food tray assembly scenarios, where the placement of food in confined spaces is a challenge. The gripper structure features two opposing tendon-driven soft-rigid components with rigid tips. To navigate narrow environments, the gripper is affixed to the end of a pipe and incorporates an RGB camera. Four picking and three placing strategies are devised, all leveraging vision inputs to fully exploit the gripper's characteristics. The Double-Scoop Gripper (DSG) undergoes testing in a pick-and-place experiment to validate its design. Experimental trials demonstrate that the gripper adeptly accommodates the fragility of food items without causing damage and effectively reaches containers in restricted spaces. The results showcase a high success rate in both grasping and releasing, demonstrating the gripper's ability to adapt to various environmental conditions.

Based on the obtained results, it can be concluded that the paradigm of embedded constraints introduced in this Thesis holds significant potential in the field of robotic manipulation, particularly as research progresses towards maximizing environmental constraint exploitation with minimal reliance on sensors.

Bibliography

[1] C. Piazza, M. G. Catalano, S. B. Godfrey, M. Rossi, G. Grioli, M. Bianchi, K. Zhao, and A. Bicchi, "The softhand pro-h: A hybrid body-controlled, electrically powered hand prosthesis for daily living and working," *IEEE Robotics & Automation Magazine*, vol. 24, no. 4, pp. 87–101, Dec 2017.

[2] H. Milane, "End-of-arm tools you must consider when automating food," https://www.softroboticsinc.com/resource/3-end-of-arm-tools-you-must-consider-when-automating-food/, March 2020.

[3] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson, "Cable manipulation with a tactile-reactive gripper," *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1385–1401, 2021.

[4] C. Piazza, G. Grioli, M. Catalano, and A. Bicchi, "A century of robotic hands," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, no. 1, pp. 1–32, 2019. [Online]. Available: https://doi.org/10.1146/annurev-control-060117-105003

[5] A. Gafer, D. Heymans, D. Prattichizzo, and G. Salvietti, "The quad-spatula gripper: A novel soft-rigid gripper for food handling," in *Proc. IEEE Int. Conf. on Soft Robotics (RoboSoft)*, New Haeven, USA, 2020, pp. 39–45.

[6] V. Babin and C. Gosselin, "Picking, grasping, or scooping small objects lying on flat surfaces: A design approach," *The International Journal of Robotics Research*, vol. 37, no. 12, pp. 1484–1499, 2018.

[7] A. K. Ramasubramanian, M. Connolly, R. Mathew, and N. Papakostas, "Automatic simulation-based design and validation of robotic gripper fingers," *CIRP Annals*, vol. 71, no. 1, pp. 137–140, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0007850622001007

[8] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4238–4245.

[9] G. Salvietti, Z. Iqbal, M. Malvezzi, T. Eslami, and D. Prattichizzo, "Soft hands with embodied constraints: The soft scoopgripper," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Montreal, Canada, May 2019, pp. 2758–2764.

[10] A. Bicchi and O. Brock, "Editorial," *The International Journal of Robotics Research*, vol. 39, no. 14, pp. 1601–1603, 2020. [Online]. Available: https://doi.org/10.1177/0278364920962422

[11] J. Hughes, U. Culha, F. Giardina, F. Guenther, A. Rosendo, and F. Iida, "Soft manipulators and grippers: A review," *Frontiers in Robotics and AI*, vol. 3, p. 69, 2016. [Online]. Available: http://journal.frontiersin.org/article/10.3389/frobt.2016.00069

[12] A. Albu-Schäffer and A. Bicchi, *Actuators for Soft Robotics*. Cham: Springer International Publishing, 2016, pp. 499–530. [Online]. Available: https://doi.org/10.1007/978-3-319-32552-1_21

[13] C. D. Santina, C. Piazza, G. Grioli, M. G. Catalano, and A. Bicchi, "Towards dexterous manipulation with augmented adaptive synergies: the pisa/iit softhand 2," *IEEE Transactions on Robotics*, In Press 2018.

[14] G. Gu, N. Zhang, H. Xu, S. Lin, Y. Yu, G. Chai, L. Ge, H. Yang, Q. Shao, X. Sheng *et al.*, "A soft neuroprosthetic hand providing simultaneous myoelectric control and tactile feedback," *Nature biomedical engineering*, vol. 7, no. 4, pp. 589–598, 2023.

[15] P. Y. Chua, T. Ilschner, and D. G. Caldwell, "Robotic manipulation of food products–a review," *Industrial Robot: An International Journal*, vol. 30, no. 4, pp. 345–354, 2003.

[16] J. Iqbal, Z. H. Khan, and A. Khalid, "Prospects of robotics in food industry," *Food Science and Technology*, vol. 37, pp. 159–165, 2017.

[17] J. Amend, N. Cheng, S. Fakhouri, and B. Culley, "Soft robotics commercialization: Jamming grippers from research to product," *Soft robotics*, vol. 3, no. 4, pp. 213–222, 2016.

[18] S. Zaidi, M. Maselli, C. Laschi, and M. Cianchetti, "Actuation technologies for soft robot grippers and manipulators: A review," *Current Robotics Reports*, vol. 2, no. 3, pp. 355–369, 2021.

[19] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.

[20] M. Malvezzi and D. Prattichizzo, "Evaluation of grasp stiffness in underactuated compliant hands," in *Proc. 2013 IEEE International Conference on Robotics and Automation (ICRA)*, no. 0, Karlsruhe, Germany, 2013, pp. 2074–2079.

[21] C. Eppner and O. Brock, "Grasping unknown objects by exploiting shape adaptability and environmental constraints," *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.

[22] G. Salvietti, M. Malvezzi, G. Gioioso, and D. Prattichizzo, "Modeling compliant grasps exploiting environmental constraints," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Seattle, USA, 2015.

[23] A. M. Sundaram, W. Friedl, and M. A. Roa, "Environment-aware grasp strategy planning in clutter for a variable stiffness hand," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9377–9384.

[24] D. Yoon and K. Kim, "Fully passive robotic finger for human-inspired adaptive grasping in environmental constraints," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 5, pp. 3841–3852, 2022.

[25] J. Bimbo, E. Turco, M. Ghazaei Ardakani, M. Pozzi, G. Salvietti, V. Bo, M. Malvezzi, and D. Prattichizzo, "Exploiting robot hand compliance and environmental constraints for edge grasps," *Frontiers in Robotics and AI*, p. 135, 2019.

[26] E. Turco, V. Bo, M. Pozzi, A. Rizzo, and D. Prattichizzo, "Grasp planning with a soft reconfigurable gripper exploiting embedded and environmental constraints," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, p. 5215–5222, 2021.

[27] E. Turco, V. Bo, M. Tavassoli, M. Pozzi, and D. Prattichizzo, "Learning grasping strategies for a soft non-anthropomorphic hand from human demonstrations," in *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2022, pp. 934–941.

[28] V. Bo, E. Turco, M. Pozzi, M. Malvezzi, and D. Prattichizzo, "Automated design of embedded constraints for soft hands enabling new grasp strategies," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 346–11 353, 2022.

[29] V. Bo, E. Turco, M. Pozzi, M. Malvezzi, and D. Prattichizzo, "A data-driven topology optimization framework for designing robotic grippers," in *2023 IEEE International Conference on Soft Robotics (RoboSoft)*, 2023, pp. 1–6.

[30] A. Holladay, R. Paolini, and M. T. Mason, "A general framework for open-loop pivoting," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3675–3681.

[31] N. Chavan-Dafle, M. T. Mason, H. Staab, G. Rossano, and A. Rodriguez, "A two-phase gripper to reorient and grasp," in *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, 2015, pp. 1249–1255.

[32] N. Chavan-Dafle, K. Lee, and A. Rodriguez, "Pneumatic shape-shifting fingers to reorient and grasp," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, 2018, pp. 988–993.

[33] X. Li and O. Brock, "Learning from demonstration based on environmental constraints," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 938–10 945, 2022.

[34] K. Haninger, M. Radke, A. Vick, and J. Krüger, "Towards high-payload admittance control for manual guidance with environmental contact," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4275–4282, 2022.

[35] S. G. Khan, G. Herrmann, M. Al Grafi, T. Pipe, and C. Melhuish, "Compliance control and human–robot interaction: Part 1—survey," *International journal of humanoid robotics*, vol. 11, no. 03, p. 1430001, 2014.

[36] C. C. Beltran-Hernandez, D. Petit, I. G. Ramirez-Alpizar, T. Nishi, S. Kikuchi, T. Matsubara, and K. Harada, "Learning force control for contact-rich manipulation tasks with rigid position-controlled robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5709–5716, 2020.

[37] X. Zhang, Y. Zheng, J. Ota, and Y. Huang, "Peg-in-hole assembly based on two-phase scheme and f/t sensor for dual-arm robot," *Sensors*, vol. 17, no. 9, p. 2004, 2017.

[38] J. Shi, J. Z. Woodruff, P. B. Umbanhowar, and K. M. Lynch, "Dynamic in-hand sliding manipulation," *IEEE Transactions on Robotics*, vol. 33, no. 4, pp. 778–795, 2017.

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

[40] Y. Karayiannidis, C. Smith, F. E. Vina, and D. Kragic, "Online contact point estimation for uncalibrated tool use," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2488–2494.

[41] K. Chin, T. Hellebrekers, and C. Majidi, "Machine learning for soft robotic sensing and control," *Advanced Intelligent Systems*, vol. 2, no. 6, p. 1900171, 2020.

[42] A. M. Dollar and R. D. Howe, "Joint coupling design of underactuated grippers," in *30th Annual Mechanisms and Robotics Conference*, 2006, pp. 903–911.

[43] M. G. Catalano, G. Grioli, E. Farnioli, A. Serio, C. Piazza, and A. Bicchi, "Adaptive synergies for the design and control of the pisa/iit softhand," *The International Journal of Robotics Research*, vol. 33, no. 5, pp. 768–782, 2014.

[44] A. Milojević, S. Linß, and H. Handroos, "Soft robotic compliant two-finger gripper mechanism for adaptive and gentle food handling," in *2021 IEEE 4th International Conference on Soft Robotics (RoboSoft)*, 2021, pp. 163–168.

[45] R. Deimel and O. Brock, "A compliant hand based on a novel pneumatic actuator," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2047–2053.

[46] Z. Wang, Y. Torigoe, and S. Hirai, "A prestressed soft gripper: Design, modeling, fabrication, and tests for food handling," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 1909–1916, 2017.

[47] A. T. Miller and P. K. Allen, "Graspit!: A versatile simulator for grasp analysis," *IEEE Robotics & Automation Magazine*, 2004.

[48] M. Malvezzi, G. Gioioso, G. Salvietti, and D. Prattichizzo, "Syngrasp: A matlab toolbox for underactuated and compliant hands," *IEEE Robotics & Automation Magazine*, vol. 22, no. 4, pp. 52–68, 2015.

[49] O. Brock, J. Park, and M. Toussaint, *Mobility and Manipulation*. Cham: Springer International Publishing, 2016, pp. 1007–1036. [Online]. Available: https://doi.org/10.1007/978-3-319-32552-1_40

[50] M. Pozzi, S. Marullo, G. Salvietti, J. Bimbo, M. Malvezzi, and D. Prattichizzo, "Hand closure model for planning top grasps with soft robotic hands," *The International Journal of Robotics Research*, vol. 0, no. 0, 2020.

[51] C. Della Santina, V. Arapi, G. Averta, F. Damiani, G. Fiore, A. Settimi, M. G. Catalano, D. Bacciu, A. Bicchi, and M. Bianchi, "Learning from humans how to grasp: a data-driven architecture for autonomous grasping with anthropomorphic soft hands," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1533–1540, 2019.

[52] C. Gabellieri, F. Angelini, V. Arapi, A. Palleschi, M. G. Catalano, G. Grioli, L. Pallottino, A. Bicchi, M. Bianchi, and M. Garabini, "Grasp it like a pro: Grasp of unknown objects with robotic hands based on skilled human expertise," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2808–2815, 2020.

[53] R. Deimel, C. Eppner, J. Álvarez-Ruiz, M. Maertens, and O. Brock, "Exploitation of environmental constraints in human and robotic grasping," in *Robotics Research*. Springer, 2016, pp. 393–409.

[54] E. Klingbeil, S. Menon, and O. Khatib, "Experimental analysis of human control strategies in contact manipulation tasks," in *2016 International Symposium on Experimental Robotics*. Springer, 2017, pp. 275–286.

[55] A. Altobelli, O. Tokatli, G. Burroughes, and R. Skilton, "Optimal grasping pose synthesis in a constrained environment," *Robotics*, vol. 10, no. 1, 2021. [Online]. Available: https://www.mdpi.com/2218-6581/10/1/4

[56] M. Gabiccini, A. Artoni, G. Pannocchia, and J. Gillis, *A Computational Framework for Environment-Aware Robotic Manipulation Planning*. Cham: Springer International Publishing, 2018, pp. 363–385. [Online]. Available: https://doi.org/10.1007/978-3-319-60916-4_21

[57] M. Suomalainen, Y. Karayiannidis, and V. Kyrki, "A survey of robot manipulation in contact," *Robotics and Autonomous Systems*, vol. 156, p. 104224, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889022001312

[58] L. U. Odhner, R. R. Ma, and A. M. Dollar, "Open-loop precision grasping with underactuated hands inspired by a human manipulation strategy," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 625–633, 2013.

[59] S. Puhlmann, F. Heinemann, O. Brock, and M. Maertens, "A compact representation of human single-object grasping," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1954–1959.

[60] C. Della Santina, M. Bianchi, G. Averta, S. Ciotti, V. Arapi, S. Fani, E. Battaglia, M. G. Catalano, M. Santello, and A. Bicchi, "Postural hand synergies during environmental constraint exploitation," *Frontiers in Neurorobotics*, vol. 11, 2017. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnbot.2017.00041

[61] M. Pozzi, M. Malvezzi, and D. Prattichizzo, "On grasp quality measures: Grasp robustness and contact force distribution in underactuated and compliant robotic hands," *IEEE Robotics and Automation Letters*, vol. 2, 2017.

[62] A. Altobelli, O. Tokatli, G. Burroughes, and R. Skilton, "Optimal grasping pose synthesis in a constrained environment," *Robotics*, vol. 10, no. 1, 2021. [Online]. Available: https://www.mdpi.com/2218-6581/10/1/4

[63] S. Ma, L. Du, E. Tsuchiya, and M. Fuchimi, "Paper-made grippers for soft food grasping," in *2020 17th International Conference on Ubiquitous Robots (UR)*, 2020, pp. 362–367.

[64] Z. Wang, H. Furuta, S. Hirai, and S. Kawamura, "A scooping-binding robotic gripper for handling various food products," *Frontiers in Robotics and AI*, vol. 8, 2021. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frobt.2021.640805

[65] W. Friedl and M. A. Roa, "Clash wrist - a hardware to increase the capability of clash fruit gripper to use environment constraints exploration," 11 2019.

[66] F. Negrello, S. Mghames, G. Grioli, M. Garabini, and M. G. Catalano, "A compact soft articulated parallel wrist for grasping in narrow spaces," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3161–3168, 2019.

[67] A. Sahbani, S. El-Khoury, and P. Bidaud, "An overview of 3d object grasp synthesis algorithms," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 326–336, 2012.

[68] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis - a survey," vol. 30, no. 2, pp. 289–309, 2014.

[69] K. B. Shimoga, "Robot grasp synthesis algorithms: A survey," *The International Journal of Robotics Research*, vol. 15, no. 3, pp. 230–266, 1996.

[70] M. Roa, R. Suárez, and J. Cornelia, "Medidas de calidad para la prensión de objetos," *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 5, no. 1, pp. 66–82, 2008.

[71] J. Cornelia and R. Suarez, "Determining independent grasp regions on 2d discrete objects," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 2941–2946.

[72] M. A. Roa and R. Suarez, "Grasp quality measures: review and performance," *Autonomous Robots*, vol. 38, no. 1, pp. 65–88, 2015.

[73] Z. Li and S. S. Sastry, "Task-oriented optimal grasping by multifingered robot hands," *IEEE Journal on Robotics and Automation*, vol. 4, no. 1, pp. 32–44, Feb 1988.

[74] B.-H. Kim, S.-R. Oh, B.-J. Yi, and I. H. Suh, "Optimal grasping based on non-dimensionalized performance indices," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 2, 2001, pp. 949–956 vol.2.

[75] T. Yoshikawa, "Manipulability of robotic mechanisms," *The international journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[76] R. Balasubramanian, L. Xu, P. D. Brook, J. R. Smith, and Y. Matsuoka, "Human-guided grasp measures improve grasp robustness on physical robot," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 2294–2301.

[77] Y. C. Park and G. P. Starr, "Grasp synthesis of polygonal objects using a three-fingered robot hand," *The International journal of robotics research*, vol. 11, no. 3, pp. 163–184, 1992.

[78] M. B. Horowitz and J. W. Burdick, "Combined grasp and manipulation planning as a trajectory optimization problem," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 584–591.

[79] J. Ichnowski, M. Danielczuk, J. Xu, V. Satish, and K. Goldberg, "Gomp: Grasp-optimized motion planning for bin picking," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 5270–5277.

[80] O. B. Kroemer, R. Detry, J. Piater, and J. Peters, "Combining active learning and reactive control for robot grasping," *Robotics and Autonomous systems*, vol. 58, no. 9, pp. 1105–1116, 2010.

[81] J. Aleotti and S. Caselli, "Interactive teaching of task-oriented robot grasps," *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 539–550, 2010.

[82] R. Jäkel, S. R. Schmidt-Rohr, Z. Xue, M. Lösch, and R. Dillmann, "Learning of probabilistic grasping strategies using programming by demonstration," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 873–880.

[83] L. M. Pedro, V. L. Belini, and G. A. Caurin, "Learning how to grasp based on neural network retraining," *Advanced Robotics*, vol. 27, no. 10, pp. 785–797, 2013.

[84] F. Leoni, M. Guerrini, C. Laschi, D. Taddeucci, P. Dario, and A. Starita, "Implementing robotic grasping tasks using a biological approach," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, vol. 3, 1998, pp. 2274–2280 vol.3.

[85] G. A. Bekey, H. Liu, R. Tomovic, and W. J. Karplus, "Knowledge-based control of grasping in robot hands using heuristics from human motor skills," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 6, pp. 709–722, 1993.

[86] K. Kleeberger, R. Bormann, W. Kraus, and M. Huber, "A survey on learning-based robotic grasping," *Current Robotics Reports*, vol. 1, p. 239–249, 12 2020.

[87] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, April 2014.

[88] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.

[89] M. Gualtieri and R. Platt, "Viewpoint selection for grasp detection," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 258–264.

[90] D. Kappler, J. Bohg, and S. Schaal, "Leveraging big data for grasp planning," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4304–4311.

[91] A. Pervez, Y. Mao, and D. Lee, "Learning deep movement primitives using convolutional neural networks," in *2017 IEEE-RAS 17th international conference on humanoid robotics (Humanoids)*. IEEE, 2017, pp. 191–197.

[92] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[93] D. Morrison, P. Corke, and J. Leitner, "Learning robust, real-time, reactive robotic grasping," *The International journal of robotics research*, vol. 39, no. 2-3, pp. 183–201, 2020.

[94] S. Kumra, S. Joshi, and F. Sahin, "Gr-convnet v2: A real-time multi-grasp detection network for robotic grasping," *Sensors*, vol. 22, no. 16, p. 6208, 2022.

[95] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," *arXiv preprint arXiv:1804.05172*, 2018.

[96] V. Satish, J. Mahler, and K. Goldberg, "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1357–1364, 2019.

[97] E. Johns, S. Leutenegger, and A. Davison, "Deep learning a grasp function for grasping under gripper pose uncertainty," 10 2016, pp. 4461–4468.

[98] S. El-Khoury and A. Sahbani, "Handling objects by their handles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, no. POST_TALK, 2008.

[99] X. Yan, J. Hsu, M. Khansari, Y. Bai, A. Pathak, A. Gupta, J. Davidson, and H. Lee, "Learning 6-dof grasping interaction via deep geometry-aware 3d representations," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3766–3773.

[100] Ü. R. Aktaş, C. Zhao, M. Kopicki, and J. L. Wyatt, "Deep dexterous grasping of novel objects from a single view," *International Journal of Humanoid Robotics*, vol. 19, no. 02, p. 2250011, 2022.

[101] D. Rao, Q. V. Le, T. Phoka, M. Quigley, A. Sudsang, and A. Y. Ng, "Grasping novel objects with depth segmentation," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 2578–2585.

[102] R. Pelossof, A. Miller, P. Allen, and T. Jebara, "An svm learning approach to robotic grasping," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 4. IEEE, 2004, pp. 3512–3518.

[103] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[104] C. Choi, W. Schwarting, J. DelPreto, and D. Rus, "Learning object grasping for soft robot hands," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2370–2377, 2018.

[105] A. G. Billard, S. Calinon, and R. Dillmann, *Learning from Humans*. Cham: Springer International Publishing, 2016, pp. 1995–2014. [Online]. Available: https://doi.org/10.1007/978-3-319-32552-1_74

[106] N. Wang, C. Chen, and A. Di Nuovo, "A framework of hybrid force/motion skills learning for robots," *IEEE Transactions on Cognitive and Developmental Systems*, 2020.

[107] K. Khokar, R. Alqasemi, S. Sarkar, K. Reed, and R. Dubey, "A novel telerobotic method for human-in-the-loop assisted grasping based on intention recognition," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 4762–4769.

[108] W. Si, N. Wang, and C. Yang, "A review on manipulation skill acquisition through teleoperation-based learning from demonstration," *Cognitive Computation and Systems*, vol. 3, no. 1, pp. 1–16, 2021. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/ccs2.12005

[109] M. Hersch, F. Guenter, S. Calinon, and A. Billard, "Dynamical system modulation for robot learning via kinesthetic demonstrations," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1463–1467, 2008.

[110] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of artificial intelligence research*, vol. 4, pp. 129–145, 1996.

[111] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 2, p. 286–298, 2007.

[112] S. Calinon, F. D'halluin, E. Sauser, D. Caldwell, and A. Billard, "Learning and reproduction of gestures by imitation: An approach based on hidden markov model and gaussian mixture regression," *IEEE Robotics and Automation Magazine*, vol. 17, no. 2, pp. 44–54, 2010. [Online]. Available: http://infoscience.epfl.ch/record/147286

[113] B. Huang, S. El-Khoury, M. Li, J. J. Bryson, and A. Billard, "Learning a real time grasping strategy," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 593–600.

[114] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 2. IEEE, 2002, pp. 1398–1403.

[115] G. Hovland, P. Sikka, and B. McCarragher, "Skill acquisition from human demonstration using a hidden markov model," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, 1996, pp. 2706–2711 vol.3.

[116] A. Paraschos, C. Daniel, J. Peters, G. Neumann *et al.*, "Probabilistic movement primitives," *Advances in neural information processing systems*, 2013.

[117] S. Calinon, F. Guenter, and A. Billard, "On learning the statistical representation of a task and generalizing it to various contexts," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 2978–2983.

[118] M. Racca, J. Pajarinen, A. Montebelli, and V. Kyrki, "Learning in-contact control strategies from demonstration," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 688–695.

[119] R. Detry, D. Kraft, A. G. Buch, N. Krüger, and J. Piater, "Refining grasp affordance models by experience," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 2287–2293.

[120] R. Detry, D. Kraft, O. Kroemer, L. Bodenhagen, J. Peters, N. Krüger, and J. Piater, "Learning grasp affordance densities," *Paladyn*, vol. 2, no. 1, pp. 1–17, 2011.

[121] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke *et al.*, "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Conference on Robot Learning*. PMLR, 2018, pp. 651–673.

[122] R. Newbury, M. Gu, L. Chumbley, A. Mousavian, C. Eppner, J. Leitner, J. Bohg, A. Morales, T. Asfour, D. Kragic *et al.*, "Deep learning approaches to grasp synthesis: A review," *IEEE Transactions on Robotics*, 2023.

[123] S. Bhagat, H. Banerjee, Z. T. Ho Tse, and H. Ren, "Deep reinforcement learning for soft, flexible robots: Brief review with impending challenges," *Robotics*, vol. 8, no. 1, p. 4, 2019.

[124] O. Kroemer, R. Detry, J. Piater, and J. Peters, "Combining active learning and reactive control for robot grasping," *Robotics and Autonomous Systems*, vol. 58, no. 9, pp. 1105 – 1116, September 2010.

[125] F. Stulp, E. Theodorou, M. Kalakrishnan, P. Pastor, L. Righetti, and S. Schaal, "Learning motion primitive goals for robust manipulation," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 325–331.

[126] M. Gualtieri and R. Platt, "Learning 6-dof grasping and pick-place using attention focus," in *Conference on Robot Learning*. PMLR, 2018, pp. 477–486.

[127] L. Berscheid, P. Meißner, and T. Kröger, "Robot learning of shifting objects for grasping in cluttered environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 612–618.

[128] S. Song, A. Zeng, J. Lee, and T. Funkhouser, "Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4978–4985, 2020.

[129] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6292–6299.

[130] M. Vecerik, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothörl, T. Lampe, and M. Riedmiller, "Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards," *arXiv preprint arXiv:1707.08817*, 2017.

[131] F. Guenter, M. Hersch, S. Calinon, and A. Billard, "Reinforcement learning for imitating constrained reaching movements," *Advanced Robotics*, vol. 21, no. 13, pp. 1521–1544, 2007.

[132] A. Gupta, C. Eppner, S. Levine, and P. Abbeel, "Learning dexterous manipulation for a soft robotic hand from human demonstrations," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 3786–3793.

[133] J. R. Napier, "The prehensile movements of the human hand," *The Journal of bone and joint surgery. British volume*, vol. 38, no. 4, pp. 902–913, 1956.

[134] M. T. Mason and J. K. Salisbury Jr, "Robot hands and the mechanics of manipulation," 1985.

[135] L. Wang, J. DelPreto, S. Bhattacharyya, J. Weisz, and P. K. Allen, "A highly-underactuated robotic hand with force and joint angle sensors," in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2011, pp. 1380–1385.

[136] L. Birglen, T. Laliberté, and C. Gosselin, *Underactuated robotic hands*, ser. Springer Tracts in Advanced Robotics. Springer Verlag, 2008, vol. 40.

[137] S. Q. Liu and E. H. Adelson, "Gelsight fin ray: Incorporating tactile sensing into a soft compliant robotic gripper," in *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)*, 2022, pp. 925–931.

[138] R. Deimel, "Soft Robotic Hands for Compliant Grasping," PhD Thesis, TU Berlin, Berlin, 2017.

[139] R. J. Schwarz and C. Taylor, "The anatomy and mechanics of the human hand," *Artificial limbs*, vol. 2, no. 2, pp. 22–35, 1955.

[140] H. Griffiths *et al.*, "Treatment of the injured workman." *Lancet*, pp. 729–33, 1943.

[141] M. Controzzi, C. Cipriani, and M. C. Carrozza, *Design of Artificial Hands: A Review*. Cham: Springer International Publishing, 2014, pp. 219–246. [Online]. Available: https://doi.org/10.1007/978-3-319-03017-3_11

[142] C. Lovchik and M. Diftler, "The robonaut hand: a dexterous robot hand for space," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 2, 1999, pp. 907–912 vol.2.

[143] J. Butterfass, M. Grebenstein, H. Liu, and G. Hirzinger, "DLR-hand II: next generation of a dextrous robot hand," in *Proceedings IEEE International Conference on Robotics and Automation, ICRA*, vol. 1, 2001, pp. 109–114.

[144] R. Deimel and O. Brock, "A novel type of compliant and underactuated robotic hand for dexterous grasping," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 161–185, 2016.

[145] G. Smit, D. Plettenburg, and F. van der Helm, "The lightweight delft cylinder hand, the first multi-articulating hand that meets the basic user requirements," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, 08 2014.

[146] M. T. Zhang and K. Goldberg, "Designing robot grippers: optimal edge contacts for part alignment," *Robotica*, vol. 25, no. 3, pp. 341–349, 2006.

[147] M. Honarpardaz, M. Tarkian, J. Ölvander, and X. Feng, "Finger design automation for industrial robot grippers: A review," *Robotics and Autonomous Systems*, vol. 87, pp. 104–119, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889015303171

[148] D. T. Pham, N. S. Gourashi, and E. E. Eldukhri, "Automated configuration of gripper systems for assembly tasks," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 221, no. 11, pp. 1643–1649, 2007.

[149] F. Sanfilippo, G. Salvietti, H. Zhang, H. P. Hildre, and D. Prattichizzo, "Efficient modular grasping: an iterative approach," in *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 2012, pp. 1281–1286.

[150] F. Sanfilippo, H. Zhang, K. Y. Pettersen, G. Salvietti, and D. Prattichizzo, "Modgrasp: An open-source rapid-prototyping framework for designing low-cost sensorised modular hands," in *5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, 2014, pp. 951–957.

[151] F. Sanfilippo and K. Y. Pettersen, "Openmrh: A modular robotic hand generator plugin for openrave," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2015, pp. 1–6.

[152] R. G. Brown and R. C. Brost, "A 3-d modular gripper design tool," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 1, pp. 174–186, 1999.

[153] L. Balan and G. M. Bone, "Automated gripper jaw design and grasp planning for sets of 3d objects," *Journal of Robotic Systems*, vol. 20, no. 3, pp. 147–162, 2003.

[154] R. C. Brost and R. R. Peters, "Automatic design of 3-d fixtures and assembly pallets," *The International Journal of Robotics Research*, vol. 17, no. 12, pp. 1243–1281, 1998.

[155] V. Velasco and W. S. Newman, "Computer-assisted gripper and fixture customization using rapid-prototyping technology," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, vol. 4. IEEE, 1998, pp. 3658–3664.

[156] P. Pedrazzoli, R. Rinaldi, and C. R. Boer, "A rule based approach to the gripper selection issue for the assembly process," in *Proceedings of the 2001 IEEE International Symposium on Assembly and Task Planning (ISATP2001). Assembly and Disassembly in the Twenty-first Century.(Cat. No. 01TH8560)*. IEEE, 2001, pp. 202–207.

[157] V. B. Velasco Jr and W. S. Newman, "An approach to automated gripper customization using rapid prototyping technology," Citeseer, Tech. Rep., 1996.

[158] M. Honarpardaz, M. Tarkian, J. Ölvander, and X. Feng, "Experimental verification of design automation methods for robotic finger," *Robotics and Autonomous Systems*, vol. 94, pp. 89–101, 2017.

[159] M. P. Bendsoe and O. Sigmund, *Topology optimization: theory, methods, and applications*. Springer Science & Business Media, 2003.

[160] J. Liang, X. Zhang, and B. Zhu, "Nonlinear topology optimization of parallel-grasping microgripper," *Precision Engineering*, vol. 60, pp. 152–159, 2019.

[161] H. Zhang, M. Y. Wang, F. Chen, Y. Wang, A. S. Kumar, and J. Y. Fuh, "Design and development of a soft gripper with topology optimization," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 6239–6244.

[162] C.-H. Liu, C.-H. Chiu, M.-C. Hsu, Y. Chen, and Y.-P. Chiang, "Topology and size–shape optimization of an adaptive compliant gripper with high mechanical advantage for grasping irregular objects," *Robotica*, vol. 37, no. 8, pp. 1383–1400, 2019.

[163] C.-H. Liu, T.-L. Chen, C.-H. Chiu, M.-C. Hsu, Y. Chen, T.-Y. Pai, W.-G. Peng, and Y.-P. Chiang, "Optimal design of a soft robotic gripper for grasping unknown objects," *Soft robotics*, vol. 5, no. 4, pp. 452–465, 2018.

[164] J. Hiller and H. Lipson, "Automatic design and manufacture of soft robots," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 457–466, 2011.

[165] R. Wang, X. Zhang, B. Zhu, H. Zhang, B. Chen, and H. Wang, "Topology optimization of a cable-driven soft robotic gripper," *Structural and Multidisciplinary Optimization*, vol. 62, no. 5, pp. 2749–2763, 2020.

[166] G. Maloisel, E. Knoop, C. Schumacher, and M. Bächer, "Automated routing of muscle fibers for soft robots," *IEEE Transactions on Robotics*, vol. 37, no. 3, pp. 996–1008, 2021.

[167] F. Chen, W. Xu, H. Zhang, Y. Wang, J. Cao, M. Y. Wang, H. Ren, J. Zhu, and Y. Zhang, "Topology optimized design, fabrication, and characterization of a soft cable-driven gripper," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2463–2470, 2018.

[168] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *IEEE Transactions on robotics*, vol. 30, no. 2, pp. 289–309, 2013.

[169] J. Kim, K. Iwamoto, J. J. Kuffner, Y. Ota, and N. S. Pollard, "Physically based grasp quality evaluation under pose uncertainty," *IEEE Transactions on Robotics*, vol. 29, no. 6, pp. 1424–1439, 2013.

[170] A. Wolniakowski, K. Miatliuk, N. Krüger, and J. A. Rytz, "Automatic evaluation of task-focused parallel jaw gripper design," in *Simulation, Modeling, and Programming for Autonomous Robots: 4th International Conference, SIMPAR 2014, Bergamo, Italy, October 20-23, 2014. Proceedings 4*. Springer, 2014, pp. 450–461.

[171] R. Deimel, P. Irmisch, V. Wall, and O. Brock, "Automated co-design of soft hand morphology and control strategy for grasping," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1213–1218.

[172] F. Hagelskjær, A. Kramberger, A. Wolniakowski, T. R. Savarimuthu, and N. Krüger, "Combined optimization of gripper finger design and pose estimation processes for advanced industrial assembly," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 2022–2029.

[173] C. Hazard, N. Pollard, and S. Coros, "Automated design of robotic hands for in-hand manipulation tasks," *International Journal of Humanoid Robotics*, vol. 17, no. 01, p. 1950029, 2020.

[174] K. Kosuge, J. Lee, J. Ichinose, and Y. Hirata, "A novel grasping mechanism for flat-shaped objects inspired by lateral grasp," in *2008 2nd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics*. IEEE, 2008, pp. 282–288.

[175] F. Levesque, B. Sauvet, P. Cardou, and C. Gosselin, "A model-based scooping grasp for the autonomous picking of unknown objects with a two-fingered gripper," *Robotics and Autonomous Systems*, vol. 106, pp. 14–25, 2018.

[176] L. Chin, F. Barscevicius, J. Lipton, and D. Rus, "Multiplexed manipulation: Versatile multimodal grasping via a hybrid soft gripper," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8949–8955.

[177] D. Prattichizzo and J. C. Trinkle, "Grasping," in *Springer handbook of robotics*, B. Siciliano and O. Khatib, Eds. Springer Science & Business Media, 2016, pp. 955–988.

[178] C. R. Houck, J. Joines, and M. G. Kay, "A genetic algorithm for function optimization: a matlab implementation," *Ncsu-ie tr*, vol. 95, no. 09, pp. 1–10, 1995.

[179] J. Heitkoetter and D. Beasley, "The hitch-hiker's guide to evolutionary computation: A list of frequently asked questions (faq). usenet: comp. ai. genetic, available via anonymous ftp from rtfm," 1996.

[180] M. Santello, M. Flanders, and J. F. Soechting, "Postural hand synergies for tool use," *The Journal of Neuroscience*, vol. 18, no. 23, pp. 10 105–10 115, 1998.

[181] X.-S. Yang, "Nature-inspired optimization algorithms: Challenges and open problems," *Journal of Computational Science*, vol. 46, p. 101104.

[182] R. c. Willow Garage, "ORK Object Recognition Kitchen," https://github.com/wg-perception/object_recognition_core.

[183] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[184] N. Amenta, M. Bern, and M. Kamvysselis, "A new voronoi-based surface reconstruction algorithm," in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 1998, pp. 415–421.

[185] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set," *Robot. Automat. Mag.*, vol. 22, 2015.

[186] D. R. Faria, R. Martins, J. Lobo, and J. Dias, "Extracting data from human manipulation of objects towards improving autonomous robotic grasping," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 396–410, 2012, autonomous Grasping. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889011001527

[187] M. Liarokapis, P. Artemiadis, and K. Kyriakopoulos, "Telemanipulation with the dlr/hit ii robot hand using a dataglove and a low cost force feedback device," 10 2013.

[188] P. Goyal, P. Shukla, and G. C. Nandi, "Regression based robotic grasp detection using deep learning and autoencoders," in *2020 IEEE 4th Conference on Information Communication Technology (CICT)*, 2020, pp. 1–6.

[189] E. De Coninck, T. Verbelen, P. Van Molle, P. Simoens, and B. Dhoedt, "Learning robots to grasp by demonstration," *Robotics and Autonomous Systems*, vol. 127, p. 103474, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889019304956

[190] M. A. c. Riedlinger, M. Voelk, K. Kleeberger, M. U. Khalid, and R. Bormann, "Model-free grasp learning framework based on physical simulation," in *ISR 2020; 52th International Symposium on Robotics*, 2020, pp. 1–8.

[191] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[192] H. G. Sung, *Gaussian mixture regression and classification.* Rice University, 2004.

[193] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, pp. 69–84, 2007.

[194] R. Smits, "KDL: Kinematics and dynamics library," http://orocos.org/kdl.

[195] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 922–928.

[196] M. Bonilla, D. Resasco, M. Gabiccini, and A. Bicchi, "Grasp planning with soft hands using bounding box object decomposition," in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, Sept 2015, pp. 518–523.

[197] L. Chevalier, F. Jaillet, and A. Baskurt, "Segmentation and superquadric modeling of 3d objects," 2003.

[198] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," *Science Robotics*, vol. 4, no. 26, p. eaau4984, 2019.

[199] J. Hughes, U. Culha, F. Giardina, F. Guenther, A. Rosendo, and F. Iida, "Soft manipulators and grippers: A review," *Frontiers in Robotics and AI*, vol. 3, p. 69, 2016. [Online]. Available: https://www.frontiersin.org/article/10.3389/frobt.2016.00069

[200] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[201] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition.* Ieee, 2009, pp. 248–255.

[202] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *arXiv preprint arXiv:1511.05952*, 2015.

[203] B. Wu, I. Akinola, and P. K. Allen, "Pixel-attentive policy gradient for multi-fingered grasping in cluttered scenes," in *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS).* IEEE, 2019, pp. 1789–1796.

[204] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox, "6-dof grasping for target-driven object manipulation in clutter," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 6232–6238.

[205] N. Kitaev, I. Mordatch, S. Patil, and P. Abbeel, "Physics-based trajectory optimization for grasping in cluttered environments," in *2015 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 2015, pp. 3102–3109.

[206] S. Hasegawa, K. Wada, Y. Niitani, K. Okada, and M. Inaba, "A three-fingered hand with a suction gripping system for picking various objects in cluttered narrow space," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2017, pp. 1164–1171.

[207] M. Dogar, K. Hsiao, M. Ciocarlie, and S. Srinivasa, "Physics-based grasp planning through clutter," in *Proceedings of Robotics: Science and Systems (RSS '12)*, July 2012.

[208] R. Datta, S. Pradhan, and B. Bhattacharya, "Analysis and design optimization of a robotic gripper using multiobjective genetic algorithm," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 1, pp. 16–26, 2015.

[209] H. Dong, E. Asadi, C. Qiu, J. Dai, and I.-M. Chen, "Geometric design optimization of an under-actuated tendon-driven robotic gripper," *Robotics and Computer-Integrated Manufacturing*, vol. 50, pp. 80–89, 2018.

[210] F. Chen and M. Y. Wang, "Design optimization of soft robots: A review of the state of the art," *IEEE Robotics & Automation Magazine*, vol. 27, no. 4, pp. 27–43, 2020.

[211] H. Zhang, A. S. Kumar, J. Y. Fuh, and M. Y. Wang, "Topology optimized design, fabrication and evaluation of a multimaterial soft gripper," in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2018, pp. 424–430.

[212] E. Rohmer, S. P. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1321–1326.

[213] C. Multiphysics, "Introduction to comsol multiphysics®," *COMSOL Multiphysics, Burlington, MA, accessed Feb*, vol. 9, no. 2018, p. 32, 1998.

[214] X. Zhu, T. Feng, and H. Culbertson, "Understanding the effect of speed on human emotion perception in mediated social touch using voice coil actuators," *Frontiers in Computer Science*, vol. 4, 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fcomp.2022.826637

[215] K. Svanberg, "The method of moving asymptotes—a new method for structural optimization," *International journal for numerical methods in engineering*, vol. 24, no. 2, pp. 359–373, 1987.

[216] F. Bader and S. Rahimifard, "A methodology for the selection of industrial robots in food handling," *Innovative Food Science and Emerging Technologies*, vol. 64, p. 102379, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1466856420303258

[217] G. Endo and N. Otomo, "Development of a food handling gripper considering an appetizing presentation," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 4901–4906.

[218] I. Naotunna, C. J. Perera, C. Sandaruwan, R. Gopura, and T. D. Lalitharatne, "Meal assistance robots: A review on current status, challenges and future directions," in *2015 IEEE/SICE international symposium on system integration (SII)*. IEEE, 2015, pp. 211–216.

[219] T. Bhattacharjee, E. K. Gordon, R. Scalise, M. E. Cabrera, A. Caspi, M. Cakmak, and S. S. Srinivasa, "Is more autonomy always better? exploring preferences of users with mobility impairments in robot-assisted feeding," in *Proceedings of the 2020 ACM/IEEE international conference on human-robot interaction*, 2020, pp. 181–190.

[220] Z. Wang, S. Hirai, and S. Kawamura, "Challenges and opportunities in robotic food handling: A review," *Frontiers in Robotics and AI*, vol. 8, 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frobt.2021.789107

[221] G.-N. Zhu, Y. Zeng, Y. S. Teoh, E. Toh, C. Y. Wong, and I.-M. Chen, "A bin-picking benchmark for systematic evaluation of robotic-assisted food handling for line production," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 3, pp. 1778–1788, 2023.

[222] G. Bradski, "The OpenCV Library," *Dr. Dobbs Journal of Software Tools*, 2000.