

DLR-IB-RM-OP-2023-97

**Bayesian Optimization for robust
robotic grasping**

Masterarbeit

Juan García-Lechuz Sierra



DLR

**Deutsches Zentrum
für Luft- und Raumfahrt**

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

**Bayesian Optimization for robust robotic
grasping**

Juan García-Lechuz Sierra

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

**Bayesian Optimization for robust robotic
grasping**

**Bayes'sche Optimierung für robustes
Roboter Greifen**

Author: Juan García-Lechuz Sierra
Supervisor: Prof. Dr. Alin Albu-Schäffer
Advisor: Dr. Maximo A. Roa, Ana Elvira Huezo, Ashok M. Sundaram
Submission Date: 15/09/2023

I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 15/09/2023

Juan García-Lechuz Sierra

Acknowledgments

I would like to express my sincere gratitude to all those who have helped and supported me during the completion of this work.

Thanks to Dr. Rubén Martínez-Cantín for his trust and assistance in starting this research.

I would also like to thank Prof. Dr. Alin Albu-Schäffer and Dr. Maximo A. Roa for giving me the opportunity to do this master's thesis at DLR, accompanied by great colleagues and researchers. I would especially like to thank Ashok M. Sundaram for his advice and Ana Elvira Huezo for her constant help, suggestions, and teachings from day one.

And finally, I would like to thank my family and friends for their vital support in the good and bad moments that a work like this entails.

Abstract

Bayesian Optimization for robust robotic grasping

ABSTRACT

Among the most complex tasks performed by humans is the manipulation of objects. In robotics, automating these tasks has applications in a variety of environments, such as the development of industrial processes or providing assistance to people with physical or motor disabilities. Using bio-inspired robotic hands is helping the emergence of increasingly robust and dexterous grasping strategies. However, the difficulty lies in adapting these strategies to the variety of tasks and objects, which can often be unknown also involving the computational overhead of identifying them and reconfiguring the grasp. The brute-force solution is to learn new grasps by trial and error. This method however is inefficient and ineffective, as it is based on pure randomness. In contrast, Bayesian optimization allows us to turn this process into active learning, where each attempt adds information to the approximation of an optimal grasp, in a manner analogous to a child's learning. The present work aims to test Bayesian optimization in this context, providing some techniques to enhance its performance, and experimenting not only in simulation but also on real robots, as well as studying different grasp metrics that allow the grasp evaluation during the optimization process and how they behave when computing from a real system. For this, along this work, we implemented a realistic simulation environment using PyBullet, which emulates the real experimental environment. This work provides experimental results using the Light Weight robotic arm, designed at the German Aerospace Center (DLR), and two tridactyl robotic hands, the CLASH (DLR) and ReFlex TakkTile (Right Hand Robotic), demonstrating the usefulness of the method for performing unknown object grasping even in the presence of noise and uncertainty inherent in a real-world environment. Consequently, this work contributes with practical knowledge to the studied field and serves as a proof-of-concept for future grasp planning and robotic manipulation technology.

Acronyms

BO Bayesian Optimization

CDF Cumulative density function

CNN Convolutional Neural Network

CP Collision Penalty

CR Contact Reward

DLR German Aerospace Center

EI Expected Improvement algorithm

GPIS Gaussian Process Implicit Surfaces

GPs Gaussian Processes

GWS Grasp Wrench Space

LHS Latin Hypercube Sampling

LN Links and Nodes middleware

LWR Light Weight Robot

MCMC Monte Carlo Markov Chain

PDF Gaussian probability density function

R Reachability index

RL Reinforcement Learning

URDF Unified Robot Description Format

Contents

Acknowledgments	iii
Abstract	iv
1 Introduction	1
1.1 Motivation and goals	1
1.2 Organization of the Master’s Thesis	2
2 Fundamentals	3
2.1 State of the art	3
2.2 Bayesian Optimization	6
2.2.1 Probabilistic model	7
2.2.2 Acquisition function	9
2.3 Grasp Metrics	10
2.3.1 Based on algebraic properties of G : Grasp isotropy index	12
2.3.2 Considering hand configuration: Uniformity of transformation	12
2.3.3 Considering limitations on the finger forces	13
3 Methodology: Active Grasping process	16
3.1 Contributions	16
3.2 Assumptions	17
3.3 Optimization process	19
3.4 Grasping process	21
3.4.1 Reachability check and Inverse Kinematics computation	22
3.4.2 Collision check	24
3.4.3 Motion planning	25
3.4.4 Force closure check	26
3.4.5 Grasp quality measure	28
4 Experimental setup	29
4.1 Software	29
4.1.1 Simulation environment: PyBullet	29
4.1.2 Links and Nodes	31

Contents

4.2	Hardware	32
4.2.1	ReFlex TakkTile Hand	33
4.2.2	CLASH Hand	35
4.2.3	Light Weight Arm	40
4.3	Objects set	42
5	Experimental results	45
5.1	Free-Hand simulation	45
5.2	Full model simulation	46
5.3	Real world experiments	51
5.3.1	Experiment 0: Pipeline test	51
5.3.2	Experiment 1: Simulation vs. real environment	52
5.3.3	Experiment 2: Optimization in real environment	58
6	Conclusion	62
6.1	Limitations and future work	64
	List of Figures	66
	List of Tables	69
	Bibliography	70

1 Introduction

1.1 Motivation and goals

The main motivation of this work is to contribute to the research on the grasping and manipulation of objects by robotic systems. The automation of this kind of tasks can have a significant impact in various fields such as industry and assistive robotics. Several factors, such as the development of increasingly sophisticated bio-inspired robotic hands, or the numerous advances in the development of artificial intelligence, are helping the emergence of increasingly robust and dexterous grasping and manipulation strategies. To enhance the efficiency and adaptability of a grasping method is crucial to make it portable to any object, even with no previous knowledge of its characteristics. However, in most cases, this involves a significant computational cost. A simplistic solution to address the grasping of an unknown object could be to learn new grasps by traditional trial-and-error methods, which can be highly inefficient, requiring a large number of attempts to find a successful grasp and, in most cases, not even finding a solution. In contrast, Bayesian optimization has the capacity to adapt to new objects by leveraging prior knowledge and experience, making it a powerful tool for creating versatile robotic grasping systems. It can significantly reduce the number of trials required to find an optimal grasp by intelligently selecting configurations and learning from each grasp attempt the probabilistic features describing the grasping process and handling the uncertainty inherent in robotic perception and control by modeling it explicitly. This process can be compared to the way a child learns basic tasks from complete ignorance.

The main objective of this work is to test the effectiveness of Bayesian optimization in the grasping of various objects whose characteristics are a priori unknown to the method. To carry out this verification, the experimentation will be performed both in simulation and in a real robotic system, taking into account the uncertainties and technical difficulties introduced when carrying out this task in the real environment. On the other hand, different grasp quality metrics based on algebraic grasping properties, geometric relationships, and forces and how they behave when estimated from the real experimental environment are studied, and used during the Bayesian optimization process. To carry out the experimentation, all the processes involved in the grasp

optimization will be first implemented in simulation, using PyBullet, since it is open source and allows us to emulate with sufficient accuracy the real environment, including the Light Weight robotic arm, designed at the German Aerospace Center, the CLASH (DLR) and the ReFlex TakkTile (Right Hand Robotics), both tridactyl robotic hands with similar characteristics. To perform the communication of the different processes involved in the grasp optimization, it is integrated using the Links and Nodes software framework for programming robotic systems, internally developed at DLR. This work provides a first approach to the application of Bayesian Optimization to the tactile exploration of grasps in a real environment, so that the techniques applied for its operation in an arm-hand system, as well as the conclusions drawn from the experimentation, provide a novel contribution to the state-of-the-art in this study.

1.2 Organization of the Master's Thesis

The document is divided into 6 chapters, in addition to the pages devoted to the lists of figures, tables, and bibliography used in the work, and including the present introduction. Chapter 2 contains a descriptive summary of the theoretical foundations required to understand the project. It details formulation and state-of-the-art in reference to the two bases of this project, which are Bayesian Optimization and grasp quality metrics. Chapter 3 introduces the methodology used to carry out the experimentation, including information on the assumptions taken into account to develop the experimental study, as well as all the processes and technical aspects developed to obtain an effective grasp of an object by means of Bayesian optimization. Chapter 4 provides an introduction to the different parts that make up the experimental setup, from the robotic hand-arm system used to perform the grasping and the software used to control the different processes to the set of objects selected for the experimentation. The results obtained during the experimentation, both in simulation and in the real environment, are shown and analyzed in chapter 5. Finally, chapter 6 outlines the conclusions extracted from the project and the results obtained, pointing out the most promising ideas to approach a possible future work.

2 Fundamentals

The fundamentals chapter serves as the groundwork for the entire thesis, exploring the essential concepts, theories, and principles relevant to the research topic, and providing a comprehensive understanding of the subject matter. Throughout it, information from existing literature is provided to support the study of the research questions.

2.1 State of the art

Early work on robotic grasping was based on deterministic models and predefined object models with programmed grasping sequences. Some examples are the work carried out by M.T Mason in [1] or the method presented by Miller et al. in [2]. These analytic methods construct force-closure grasps for multi-fingered dexterous robotic hands but are constrained by the need to have a precise geometric and physical model of the object to be grasped, which sometimes is not possible. However, with the expansion of machine learning and AI, the focus has shifted to more data-driven and adaptive grasping strategies that enable effective grasps to be obtained by sampling grasp candidates and ranking them according to a specific metric. A complete review of data-driven grasp methods can be found in [3], as well as some insights into the open problems in this field.

Consequently, the development of artificial intelligence has led to the emergence of approaches based on deep learning, such as the Convolutional Neural Network (CNN) used for grasp detection by Redmon et al. in [4] and the Reinforcement Learning (RL) approach presented by Joshi et al. in [5]. The Dex-Net system developed in [6] by Mahler et al. is an example of a deep learning approach using analytic grasp metrics to plan robust grasps. Nonetheless, deep learning approaches are not without drawbacks. These models require a large amount of data to train, such as the Cornell dataset used in [4], are computationally expensive, and often offer poor interpretability, which can complicate their implementation in the real world.

In parallel, model-free approaches like Gaussian Processes (GPs), such as work carried out by Dragiev et al. [7] and Mahler et al. [8] using Gaussian Process Implicit Surfaces (GPIS), have also been explored, resulting in different ways of estimating the

characteristics of the object to be grasped and planning the grasp based on existing grasp experience and heuristics, handling uncertainty and variation in grasping scenarios due to their statistical foundations. Unlike deep learning approaches, they do not require massive datasets or computational resources.

Among these model-free approaches and considering the growing interest in integrating uncertainty into the robotic grasping framework, Bayesian optimization (BO) [9] has emerged as a promising technique with a sequential design strategy for addressing the grasping problem. As pointed out by different authors such as Leco et al. in [10] or Bohg et al. in [3], systematic and random errors are inherent to a robotic system and are a direct consequence of noisy sensors and inaccurate models of the robot's kinematics and dynamics. BO is able to cope with this real-world uncertainty and provide an effective solution to the grasping problem.

This was demonstrated as a first contribution in the work of Nogueira et al. [11]. The paper introduces the unscented transformation as a system to propagate the noise distribution from the input query through all the models and decisions, applying a nonlinear transformation to the probability distribution. By considering input noise, this approach enables the identification of optimal and safety grasps.

The work was later continued by Castanheira et al. in [12], where a novel 3D haptic exploration strategy combining unscented BO with a collision penalty heuristic is proposed. This approach accelerates the convergence of the method by guiding the search algorithm away from potential collision configurations. Collision penalty allowed them to increase the search space to 3D, facilitating the discovery of better grasps without increasing the number of exploration steps.

To address scalability and communication challenges in Bayesian optimization observable in [13], a fully distributed approach was introduced by García-Barcos et al. in [14]. The proposed method enabled high scalability and the addition of new resources on demand while reducing communication between nodes something that can be applied to make the grasp planning process more efficient and flexible.

Another important contribution was made by De Farias et al. in [15]. This paper introduces object shape representation along with tactile exploration into the grasp planning process. They used a probabilistic object shape representation constructed using GPIS combined with Bayesian optimization, achieving stable grasp configurations with a high probability of success.

Recently, last year our contribution with Herrera et al. [16], compared different parameterizations of the hand palm position and orientation and how the number of parameters and their ranges affected the grasp success when combined with BO. On the other hand, extensions of the BO algorithm were designed and evaluated to obtain multiple solutions to obtain robust grasps of the object, even if the object is partially occluded or requires to be grasped by a specific area to be manipulated or perform the task.

As it will be better explained in section 2.2, in order to model grasping as a function to be optimized by BO, the success or failure of grasps must be transformed into a series of real and continuous values, that can be handled by the probabilistic model to be updated. For this reason, grasp ranking relies on quality metrics, based on analytic formulations. These metrics provide a systematic approach to evaluating the efficacy of robotic grasps, enabling the evaluation of stability, robustness, and overall performance.

In the mid-1980s Salisbury et al. included in their kinematic analysis of articulated hands [17] the first ideas on how to define a stable grasp. This work was extended by Li and Sastry in [18] to consider frictional contact forces and demonstrate that stability was a necessary but not sufficient condition to characterize a good grasp. They quantified the overall contribution of forces to the grasp and defined the Grasp Wrench Space (GWS) to analyze its robustness. This concept was later used by Ferrari and Canny in [19] to introduce Force Closure as a minimum condition that any grasp must fulfill. In this same work, the epsilon-metric is proposed, widely used in numerous works up to the present day.

Later works, such as Borst et al. [20] and Liu et al. [21] will provide efficient and accurate methodologies to calculate the GWS, which will be introduced in section 2.2, together with the rest of the definitions. The kinematic and force analysis of the grasp was also exploited by Kim et al. in [22], to describe optimal grasps based on Non-dimensional indexes (grasp metrics) such as the isotropy index.

A review of the metrics named before can be found at [23], where also a classification of such metrics is done regarding the considered algebraic properties or contribution forces. A similar classification is used in the work of León et al. [24], where the results obtained with each metric are also analyzed based on different criteria, in order to establish the minimum number of metrics necessary to evaluate a robotic grasp. The conclusions drawn from this work are constrained to the set conditions of the simulation environment and therefore do not directly apply to the effectiveness in reality of the grasp predictions.

The discussion presented in Goins et al. [25], shows the influence of noise and uncertainty introduced during the robotic grasp analysis, indicating some weaknesses in conventional metrics for predicting grasps using GPs. More recent works such as those carried out by Krug et al. [26] and Rubert et al. [27], show how transferable to real-world grasps are the conventional metrics, using different methods. They concluded that none of these metrics guarantees an acceptable grasp by itself, being necessary their combination.

When working in real-world environments using analytic grasp quality metrics, uncertainty is introduced from different sources. Among these sources, the main ones to take into account as described by de Farias et al. in [15] are the estimation of the contact normals due to mechanical and feedback errors, the error in the achieved contact positions due to unmodeled dynamics or imprecise kinematics of the robotic arm, the friction coefficient μ , which affects the stability of the grasp and is often unknown and the center of mass of the object, which in our case is also unknown. To cope with those uncertainties, several recent methods, such as the vision-based method presented in Morales et al. [28] or the previously mentioned GPIS-based method [15], as well as the active grasping method presented in this thesis, are focused on learning robotic grasps by experience obtained during grasp execution, as it is done using BO.

2.2 Bayesian Optimization

Bayesian Optimization (BO) [9] is an efficient method for tackling global optimization of black-box functions [29], especially functions that are computationally, energetically, or temporally demanding to evaluate. As described in [30], it can be used in many applications such as Automatic Machine Learning and Hyperparameter Tuning, Natural Language Processing, and the one which concerns this thesis, Robotics and RL.

This trial-and-error technique is characterized by its generality, handling complexity, and uncertainty introduced by real-world problems. In the case of robotics and more specifically in robotic grasping, this feature would make BO capable of dealing with diverse objects and end-effectors just with the knowledge from previous trials.

In a formal way, BO finds the optimum (e.g. maximum) of an unknown real-valued function $f : \mathcal{X} \rightarrow \mathbb{R}$ where \mathcal{X} is a compact space, $\mathcal{X} \subset \mathbb{R}^d$, with dimension $d \geq 1$, with a maximum budget of N evaluations of the target function f . The target function f can be freely set according to the task, in our case grasping an object effectively

and safely, and \mathcal{X} is the space of parameters that describe the end-effector pose. The algorithm selects the best query point $x_i \in \mathcal{X}$ to evaluate each iteration with outcome $y_i = f(x_i) + \eta$, where η is zero-mean noise with variance σ_η^2 , finding an optimum x^* so that $|y^* - y_n|$ is minimum for the considered budget.

To find the optimum, BO is composed of two main ingredients. The first one is a probabilistic surrogate model, which is a distribution over the family of functions $P(f)$, where the target function $f()$ belongs, built incrementally sampling over \mathcal{X} . The second one is an acquisition function $\alpha(x, p(f))$ that uses the information from the surrogate model to select the next query point x to evaluate while balancing between exploration and exploitation. This sequential process, shown graphically in 2.1, has been applied to the present work using the BayesOpt library by Martinez-Cantín [31], characterized by using a Gaussian process as a surrogate model as explained later. Its general operation is described in algorithm 1.

Algorithm 1 General algorithm for Bayesian Optimization

Require: N : budget, \mathcal{C} : criterion(Ac.function), $f(.)$: target

Ensure: x^* : optimum

Build a data set \mathcal{D} of points $X = x_1 \dots x_l$ and its response $y = y_1 \dots y_l$ using an initial design.

while $i < N$ **do**

Update the distribution with all data available

Select the point x_i which maximizes the criterion: $x_i = \operatorname{argmax} \mathcal{C}(x | P(f | \mathcal{D}))$.

Observe $y_i = f(x_i)$

Augment the data with the new point and response: $\mathcal{D} \leftarrow \mathcal{D} \cup \{x_i, y_i\}$

$i \leftarrow i + 1$

end while

2.2.1 Probabilistic model

As implemented in BayesOpt [31] we will consider the probabilistic surrogate model as a Gaussian process $GP(x | \mu, \sigma^2, \theta)$ with inputs $x \in \mathcal{X}$, scalar outputs $y \in \mathbb{R}$ and an associated kernel or covariance function k with hyperparameters θ , estimated using a Monte Carlo Markov Chain (MCMC) algorithm. Gaussian processes are well-suited for this kind of task due to several advantageous properties such as being flexible, non-parametric, and great uncertainty predictors.

Given at step n a set of observations $\mathcal{D}_n = (X, y)$, being $X = \{x_{1:n}\}$ the input points

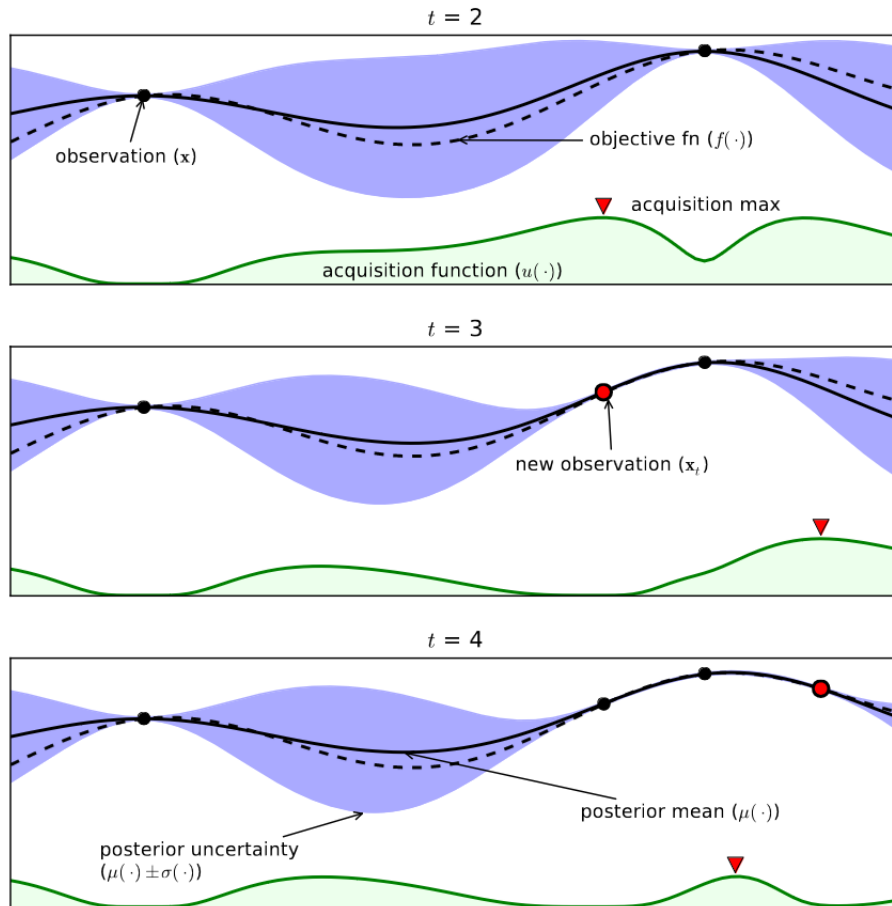


Figure 2.1: An example of using Bayesian optimization on a toy 1D design problem. The figures show a GP approximation of the objective function over four iterations of sampled values of the objective function. The figure also shows the acquisition function in the lower-shaded plots. The acquisition is high where the GP predicts a high objective (exploitation) and where the prediction uncertainty is high (exploration). Note that the area on the far left remains unsampled, as while it has high uncertainty, it is (correctly) predicted to offer little improvement over the highest observation [9].

until that step and $y = \{y_{1:n}\}$ their respective outcomes such that $y_i = f(x_i) + \epsilon$, being $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$ Gaussian noise with zero mean and σ_ϵ^2 variance, then the prediction of the GP $y_{n+1} = \hat{y}(x_{n+1})$ at a new query point x_{n+1} , with kernel k_i conditioned on the i -th hyperparameter sample $k_i = k(\cdot, \cdot | \theta_i)$ is normally distributed and given by 2.1:

$$\hat{y}(x_{n+1}) \sim \sum_{i=1}^m \mathcal{N}(\mu_i, \sigma_i^2 | x_{n+1}) \quad (2.1)$$

where:

$$\mu_i(x_{n+1}) = k_i(x_{n+1}, X) K_i^{-1} y \quad (2.2)$$

$$\sigma_i^2(x_{n+1}) = k_i(x_{n+1}, x_{n+1}) - k_i(x_{n+1}, X) K_i^{-1} k_i(X, x_{n+1}) \quad (2.3)$$

In 2.2 and 2.3 $k_i(x_{n+1}, X)$ is the kernel i evaluated at the query point x_{n+1} with respect to the dataset X and $K_i = K_i(X, X) + I\sigma_\epsilon^2$ is the Gram matrix corresponding to kernel k_i for the dataset X , with noise variance σ_ϵ^2 . The sampling distribution used to estimate θ results in a predictive distribution at any point x as a mixture of Gaussians.

The posterior mean μ_i and variance σ_i^2 evaluated in the objective function at x_{n+1} represent the prediction and uncertainty of the model at this point.

2.2.2 Acquisition function

The acquisition function considers the predictive distribution for each point in \mathcal{X} , whose mean and variance are defined in equations 2.2 and 2.3, to select the next point at each iteration.

The criteria provided by BayesOpt [31] to perform this task is the Expected Improvement algorithm (EI) described in [29] and [30]. The EI is the expectation of the improvement function:

$$I(x) = \max(0, \hat{y}(x) - \rho) \quad (2.4)$$

where ρ is an incumbent value, i.e. the best outcome found until now (step n):

$$\rho = \max(y_{1:n}) \quad (2.5)$$

Taking the expectation over the mixture of Gaussians of the predictive distribution, we can compute the expected improvement as:

$$EI(x) = \mathbb{E}_{(\hat{y} | \mathcal{D}_n, \theta, x)}[\max(0, \hat{y}(x) - \rho)] = \sum_{i=1}^m [(\mu_i(x) - \rho)\Phi(z_i) + \sigma_i(x)\phi(z_i)] \quad (2.6)$$

where ϕ and Φ correspond to the Gaussian probability density function (PDF) and Cumulative density function (CDF) respectively and $z_i = (\rho - \mu_i(x))/\sigma_i(x)$. Without forgetting that the pair (μ_i, σ_i^2) is the prediction computed with equations 2.2 and 2.3.

In order to compute the next point to be evaluated, EI can be maximized by optimizing two main elements of equation 2.6, which are $(\mu_i(x) - \rho)$ and $\sigma_i(x)$. If $(\mu_i(x) - \rho)$ is maximized, the next point will be located where the value is expected to improve the most. On the other hand, if $\sigma_i(x)$ is maximized, the next point should be located where the model has less information about. By maximizing these two elements the algorithm balances the exploration versus exploitation trade-off.

As we can see in algorithm 1, finding an optimal point x_i requires evaluating the acquisition function multiple times. One of the most important advantages of using BayesOpt [31] is that it precomputes certain terms of the posterior distribution and acquisition functions that are independent of the specific query point x . Other works such as [14] attempt to improve the efficiency of the BO method by reducing its sequential nature.

In the case of BayesOpt, an initial set of points \mathcal{D} is selected using Latin Hypercube Sampling (LHS) by default, since EI has been shown not to be stable during the first steps of the optimization due to the shortage of information [29]. Also, a well-informed prior and a high-quality posterior distribution allow the surrogate model to accurately represent the objective function, with a more efficient exploration of the search space. This can reduce considerably the number of evaluations needed to achieve the desired optimization performance.

2.3 Grasp Metrics

Grasp quality metrics play a crucial role in robotic grasping, serving as useful evaluative tools for assessing the effectiveness and stability of robotic grasps. As will be demonstrated later, they can be used not only as an index of performance but also for optimizing grasps during execution. However, to be able to execute the planned grasps in reality additional conditions need to be considered or analyzed, since most of the analytic grasp metrics do not consider the complexity and the uncertainties in a real environment. In this scope, using sensor feedback to evaluate and optimize grasps can significantly reduce grasping failures.

In order to compute metrics we must consider contacts between the fingers and

the object as punctual contacts with friction (hard contacts), and forces acting only against the object. Theoretically, as described in [23], a force F_i applied on the object at a point p_i generates a torque $\tau_i = p_i \times F_i$ with respect to the object's center of mass. It should be noted that since unknown objects are considered in this work, as will be explained in the next chapter, the center of mass of the object has been approximated to the center of its bounding box. On the other hand, the most common way of considering friction is using Coulomb's friction cone. It represents the set of all possible frictional force vectors that can be exerted by the contact point without causing slippage.

In the case of 3D objects, forces and torques in the x, y , and z directions can be grouped in a six-dimensional wrench vector $\omega = (F_x, F_y, F_z, \tau_x, \tau_y, \tau_z)^T$. The force f at the fingertips and the total wrench ω applied on the object can be related using the grasp matrix G , as well as velocities v at the contact points and the twist \dot{x} :

$$\omega = Gf \quad (2.7)$$

$$v = G^T \dot{x} \quad (2.8)$$

Additionally, forces at the fingertips and torques at the finger joints T are related through the hand Jacobian J_h , as well as v and velocities $\dot{\theta}$ at the finger joints:

$$v = J_h \dot{\theta} \quad (2.9)$$

$$T = J_h^T f \quad (2.10)$$

From equations 2.8 and 2.9, the relation between the velocities of the finger joints and the velocities of the object can be obtained:

$$J_h \dot{\theta} = G^T \dot{x} \quad (2.11)$$

From the previous equation we can define the transformation in the velocity domain from the higher dimensional hand joint space to the lower dimensional object space using the hand-object Jacobian matrix H :

$$\dot{x} = H \dot{\theta} \quad (2.12)$$

where $H = (G^T) + J_h$

As concluded in [23], a proper grasp should be optimal with respect to two groups of quality metrics, the first considers the locations of the contact points on the object and directions and magnitudes of the forces applied at the contact points, and the other that considers the gripper configuration. The quality measures studied in this work

were selected following the correlation analysis conducted in [32] and [24] and will be explained in more detail below.

2.3.1 Based on algebraic properties of G : Grasp isotropy index

In the case of 3D objects, the matrix G presented in Eq. 2.7 is of the form $G \in \mathbb{R}^{6 \times m}$, being m the number of forces applied at the fingertips. Consequently, a full-rank G has 6 singular values computed from the positive square roots of the eigenvalues of GG^T . At least one of the singular values becomes zero when performing a singular grasp configuration, indicating that the grasp is unable to exert forces in at least one direction. As described in [18], this property can be exploited to indicate how far a grasp is from falling into a singular configuration, considering the minimum singular value of G , $\sigma_{min}(G)$. However, this singular value depends to a large extent on where the torque origin is located.

Another way of considering singular grasp configurations is the Grasp Isotropy Index [23][22]:

$$Q_{iso} = \frac{\sigma_{min}(G)}{\sigma_{max}(G)} \quad (2.13)$$

where $\sigma_{max}(G)$ corresponds to the maximum singular value of G .

Apart from indicating the closeness to a singular configuration, the similarity between the maximum and minimum values indicates an equivalent grasp behavior in all directions. As can be deduced, this index obtains values between 0 and 1, approaching 1 when the grip is more isotropic.

It is necessary to emphasize that although this index indicates the capacity to resist forces in all directions, it does not indicate anything about the amount of force that can be resisted, so it is not a measure of robustness by itself.

2.3.2 Considering hand configuration: Uniformity of transformation

As occurred with matrix G , in a singular grasp configuration the hand-object Jacobian $H \in \mathbb{R}^{6 \times n}$ (being n the number of finger joints) has one of its singular values equal to zero. As a way to measure if the contribution of joint velocities is the same in all the components of the object velocity the condition number of the Jacobian matrix, introduced in [17], is computed. In order to achieve a metric that corresponds to a more

uniform transformation the higher its value is, the relation can be inverted as in [24] and defined as:

$$Q_{uni} = \frac{\sigma_{min}(H)}{\sigma_{max}(H)} \quad (2.14)$$

where $\sigma_{max}(H)$ and $\sigma_{min}(H)$ correspond to the maximum and minimum singular values of H .

When the inverse of the condition number is high the hand can be considered to have a better manipulation ability and control.

2.3.3 Considering limitations on the finger forces

Robotic grippers, as well as a human hand, have finite force capabilities, and exceeding these limits can lead to unstable or unsuccessful grasps. Therefore, quality measures that, unlike the one explained above, account for these force limitations are essential for describing the robustness of a grasp. In order to do so, these qualities consider the module of the perturbation wrench that a grasp is able to resist.

The Grasp Wrench Space (GWS) [18][19] [20] describes the set of external wrenches that can be resisted by a grasp. To obtain those wrenches, the friction cone at the contact point can be approximated with a polygonal pyramid of m edges, as described in Liu et al. [21], so that the force f_i applied by the fingertip i can be expressed as a positive linear combination of the unitary forces f_{ij} , with $j = 1, \dots, m$ denoting the j^{th} primitive force of the friction cone. Similarly, the primitive wrench ω_i can be expressed as a positive linear combination of the wrenches ω_{ij} produced by f_{ij} . Finally, the resultant wrench that can be produced by the contact forces is written as:

$$\omega = \sum_{i=1}^n \omega_i = \sum_{i=1}^n \sum_{j=1}^m \alpha_{ij} \omega_{ij} \quad (2.15)$$

with $\alpha_{ij} \geq 0$ and $\sum_{j=1}^m \alpha_{ij} \leq 1$.

As described in [19], one way of constructing the set \mathcal{P} of possible resultant wrenches is limiting the sum of modules of the contact forces, considering a limited common power source for all the fingers. For a normalized limit of 1, $\sum_{i=1}^n \|f_i\| \leq 1$, and $\sum_{i=1}^n \sum_{j=1}^m \alpha_{ij} \leq 1$. Consequently, the GWS \mathcal{P} can be computed as:

$$\mathcal{P} = CH(\cup_{i=1}^n \{\omega_{i1}, \dots, \omega_{im}\}) \quad (2.16)$$

where CH denotes the convex hull operation over the union of primitive wrenches.

If the GWS is convex and contains the origin (zero wrenches), then the grasp is said to achieve force closure, which means that it is possible to apply wrenches at the contact points such that any external wrench can be balanced [19].

Largest-minimum resisted wrench

The largest-minimum resisted wrench, Q_ϵ , is one of the most used grasp quality measures. It indicates the magnitude of the maximum perturbation wrench that a grasp can counterbalance in any direction [33].

Its computation was first introduced in [19] as the distance from the origin to the closest facet of \mathcal{P} or, its geometrical equivalence, the radius of the largest ball centered at the origin of the wrench space and fully contained in \mathcal{P} , which is represented in Fig. 2.2.

$$Q_\epsilon = \min_{\omega \in \partial\mathcal{P}} \|\omega\| \quad (2.17)$$

where $\partial\mathcal{P}$ is the boundary of \mathcal{P} .

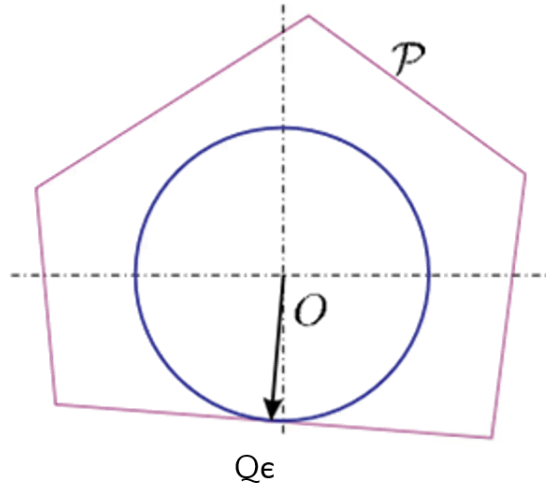


Figure 2.2: Radius of the largest ball centered at the origin of the wrench space and fully contained in \mathcal{P} (Epsilon measure, adapted from [23])

Although Q_ϵ is a great indicator of robustness for general-purpose grasp, it depends on the origin of the reference selected to compute torques. When the object's center of mass is not known the selected origin can significantly change its accuracy.

Volume of the Grasp Wrench space

To overcome the dependence that some metrics as Q_ϵ and Q_{iso} have on the origin of torques, the volume of \mathcal{P} is proposed by Li and Sastry in [18] as a quality measure. This measure is independent of the reference system used to compute torques and indicates the bound of external disturbances that a given grasp can resist. However, it says nothing about the capacity to handle perturbations in specific directions.

$$Q_v = volume(\mathcal{P}) \quad (2.18)$$

An interesting aspect of this measure is that it theoretically implies that the grasp is more robust to uncertainties, such as object pose estimation errors or variations in the object's physical properties. A larger volume indicates that the robot is better able to compensate for these uncertainties by exerting forces and torques within the GWS. This characteristic has been proven in the experiments carried out at [24].

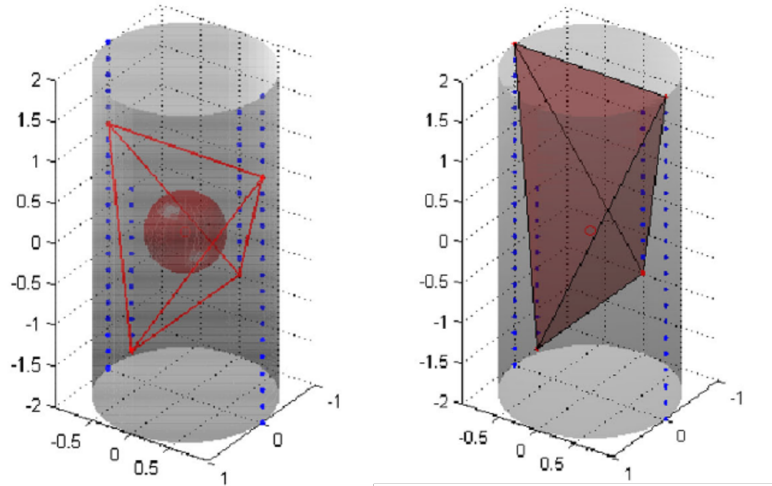


Figure 2.3: Comparison between the Largest minimum resisted wrench Q_ϵ (left) and the Volume of the GWS Q_v (right) metrics of a grasp with 4 contacts [23].

3 Methodology: Active Grasping process

The next chapter presents a comprehensive overview of the developed techniques and procedures employed in this study. It outlines the step-by-step approach used to address the research objectives and provides a detailed account of the research methods, as well as all the assumptions made.

3.1 Contributions

In this section, we outline the key contributions of BO compared to other data-driven grasp approaches and the contribution of this thesis to the existing body of knowledge in the use of Bayesian optimization for robust robotic grasping. The following contributions have been studied to advance the state of the art, both theoretically and practically, with potential implications for future research and real-world applications.

- Contribution of BO to the development of data-driven grasp methods:
 - **Is a sample-efficient optimization algorithm.** This means that it can find the optimal solution with fewer samples than other optimization methods, which is particularly important in robotics where the cost of sampling can be high.
 - **Addresses uncertainty in sensory measurements and object properties,** allowing the methodology to make grasp decisions while considering the reliability of information in real-world environments, a feature lacking in many other deterministic or rule-based approaches.
 - **Provides a framework for continuous learning and adaptation.** It can refine grasping strategies over time by updating its probabilistic models based on feedback from previous grasping attempts. This self-improvement differentiates it from static rule-based methods.
 - **BO method adapts to a wide range of object shapes, sizes, and materials.** It handles uncertainties and can efficiently explore the grasp space, making it well-suited for handling diverse and previously unseen objects, surpassing the limitations of traditional heuristic-based approaches.

- Specific contribution of this thesis to the field of study:
 - **Implementation of a complete pipeline for utilizing Bayesian optimization in an arm-hand system**, with portability to any middleware for robotic system integration.
 - **Development of new heuristics** in the grasp evaluation function that allows the application of the method to a real environment and improves its convergence. The improvement in performance with respect to state of the art has been verified in the experimentation of chapter 5.
 - The method **combines analytic and data-driven grasping approaches** to obtain an optimal result, addressing the uncertainty of traditional metrics when computed from real sensors and robots. The performance of this combination is proven in the experimental chapter as well.
 - The development of this method has allowed the **experimentation in a real environment** to provide conclusions and knowledge to the state-of-the-art in this subject.

The work done to secure these contributions is described in detail in the following sections.

3.2 Assumptions

Throughout the work, a series of assumptions are considered to facilitate the study of grasps, the obtaining of data, and the approximation of the simulation to the real environment.

An example of this is found in the way grasp qualities are calculated. Each one of the metrics explained in section 2.3 considers an approximately static state at the moment of grasping, neglecting any dynamic disturbance that may affect it. In addition, a way to facilitate the normalization of the results, the adaptability of the study to the available hardware, and the generalization to different objects, is to consider the normal forces exerted at the point of contact as unitary. In this way, the analysis can focus more on the geometric aspects of the grasp, such as the distribution of contact points, the shape of the grasp wrench space, and the overall stability of the grasp configuration. In any case, it is necessary to emphasize that some metrics such as Q_e can lose part of their meaning when considering unitary forces.

But the most critical consideration to take into account is the amount of information that is known about the object to be grasped. In the work done by Bohg et al in [3],

we can find a detailed classification of grasping methods according to the available knowledge about the object. Following this classification, an object can be **known** if there is access to a database with an accurate geometric object model from which to compute a set of predefined good grasps. When the object shares some similarity with previously known objects, for example in the shape or color, it is considered **familiar** and the grasps are computed with configurations similar to those chosen for known objects. Finally, the object can be considered **unknown** if there is no access to the object model or previous grasp experience and the necessary information must be obtained by means of sensory data.

This work aims to evaluate the performance of BO when dealing with unknown objects. Therefore, all the object information used to carry out the grasping exploration is assumed to be possible to obtain using RGB-D sensors or another kind of image and depth sensors, from which we would be able to get the object position, and a bounding box to search for a grasp pose, as represented in Fig. 3.1. The model represented in this figure corresponds to the one used in the simulation to obtain collision data. However, this model is not necessarily useful as there are other methods to prevent collisions and the bounding box can be estimated in a real environment from a single point of view.

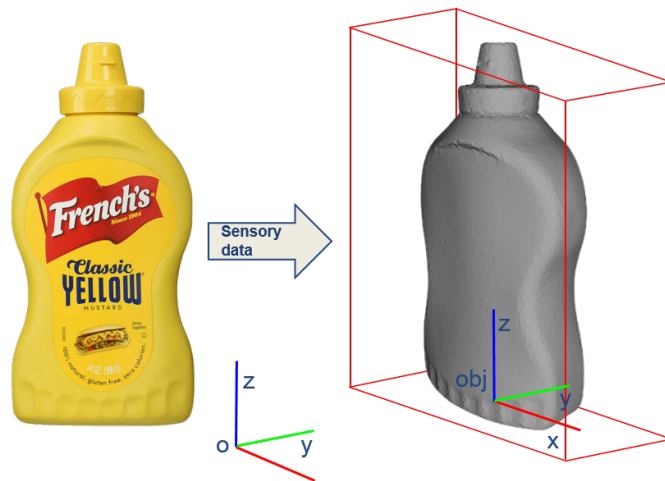


Figure 3.1: Graphical representation of the search space (bounding box).

On the other hand, and given that this work does not focus on the perception methodology to be used, in the realization of the experiments it is considered that the object is static at all times. This characteristic does not correspond to reality and can be considered a limitation of this method. However, it facilitates the study of the same

in this first approximation and including more realistic scenarios can be approached in future work. It is also worth mentioning that other works such as the one carried out by de Farias et al. in [15] offer solutions to obtain online information on unknown objects, for example using tactile feedback, based on the same methodology.

3.3 Optimization process

In the previous work mentioned in state-of-the-art (Herrera et al. [16]), the experiments showed how better results were obtained for less number of optimization parameters, computing orientation of the grasp as the direction from the grasp position to the grasp origin, considering the origin the center of the object. Following this work we have chosen as optimization parameters the position in Cartesian coordinates and an offset angle in roll, with respect to the orientation computed from the grasp position.

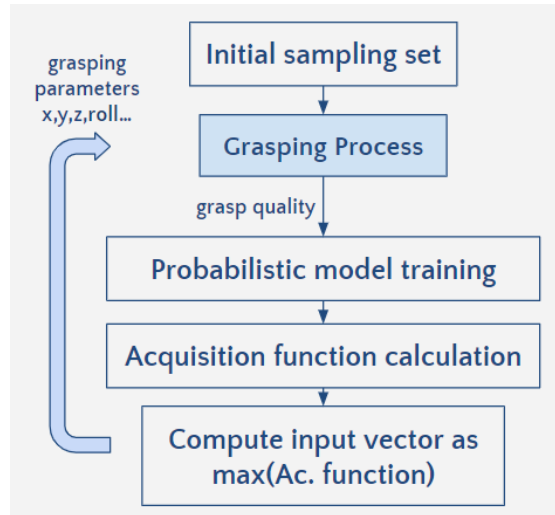


Figure 3.2: Optimization process scheme. Integration of the grasping process in each iteration.

To further understand the iterative process that takes place when performing BO, we can look at Fig. 3.2. In the first step of the optimization, the grasping process is fed with an initial set of samples of the grasp pose selected using LHS as mentioned in section 2.2, which number can be configured using the "n_init_samples" parameter of BayesOpt. As described in the previous section 3.2, These initial samples and the grasp poses calculated at each iteration are restricted to lie within the search space defined for each object. For the selected set of samples, the grasping process takes place, updating

the probabilistic model and the acquisition function with the grasp quality obtained, and in every iteration, the pose of the hand that maximizes the acquisition function is computed. Each time a new target pose is received, the hand is placed in the new location, and a new evaluation process is carried out.

Regarding the search space, the limits of the bounding box defined for each object are computed according to the following equation, also represented in Fig. 3.3:

$$\begin{aligned} upper_bound_i &= \frac{s_i}{2} + l_{proximal} \\ upper_bound_z &= s_z + l_{proximal} \end{aligned} \quad (3.1)$$

where s is the size of the object on the specific coordinate axis, $l_{proximal}$ is the proximal length of the robotic hand used, and $i = \{x, y\}$.

This equation is the result of part of the first experimental trials, in which considerable improvement is observed when changing the grasp pose reference from the wrist, or end-effector, to the palm of the hand. This is because the choice varies considerably in the size of the bounding box, which in the case of placing the reference on the wrist can result in positions too far away from the object at critical points where there is some displacement in more than one reference axis.

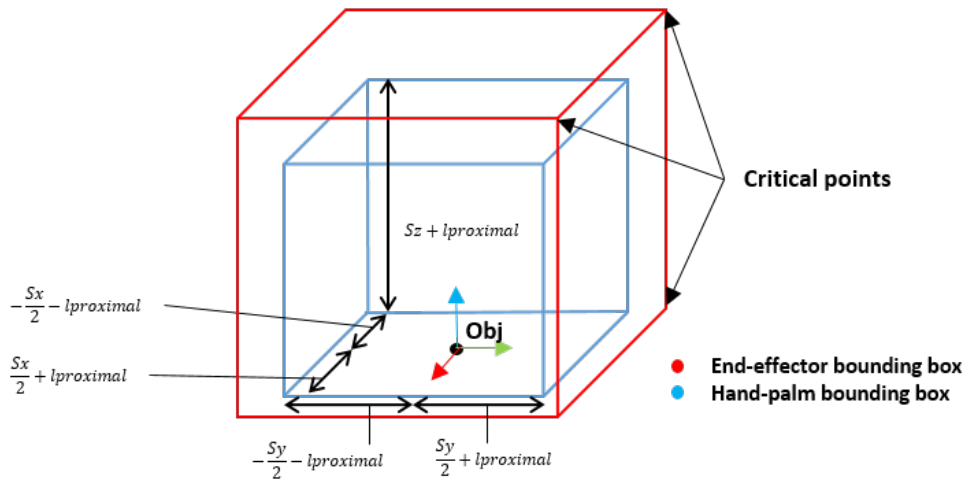


Figure 3.3: Bounding box limits. Difference between considering the end-effector pose and the hand-palm pose.

The lower bounds in X and Y are computed as the negative of the upper bounds using as the origin of the bounding box the center point of the base of the object. The lower bound in Z is 0, and the orientation is optimized using an offset angle in roll between 0 and $\frac{\pi}{2}$.

Although it makes the learning process more expensive, kernel parameters are learned every iteration, using MCMC for a more robust and accurate method. This can be configured when using BayesOpt by setting the "n_iter_relearn" parameter to 1.

3.4 Grasping process

In the previous work developed with Herrera et al. [16], the grasping process simulation only took into account a simple environment in which there is only the object to be grasped and the hand can move freely within the workspace (setup shown in Fig. 3.4.A). To bring this method to a real context, the first step is to include the arm and the workbench in the simulation, as well as all the processes that allow the robot to move safely through the environment and that allow the method to be transferred to the real world (setups in Figs. 3.4.B and 3.4.C). All these processes will be explained below.

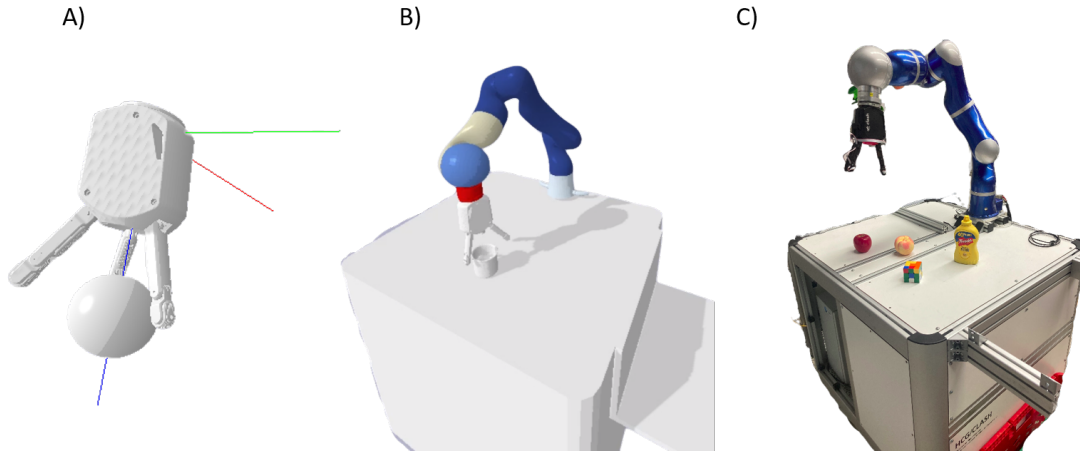


Figure 3.4: Experimental setups: A) Hand in free space; B) Full robot model simulation; C) Real environment.

3.4.1 Reachability check and Inverse Kinematics computation

Although the exploration space is bounded, as explained in section 3.3, within these limits the grasping poses computed by BayesOpt are totally unconstrained. This means, that some hand-palm poses and robot configurations can be unfeasible for different reasons.

As a first step to evaluate the grasping poses provided by the acquisition function of the BO, the reachability of such pose is checked by comparing it with a capability map generated by the software developed by Porges in [34], from the same kinematic model of the robotic arm used to carry out the simulation. A capability map (Fig. 3.5) is a representation of the 6D space (position and orientation) of the regions where the tool frame is able to move. Each cell of the capability map has an associated Reachability index (R), which numerically represents the dexterity and manipulability of the robot when in that location.

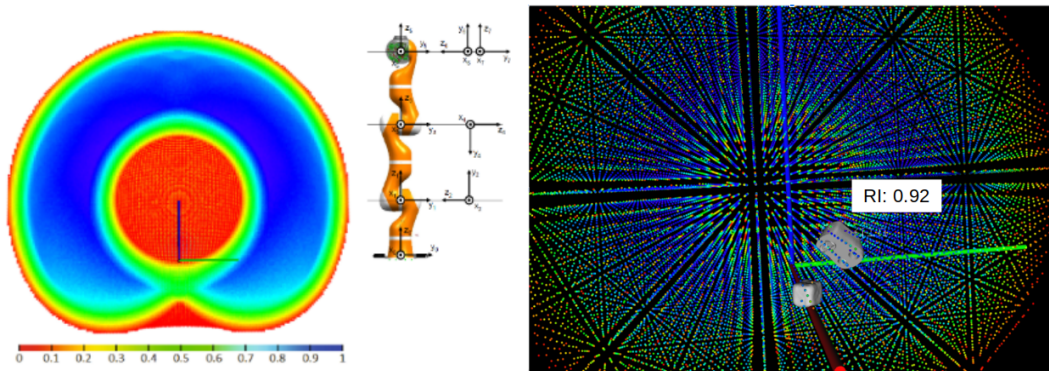


Figure 3.5: Left: Capability map for the Kuka LBR arm, with similar kinematics to DLR's LWR [34]; Right: 3D representation of the hand palm (dark gray) in a reachable position ($R: 0.92$) inside the capability map

Once the first check has been passed, the configuration of the robot joints is obtained in order to reach the desired position. For this purpose, an iterative computation of the inverse kinematics has been implemented to obtain a more accurate result with an error in the position and orientation of the end-effector lower than some established thresholds. In each iteration, the inverse kinematics is solved using the Selectively Damped Least Squares (SDLS) method, presented by Buss and Kim in [35]. Although

this method is a bit slower than the original (DLS), it has proven to be more effective in reaching the desired end-effector position and returning positions as close as possible when the target position is not reachable, which favors our iterative refinement. Next, the resulting configuration is simulated, obtaining the position and orientation of the end-effector with respect to the world frame and comparing it with the target position and orientation. To compare the positions, we obtain the Euclidean distance d_E between them, while to compare the orientations, expressed in quaternions, we calculate the geodesic distance Φ_g between them, as described by Q. Huynh in [36], that indicates the amount of rotation required to bring one rotation aligned to another. The thresholds used were $d_E^t = 0.01$ and $\Phi_g^t = 0.01$. As it is an iterative process, it allows establishing a maximum number of iterations after which the pose of the end-effector is considered unreachable. This second check allows us to discard those unreachable poses due to the configuration of the joints from which the arm motion starts. The resulting method is described in algorithm 2.

Algorithm 2 Accurate Inverse Kinematics method

Require: pos : target position, ori : target orientation, d_E^t : euclidean distance threshold, Φ_g^t : geodesic distance threshold, n_{it} : number of iterations

Ensure: q : joint configuration, $reachable$: boolean reachability

```

while  $reachable$  not True and  $i \leq n_{it}$  do
     $q = calculateInverseKinematics(pos, ori)$ 
    Check  $q$  is inside joint limits
    Simulate  $q$  and obtain  $newPos$  and  $newOri$  from simulation
    Compute  $d_E(pos, newPos)$  and  $\Phi_g(ori, newOri)$ 
    if  $d_E < d_E^t$  and  $\Phi_g < \Phi_g^t$  then
         $reachable = \text{True}$ 
    end if
    if  $i == n_{it}$  then
         $reachable = \text{False}$ 
         $q = \text{None}$ 
    end if
end while

```

3.4.2 Collision check

Another type of unfeasible grasping pose occurs when the robot collides with either the object or the workbench since self-collisions are dealt with by the reachability check. In the case that any of the robot links is in collision with the workbench, the proposed grasp pose is classified as invalid and the result returned by the Grasping Process, which, as already explained, acts as the function to be optimized, would be 0, moving the acquisition function away from poses similar to this one. On the other hand, to handle collisions between the hand and the object, the approach selected by Castanheira et al. in [12] is used. In this approach, a Collision Penalty (CP) is added, taking into account the level of penetration of the hand into the object, directing the acquisition function to poses closer to it, since it differs them from collisions with the workbench or poses that do not even touch the object, reducing the regions that are worth exploring. It should be noted that the collision penalty has been modified with respect to that used by Castanheira et al., indicated in the Eq. 3.2, where $n_j \in \mathbb{N}$ corresponds to the number of joints of the robot's hand that are in collision with the object and λ is a tuning parameter used to smooth the penalty.

$$CP_{Castanheira}(n_j) = 1 - e^{-\lambda n_j} \quad (3.2)$$

This penalty assigns a value between 0 and 1 to collisions. The closer to 1 the greater the penetration of the hand into the object, i.e. the greater the number of hand joints in collision with the object. However, the intention is to bring the optimization closer to the object, while avoiding those poses in which the hand collides with it. Therefore, the CP has been redefined in such a way that it assigns a higher value the lower the level of penetration of the hand into the object, resulting in the following expression:

$$\begin{aligned} CP(n_j) &= \alpha e^{-\lambda n_j} \\ CP(n_j = 0) &= 0 \end{aligned} \quad (3.3)$$

The α coefficient has been added to regulate the range of values between which CP moves, depending on the magnitude of the values obtained when calculating the grasp quality metrics. In this way, a higher result is never given for a pose with collision than for a pose in which the object becomes grasped. Since, as indicated in the mentioned paper, CP is a heuristic that only improves the convergence of the method, but does not intervene in the final evaluation, when there are no collisions between the hand and the object and $n_j = 0$, CP is also 0. The tuning parameter has been set to $\lambda = 0.1$ as indicated in the paper and a coefficient $\alpha = 0.1$ has been used, so that the CP values remain between 0 and 0.1, being substantially lower than those provided by the grasp

quality metrics, which have been normalized between 0 and 1.

3.4.3 Motion planning

In this section, it is necessary to distinguish between two types of processes. The first one is the one in which the grasping process is completely simulated and will be used in the first part of the experiments. In the second one, once the reachability of the grasping pose has been verified and there are no collisions in simulation, the robot movement and the grasping of the object in the real environment are executed. This is the kind of process used in the second part of the experimentation. When grasping is performed in simulation, the arm's motion is non-existent, as it is replaced by the instantaneous update of the arm's joint configuration that is necessary to reach the target pose. In this way, each of the iterations in which the object is grasped becomes much faster than it would be in the real environment.

On the other hand, one restriction of performing movements on the real arm is that most grasping poses result in collisions when the hand approaches the target directly. To solve this type of collision it is necessary to establish a pre-grasp position, from which the hand approaches the object in a controlled manner without collisions with the fingers. The pre-grasp position used is -8 cm away from the target pose, in the Z-axis direction of the tool reference. An example can be seen in Fig. 3.6.

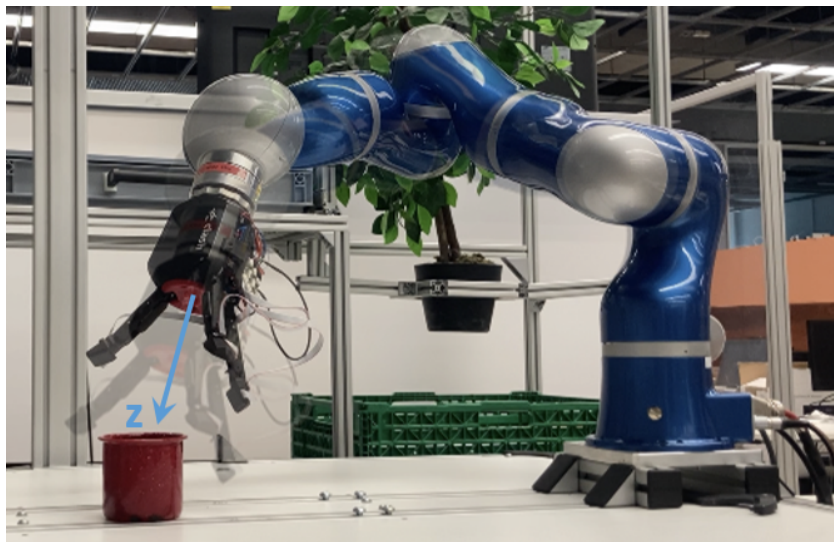


Figure 3.6: Example of arm motion from pre-grasp pose to target pose.

To reach this position, the joint configuration is computed using inverse kinematics, considering as initial position the one from which the movement starts (home), and choosing among the possible solutions the one closest to the target configuration. This consideration is important, especially in a redundant system, because choosing a robot configuration with joint configurations far from the target configuration results in aggressive and unpredictable movements from the pre-grasp pose to the target pose, which can damage the robot. Once the pre-shape configuration is computed, the arm is moved in the joint space to it, and then the target configuration is reached, always in the joint space. Then, the hand is closed using constant velocity until a specific torque is reached.

3.4.4 Force closure check

Once the hand is closed, a first evaluation of the capacity of the grasp to hold the object is made by checking the existence or not of force closure. For this purpose, once the information at the contact point is obtained, the GWS defined in section 2.3 is computed, using the efficient Incremental Convex Hull Calculation implemented by Sundaram et al. in [33], from the one presented by Borst et al. in [20]. Since the known information of the object is limited to its bounding box, its center of mass is approximated to the central point of its bounding box, in order to calculate the torques performed during the grasp. On the other hand, the friction cone has been approximated using 5 as the discretization factor.

While performing the first experiments, the original implementation presented in section 2.3 of the Force Closure calculation, proved not to be a sufficient condition to discard visibly weak grasps, which resulted in spurious data of grasping qualities due to the shape of the object. To solve this problem the force closure check was reimplemented by calculating the distance between the origin and the nearest point of the convex hull, d_{min} , so that if this distance is greater than a certain threshold, it can be considered a strong force closure. This reimplementaion is described in algorithm 3 and represented in Fig. 3.7.

In the previous work, not meeting the Force Closure condition implied an outcome equal to 0 of the grasping function to feed the BO. However, taking as inspiration the CP explained in section 3.4.2, a Contact Reward (CR) has been implemented to improve the convergence of the method. This CR is defined in Eq. 3.4, where n_c corresponds to the number of contact points at the hand's fingertips.

$$CR(n_c) = \alpha(1 - e^{-\lambda n_c}) \quad (3.4)$$

Algorithm 3 Force closure condition

Require: $wrenches$: grasp wrenches, th : distance threshold

Ensure: $isForceClosure$: True/False

$GWS = ComputeConvexHull(wrenches)$

d_{min} = distance from origin to the closest point in GWS

if origin inside GWS and $d_{min} > th$ **then**

$isForceClosure = True$

else

$isForceClosure = False$

end if

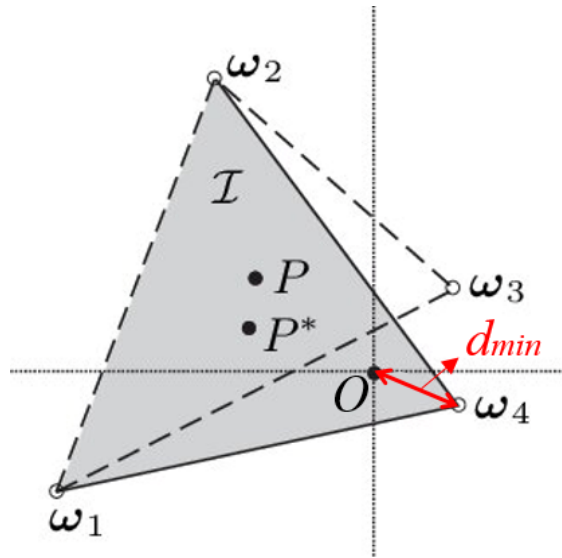


Figure 3.7: Force closure condition adapted from [37]. The wrench set $W = \{\omega_1, \omega_2, \omega_3\}$ (CH represented in discontinuous lines) is non-FC. The wrench set $W^* = \{\omega_1, \omega_2, \omega_4\}$ (CH represented in continuous lines) is a FC grasp with d_{min} represented in red.

In this way, the acquisition function is directed towards those positions where there is a higher number of contacts of the hand fingertips with the object. In Eq. 3.4, α and λ have the same meaning as in Eq. 3.3, and the value considered for them is exactly the same.

3.4.5 Grasp quality measure

Finally, when an iteration results in a grasp with force closure, the grasp quality metrics explained in section 2.3 are computed. To make these metrics comparable with each other they have all been normalized concerning their corresponding maximum values, obtained in the first steps of the experimentation as detailed in the next chapter. In this way, all of them will take values between 0 and 1.

As previously indicated in section 2.1, several works [23][24][32] have concluded in the use of a combined metric that gives rise to an optimal grasp concerning multiple considerations such as the contact points or the gripper configuration. In the second part of the experimentation, tests were carried out both in simulation and on the real robot, to assign weights to each of the metrics studied, in order to obtain a combined metric Q_m resulting from the arithmetic sum of the previous ones:

$$Q_m = w_1 * Q_{iso} + w_2 * Q_{uni} + w_3 * Q_{\epsilon} + w_4 * Q_v \quad (3.5)$$

From the steps explained above one can infer the final expression used as an outcome to feed each iteration of the BO, in other words, the function to be optimized, which is of the form:

$$Q_{grasp} = Q_r * (Q_m + CP + CR) \quad (3.6)$$

where Q_r is a binary index indicating whether the grasping pose is reachable ($Q_r = 1$) or not ($Q_r = 0$). It can also be observed how when the obtained grasp is able to grasp the object (force closure condition), both CP and CR are equal to 0, and the grasp is evaluated only by the grasp quality metric.

4 Experimental setup

4.1 Software

In the following section, there is an introduction to the different software components used both in the simulation and the implementation of a functional robotic system.

4.1.1 Simulation environment: PyBullet

PyBullet [38] is an open-source physics simulation engine built on top of the Bullet Physics Library. It offers a wide range of capabilities for simulating rigid body dynamics, collision detection, and contact resolution. It also provides an intuitive and user-friendly Python interface, enabling researchers and developers to integrate it into their robotics projects easily.

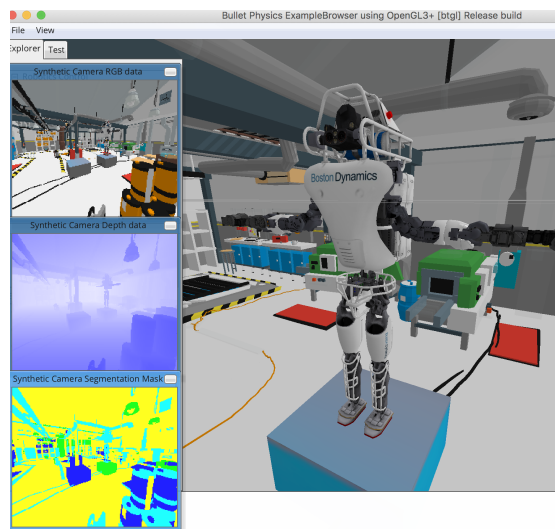


Figure 4.1: PyBullet example of a complex simulation environment [39].

One of the key advantages of PyBullet is its high degree of flexibility, supporting various robot models, including articulated robotic arms and grippers, that can be pro-

vided to the simulation using Unified Robot Description Format (URDF) files. PyBullet allows us to create and control the dynamics of these robotic systems, define their physical properties, and simulate their interactions with objects and environments. This allows us to implement complex grasping algorithms to grasp and manipulate objects with precision. Any object can be added to the simulation as long as you have a URDF model containing its physical characteristics. An example of a complex environment simulated with PyBullet can be seen in Fig. 4.1.

In addition to creating a realistic and accurate simulation environment for studying robotic grasping, PyBullet provides different tools to evaluate our robotic grasping algorithm and perform different calculations during the simulation. It incorporates advanced collision and proximity detection algorithms and contact resolution methods, ensuring realistic interactions between the robotic hand and the grasped objects, and describing the forces and wrenches exerted on the objects during grasping.

During each BO iteration, the collision between the robot, the workbench, and the object has been detected using the PyBullet method *getClosestPoints*, which allows us to detect those points of a given body (the robot) that are at a given distance from another body (the workbench and the object). The safety distance chosen to detect collisions with the table has been 2cm, which is the same value used in the real robot motion planner. To detect collisions with the object, a distance of 1mm has been used, so as not to affect grasping positions very close to the object. All the analysis of the forces involved in the grasp to compute the force closure or the quality metrics in simulation has been implemented using the PyBullet method *getContactPoints* that allows us to obtain the information regarding the position and forces corresponding to the different points of contact between two simulated bodies.

Another important part of executing the grasping process in simulation is related to the robot's kinematics and dynamics. The accurate inverse kinematics method described in section 3.4.1 has been implemented using the *calculateInverseKinematics()* method of PyBullet. Once a new configuration has been computed, the update of the position of the robot joints has been executed by iterating over each of them with the method *resetJointState*. In the last part of the grasping process, the closing motion of the hand was implemented moving the hand joints simultaneously at a constant speed, in this case, 1rad/s, using the method *setJointMotorControl2()*, until a given motor force or time limit is reached.

Other algorithms to compute the forward kinematics of the robot are available in this library, as well as methods to perform transformations between reference frames.

4.1.2 Links and Nodes

As defined by F.Schmidt in [40], Links and Nodes middleware (LN) is a middleware to create and manage flexible distributed real-time systems, specifically designed to develop and control embedded robotic systems. Within this work, it has been used as a link between the processes that perform the BO calculations, the processes that perform the simulation in the PyBullet environment, and the processes that require the use of the real robots presented in section 4.2.

Functions

During the realization of this project, all the LN functions have been implemented using Python3. However, LN also supports other programming languages such as C++.

We can distinguish between two main functions performed by LN:

- **Process management:** This allows us to control each process of our robotic system, and allows the user to interact with it, taking care of the different aspects explained below:
 - **Process dependencies:** It supports the flexible development of robotic applications by handling dependencies between different processes, and restarting modified dependencies when needed.
 - **Priorities:** It takes care of the order in which processes tasks are executed, to optimize the way in which the CPU works on them.
 - **Nodes:** A robotic system comprises several nodes sharing information to perform complex executions. The different processing units could be different computers with different CPUs and OS. The assignment of processes to the different processing units is regulated by Process Management.
- **Communication:** LN also provides an easy and robust way of exchanging data across the different processes even supporting real-time operations. The two main ways of exchanging information in LN are described below.
 - **Topics:** They are information channels to which a node can publish data and update it regularly or subscribe, receiving information from it. The data is attached to a name, allowing subscribers to access data by referring to it. There can be multiple concurrent subscribers to the same topic, but only one node can create and write in it. This communication is done asynchronously and in a unidirectional way.

- **Services:** Services implement a form of request and remote response procedure, in which the call runs code from another node in the system, by transmitting the function arguments, then runs the computation, and transmits the result back, resulting in a synchronous transaction between nodes.

Application components

The following lines list the main components that a robotic system in LN must have:

- **The LN Manager:** The LN Manager is the central program that manages processes, executions, and dependencies. It consists of a GUI from which you can execute and control processes and that can be edited via a configuration file.
- **LN Clients:** An LN Client corresponds to a process that uses the LN client library API for the language in which it is written to communicate and exchange data with other clients.
- **Message Definitions:** Messages are the type of data structure used in internal communication between different LN clients, to exchange data even between different hosts, architectures, or programming languages.
- **LN daemon:** The LN daemon is a program that runs in the background hosting the data exchanges and setting up connections and communication.

As mentioned in section 2.2, the calculations concerning BO are performed by means of the BayesOpt software [31]. In order to perform the grasping process and feed BayesOpt with the data coming from the object's grasp, in an iterative and coordinated way, the LN middleware has been used. The implementation consists of a service function, in which the node that adapts the BayesOpt functionalities acts as a client of the service, asking the node in charge of executing the grasping process, to evaluate new poses, sent as LN messages, and provide it with new data. The general scheme is represented in Fig. 4.2 and the node acting as a service provider will be explained in more detail in the next section.

4.2 Hardware

The following section is a short introduction to the different robots used in this work, including one robotic arm and two robotic hands. It should be noted that the fact that both hands are underactuated adds additional uncertainty to the optimization estimates and serves to check how good the method is at handling them.

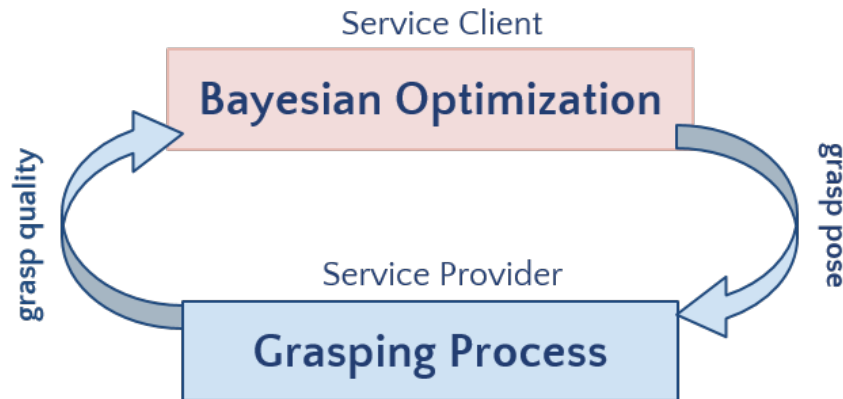


Figure 4.2: General scheme of the active grasping process implemented in Links and Nodes.

4.2.1 ReFlex TakkTile Hand

The ReFlex TakkTile hand [41], developed by RightHand Robotics, is a low-cost three-fingered robotic gripper equipped with tactile sensing capabilities that enable it to perform precise and adaptable object manipulation. This commercial hand served as the first introduction of the method to a real hand, testing the portability of the pipeline implemented in LN, to a robotic system controlled by ROS (Robot operating system).

Design

This gripper contains underactuated joints in the three fingers, meaning one less degree of freedom for each finger. These fingers have a revolution joint in the proximal that connects the proximal link to the knuckles. The joint that connects the proximal link and the distal link is a piece of cast urethane that can't be commanded directly because it is coupled to the proximal joint. This coupling is done using a tendon that allows a single actuator to control the movement of the entire finger. In addition, this gripper has a fourth actuator to modify the preshape angle of two of the fingers at the same time, resulting in a total of 4 degrees of freedom.

The ReFlex hand uses the Dynamixel actuators developed by Robotis, which allow for a fingertip payload of about 15 N and a maximum joint velocity of about 330°/s and provide back speed, position, torque, temperature, and voltage data while working.



Figure 4.3: ReFlex TakkTile by Right Hand Robotics (taken from [41])

Figure 4.4 shows the graphic scheme of the hand indicating the position of the actuators and the different links, with a proximal length of 60 mm, a distal length of 40 mm, and a flexible joint length of 10 mm.

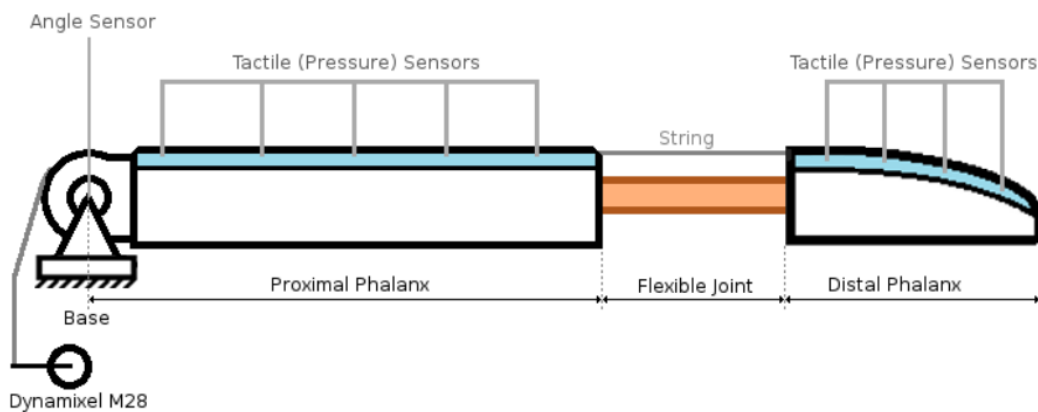


Figure 4.4: A diagram showing the location of the ReFlex TakkTile's sensor concerning the base of a finger. [42]

Sensors

ReFlex Takktille contains several pressure sensors on each finger, five on the proximal phalanx and four on the distal phalanx, as shown in Fig. 4.4. This array of sensors uses the MPL115A2 from Freescaler [43], which In addition to other instrumentation includes a MEMS pressure sensor. These pressure sensors provide the magnitude of the applied force perpendicular to the object surface in a relative or normalized form, with a sampling rate of 100 Hz. To calculate the studied metrics, ReFlex allows the position and orientation of the sensors to be estimated using the ROS package *tf*. However, since it is an underactuated hand, the position, and orientation of the coordinate frame that *tf* provides for each sensor is an approximation established using forward kinematics. To perform kinematics computations, this gripper counts with angular sensors, based on the Hall effect, and is located in the base-proximal joint, to measure position and velocity. It also contains load sensors directly located in the actuators to calculate the force exerted based on inferred values.

On the other hand, the angle of the distal joint is calculated from the difference between the tendon encoder and the proximal joint encoder, which makes it less accurate.

4.2.2 CLASH Hand

The CLASH (Compliant Low-cost Antagonistic Servo Hand, [44]) hand, shown in image 4.5, is a compliant low-cost hand designed at DLR with a three-fingered humanoid design to grasp safely and reliably fruits and vegetables.



Figure 4.5: CLASH (Compliant Low-cost Antagonistic Servo Hand) (taken from [45])

Design

CLASH is inspired by DLR's variable stiffness actuation design in the Awiwi hand. It is characterized by strong modularity, with all the actuators located in the hand palm and a structure that favors the integration of different sensors. This hand counts with a thumb plus two differential fingers, which provide sufficient dexterity and task versatility, without involving too much control complexity. While the thumb has three degrees of freedom, including proximal and distal joints, the other two fingers are underactuated in the distal joints, which, as in the case of ReFlex TakkTile, means one less degree of freedom for each. Figure 4.6 shows the kinematic chain of the hand, with a palm baseline of $L_0 = 35$ mm, a proximal length of $L_1 = 70$ mm and a distal length of $L_2 = 30$ mm.

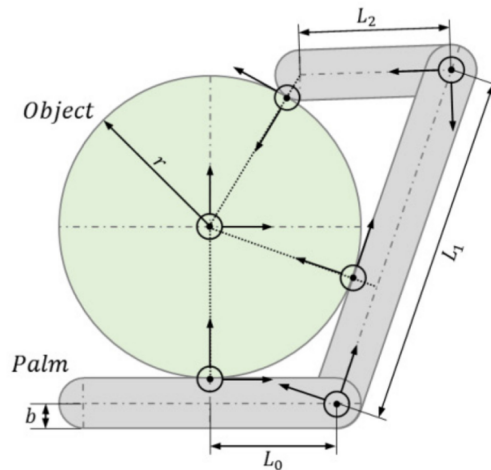


Figure 4.6: Kinematic optimization of the length of palm and finger segments (taken from [44]).

The proximal joints of the differential fingers can be actuated independently, but the distal joints of the two fingers are coupled, allowing for better handling of environmental constraints, such as sliding over curved surfaces to obtain better contact. Both types of fingers use the same servo module including four motors, two of which are used to increase the fingertip force. This translates into a maximum fingertip force of 10 N in the case of differential fingers, resisted by a maximum fingertip force of 20 N on the thumb and a maximum joint velocity of about $360^\circ/\text{s}$. The scheme of the mechatronic structure of the hand can be found in Fig. 4.7, as well as additional information on tendon coupling in the corresponding bibliography.

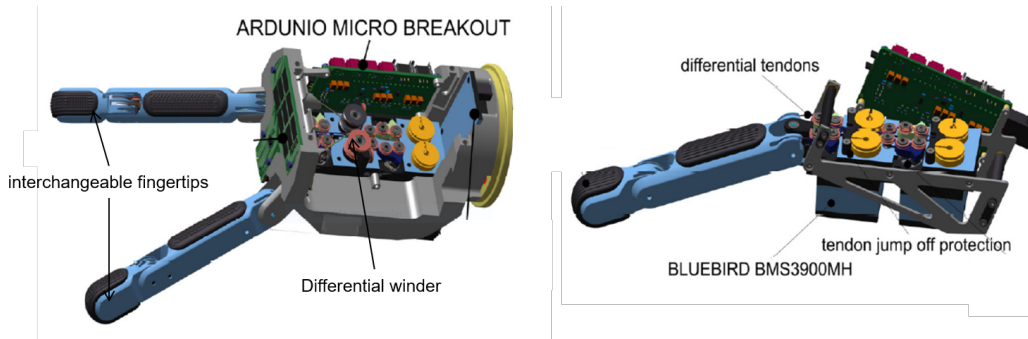


Figure 4.7: Mechatronic structure of the CLASH 3F hand (taken from [44]). (right) Thumb module. (left): Differential finger module.

Compared to a stiff hand such as the ReFlex TakkTile, CLASH offers greater robustness in many situations where the grasp configuration results in collisions with the environment. In these cases, a rigid joint may break completely, whereas the impact on a tendon would only dislodge it from its guide. Furthermore, due to its modularity CLASH has the ability to dislocate in those situations where an impact causes a lateral displacement in the direction in which the proximal joints have no degree of freedom. Together with the finger coupling, variable stiffness allows greater adaptability in grasps with different types of objects, in which a lower tendon pretension can prevent the slippage of a curved object or deformation of objects of low consistency when pressure is exerted.

When performing the grasping of the object there is an important difference between how it is simulated and how it occurs on the real robot when working with the CLASH hand. CLASH's motion control also allows its joints to move at a constant speed until a certain torque is reached, however, they do not move simultaneously in both proximal and distal joints. First, the proximal joints are closed until the limit torque is reached at this joint and then the distal joints are closed until the same condition is reached. This difference can result in substantial changes in how the object is placed between the fingers for the same hand pose in simulation and reality. In addition, as indicated before, CLASH allows the pretension in the tendons of the differential fingers and thumb to be configured. The consequences of this change in stiffness are also analyzed in the experimentation.

Sensors

One feature that makes this hand low-cost is that it uses Arduino to control the servos and most of its sensors. It uses Bluebird BMS-3900 MH and the measurements on their potentiometers are fed back to the Arduino microcontrollers to estimate the finger positions. Tendon forces are obtained from analog Hall sensors and taken into account, together with the coupling information, to estimate the joint torques needed in the admittance control.

CLASH hand can be provided with different tactile sensors. For this project, XELA uSkin Patch 4x4 sensors [46] are used on the fingertips. Each fingertip has one patch with 16 sensing points, each measuring a 3-axis force based on the Hall effect that includes the shear forces (X , Y), tangential to the surface, and the normal force (Z), perpendicular to the surface. The sampling rate is 100 Hz and the maximum normal and shear forces of the sensor are 18 N and 4.5 N approximately. Figure 4.8 shows the distribution of the sensors in the patch.

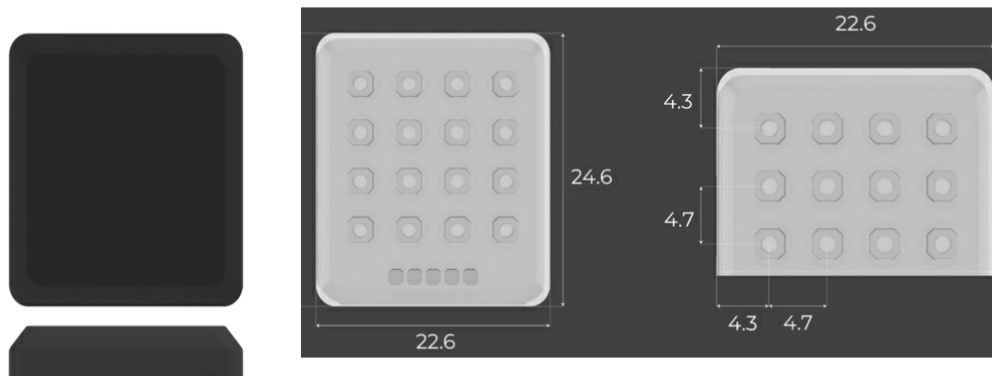


Figure 4.8: XELA uSkin Patch 4x4 sensor scheme

A disadvantage of the XELA patches is that they provide raw readings of the force exerted on the sensors, so these uncalibrated measurements cannot be taken directly as the actual force exerted on the sensor. In order to make use of these values, the first step was to distinguish what raw magnitude difference was observed when contact was applied to each of the sensors. For this purpose, ten contacts were made with very light pressure, and an average value was calculated for the difference between the raw

magnitude obtained when no contact was applied and the magnitude obtained when pressure was applied. The value obtained was 20 units and its validity was checked in ten new contacts and in the rest of the experimentation.

An important aspect to take into account is that each contact exerted on the patch causes a deformation in the silicone membrane that covers the sensors, shown in Fig. 4.9.A, which can directly affect the readings obtained by them. Therefore, when making consecutive contacts it is necessary to update the raw measurement obtained when there is no contact on the patch, and then obtain the difference when the contact is made. This process can be understood as re-calibrating the measurements to zero. Once a contact has been determined in one of the sensors, the force vector on the sensor is obtained directly from the measurements in the three coordinate axes provided by the XELA software itself. Since it is not possible to obtain the module of this vector in Newtons, when working on the real hand, unit force vectors on the contacts are assumed so that all subsequent analysis will be focused on the direction of the forces and not on their magnitude. This is an important limitation that will be addressed later in the experimentation. Moreover, as can be seen in Fig. 4.9.B, the direction of the XELA values obtained is towards the inside of the sensor, so the force that the fingertips exert on the objects, which is the one we want to consider, will have the opposite direction.

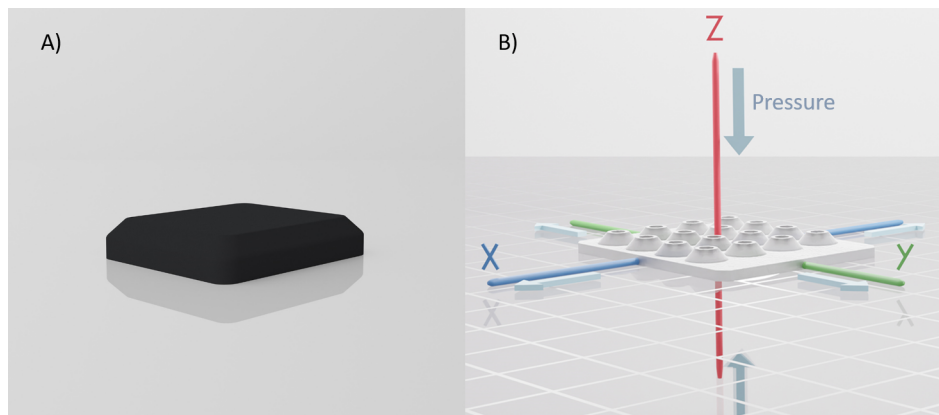


Figure 4.9: 3D representation of the XELA uSkin Patch 4x4 (taken from [47]). A) Silicone membrane; B) Force direction in the 3 Cartesian axes.

Apart from obtaining information on the contact forces from the XELA sensors, the contact points analysis when working with clash has required the computation of the position and orientation of each sensor through the kinematic analysis of the arm-hand assembly. To compute the position and orientation of each one of the sensors with

respect to the world reference, the scheme shown in Fig. 4.8 is used to obtain their position concerning one of the corners of the patch. Finally, the computation of the position and orientation of this same corner with respect to the world reference has been implemented using forward kinematics, by means of the same kinematic model of the hand used in the simulation. The transformation considered for each sensor is summarized in the Eq. 4.1:

$$T_{world}^{sensor} = T_{patch}^{sensor} * T_{distal}^{patch} * T_{proximal}^{distal} * T_{eeef}^{distal} * T_{world}^{eeef} \quad (4.1)$$

One of the drawbacks of this kinematic model is that it considers the hand to be fully actuated, which does not correspond to reality. The fact of carrying out the real experimentation with an underactuated hand, adds more error to the calculations performed to evaluate the grasp through the position of the sensors, which translates into another source of uncertainty that BO has to handle.

The LN middleware explained in section 4.1.2 provides all the inter-process communications related to the XELA sensors.

4.2.3 Light Weight Arm

The Light Weight Robot (LWR) III [48] [49] is a robotic arm composed entirely of revolute joints, to imitate human manipulation capabilities. This robot is prepared to be connected to any hand or gripper by a standard robot interface flange, which makes it a perfect arm for carrying out different manipulation tasks. Figure 4.10 shows the LWR in its home position and the workbench on which it is located.

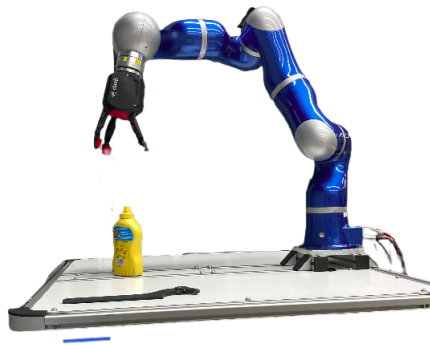


Figure 4.10: Light Weight Robot III on the laboratory bench.

Design

The main characteristic of this robot is that it has 7 degrees of freedom, coming from its seven revolution joints. More degrees of freedom than is necessary for basic functionality. Its entire design is focused on modularity, with all the electronics (except the PC controller) integrated into intelligent joint units inside the robot and protected by links made of carbon fiber composite, which allow a relatively small weight of about 14 kg.

The human-like design is composed of a first roll-pitch-roll combination imitating the shoulder and upper arm, then pitch-roll rotations for the elbow and forearm, and the final pitch-roll rotations for the wrist, where the gripper is attached. This allows multiple and versatile configurations with a maximum reachability of 936 mm. The joints units count with a brushless DC motor (DLR robodrive in Fig. 4.11), specifically developed in DLR, achieving great efficiency with half of the size and allowing the robot to handle a maximum payload of 14 kilos and maximum joint speeds of 1.9 rad/s.

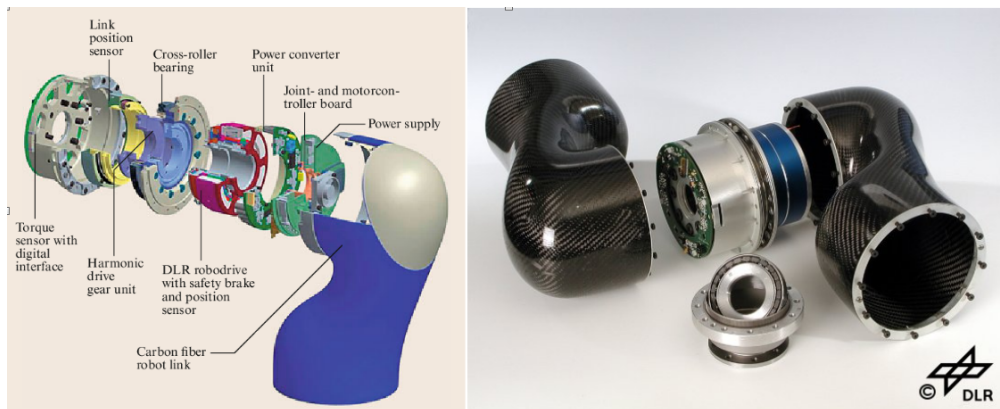


Figure 4.11: LWR III joint unit (taken from [49]): joint components scheme (left), real joint unit in carbon fiber (right).

Sensors

Most of the sensing components were developed for the previous version LWR II and transferred to the new one too. This includes a motor position sensor, a link side position sensor, and a joint torque sensor at each of the joints, which allows the robot to be position, velocity, and torque controlled. All of them can be seen represented in Fig. 4.11.

We can find two different types of position sensors based on the Hall effect, including a high-resolution incremental sensor to obtain smooth motor commutation and accurate joint position for control, and an absolute sensor to provide absolute position feedback of the output shaft for joint reference.

On the other hand, a torque sensor is mounted on the joint output. It consists of strain gauges in a full bridge configuration, to increase sensitivity, that operate at a 3kHz rate. This provides a higher control bandwidth for responding to disturbances, in contact with stiff and unknown environments.

4.3 Objects set

Experimentation has been carried out with several objects belonging to the YCB object and model set [50], shown in Fig. 4.12. It allows us to test the generality of the method in objects with different shapes, weights, sizes, and variety in other characteristics such as symmetry or the position of the center of mass.



Figure 4.12: YCB objects set (taken from [50]).

From the whole set of objects, the table 4.1 shows those that have been selected for use in this work. First of all the **chips can** provides a simple and symmetric shape to start the experiments and a considerable size to find enough grips. In the next degree of difficulty, we find the **mustard can**, with a lower degree of symmetry and a slightly more complex shape in the cap area and the middle area of the bottle. In addition, it should be noted that it is a heavier object containing a fluid inside so its center of mass differs to some extent from the center of the object. The **mug**, is a less heavy object but also of smaller size, which means a smaller area to grip. In addition, its shape is more complex and asymmetrical, with the hollow inner part and the handle, which presents the challenge of finding a predefined grasp on the design of the object. In the **power drill**, its size, weight, and center of mass add complexity to the grasp. Its shape

4 Experimental setup

is much more complex than in the previous cases and includes the handle area as a predefined grasp in the design. Finally, the **Lego** toy allows experimentation with a shape that is also complex, including completely flat areas as well as edges and curved areas. Although its weight is lower, its center of mass is not in the center of the object due to its asymmetry and it has the complexity of being made of a slippery material.






ID	Picture	Object	Mass (g)	Dims. (mm)
1		Chips can	205	75 x 250
6		Mustard bottle	603	50 x 85 x 175
25		Mug	118	80 x 82
35		Power drill	895	46 x 170 x 184
73 (a-m)		Lego	225	130 x 180 x 200

Table 4.1: Objects of the YCB object and model set used for the experimentation. Object IDs are consistent with Calli et al. [50].

During experiments, objects are placed at a distance of 60 cm from the base of the LWR, on the X-axis of the arm reference (world). This position has been measured and marked on the table that supports the arm using adhesive tape and marker, as well as two perpendicular lines to configure the orientation of the objects. Since in our experimental case the objects must be completely static, a "blue-tack" type adhesive has been used to fix them to the table at different points, since it is an adhesive that adapts to different roughnesses and materials.

5 Experimental results

This chapter contains the experimental results obtained from the research underlying this thesis. The primary objective of this study was to empirically evaluate and analyze the results obtained when grasping different objects and shed light on the performance of BO in real robotic grasping. The previously explained experimental setup, aims to provide valuable insights and contribute to the existing body of knowledge in the field, serving as a foundation for the subsequent discussion and conclusions.

5.1 Free-Hand simulation

The first section shows some results in the free-hand simulation, the same setup used in [16] and shown in Fig. 3.4.A, using the robotic hands and objects considered in this thesis. The outcomes shown in Tab. 5.1 and Fig. 5.1 correspond to the Q_e measure, normalized with respect to the maximum value obtained from experiments with both hands (CLASH and ReFlex TakkTile) in all the objects set. The results were obtained from the execution of 10 experiments for each case, using 100 iterations and 20 initial samples for each experiment. It should be noted that, in this case, if the hand collides with the object or does not obtain a grasp with force-closure, the final evaluation of the grasp Q_{grasp} is considered 0 ($Q_c = Q_f = 0$), as described in the Eq. 5.1, where Q_c and Q_f work as binary coefficients.

$$Q_{grasp} = Q_c * Q_f * Q_m \quad (5.1)$$

Hand	Object	Mean	Maximum
clash	Mustard bottle	0.7676	0.9669
	Power drill	0.6383	0.9496
reflex	Mustard bottle	0.7427	0.8033
	Power drill	0.5794	0.8633

Table 5.1: Free hand simulation: Mean and maximum outcomes

As expected, the mean value of the quality outcomes is better for the mustard bottle, as the simplicity and symmetry of its shape result in a higher quality of grasps

throughout the iterations. On the other hand, both hands show great results, with a quality higher than 50 percent of the maximum. Although CLASH shows considerably higher maximum values, both the learning over iterations shown in Fig. 5.1 and the mean outcomes are similar, so the difference may be due to how the hands are defined in the URDFs and how this influences the forces generated. Contrary to what might be expected, taking into account the contacts along the finger does not seem to improve the convergence of the method with respect to only considering fingertips. As a last observation, the method shows the difficulty in finding specific pre-defined grasps according to the design of the object, in this case, the power drill handle.

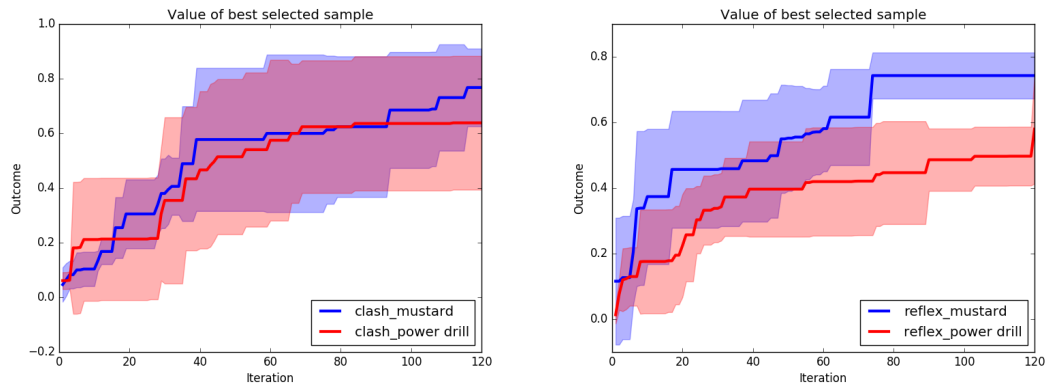


Figure 5.1: Evolution of the quality of the grasps obtained in the mustard bottle and power drill. Left: Grasps obtained with CLASH. Right: Grasps obtained with ReFlex TakkTile.

This experiment served as a basis for developing the simulation on the complete model, obtaining the values to normalize the metrics, and verifying that both the metrics and the robotic hands obtained good outcomes and performance.

5.2 Full model simulation

The following experiments show a comparison between the results obtained during the implementation of the method on the complete robot model simulation, shown in Fig. 3.4.B, and its performance using the different grasp metrics. In this case, For all the presented experiments, the method was tested using 20 initial samples and 50 BO iterations for each experiment in three different objects: the mustard bottle, the mug, and the power drill.

The first thing we can observe, both in Tab. 5.2 and Fig. 5.2, is how by adding the LWR and the workbench in the simulation, the outcomes obtained when evaluating the grasp according to the Eq. 5.2 (the same as in Eq. 5.1 but adding the coefficient of reachability) are much lower than those obtained in the case of the free-hand simulation. In this new setup, the kinematic limitations of the LWR, which reduce the reachability of the grasping pose, and the collisions detected when including the arm and the workbench result in a higher number of grasping poses evaluated as 0, which hinders the convergence of the method.

$$Q_i = Q_r * Q_c * Q_f * Q_\epsilon \quad (5.2)$$

The addition of CP to differentiate between a collision and not touching the object results in Eq. 5.3.

$$Q_{ii} = Q_r * Q_f * (Q_\epsilon + CP) \quad (5.3)$$

Tab. 5.2 shows the comparison between the results obtained by applying the $CP_{Castanheira}$ and the CP implemented in this work, both explained in section 3.4.2. Both mean and maximum outcomes increase with the addition of CP, and in Fig. 5.2 we can see how active learning reaches much higher values already in the first iterations. These results demonstrate how assigning a higher value to CP the lower the degree of collision with the object drives the acquisition function towards near-object poses while preventing collisions from occurring and improving the convergence of the method.

In the next step of the experimentation, the CR explained in section 3.4.4 is added to differentiate between a non-force-closure grasp and not touching the object, resulting in Eq. 5.4

$$Q_{iii} = Q_r * (Q_\epsilon + CP + CR) \quad (5.4)$$

The results of Tab. 5.2 show the increase in performance of the method when including this coefficient, and also the results obtained on more complex objects with maximum outcomes above 50 percent. The green line in Fig. 5.2 also shows an improvement in the learning trend at each iteration, concluding in the improvement of the convergence of the method in a good grasp.

This experiment also served to test the performance of the method by including R in the grasp evaluation, resulting in Eq. 5.5, where a is the weight given to R over Q_ϵ and $b = 1 - a$. The intention is to reward those grasps with higher reachability.

$$Q_{iii} = Q_r * (a * R + b * Q_\epsilon + CP + CR) \quad (5.5)$$

However, Tab. 5.2 shows how R greatly decreases the outcomes. This is because when the position of the object is not difficult to access for the robot, R obtains very high values (between 0.8 and 1) when the pose is reachable. Therefore, its influence on the final result causes that the acquisition function does not prioritize improving the grasping quality obtained from Q_e and the outcome is lower.

		Mustard bottle		Mug		Power drill	
		Mean	Maximum	Mean	Maximum	Mean	Maximum
Q_i		0.3177	0.5182				
Q_{ii}	Castanheira	0.5464	0.8154				
	García-L	0.5621	0.9013				
Q_{iii}		0.6572	0.9345	0.6062	0.8713	0.4964	0.6246
Q_{iiii}	a = 5%	0.4147	0.7382				
	a = 10%	0.5044	0.8729				
	a = 15%	0.4635	0.7192				

Table 5.2: Full model simulation: Mean and maximum outcomes

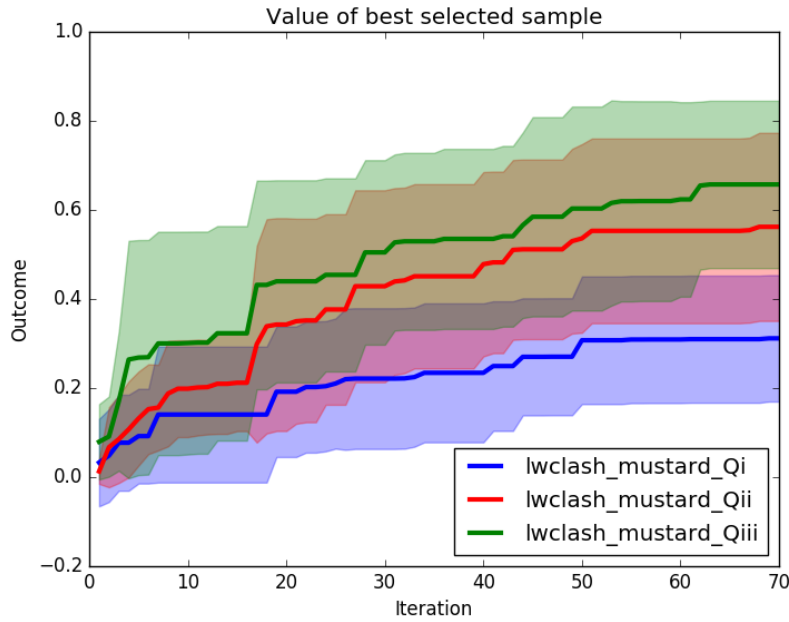


Figure 5.2: Comparison of the evolution of the quality of the grasps obtained using different implementations of the grasping process (Q_{grasp}).

In the next part of the full model simulation experiments, the performance of the method using each metric was tested. Tab. 5.3 and Fig. 5.3 show the time for each metric to achieve 50% of the maximum outcome (obtained in the free-hand experiments) for each metric, and the number of times that the 50% outcome was not reached. Each metric was tested in 10 experiments for each one of the three different objects studied. The function considered to evaluate these grasps is the same as in Eq. 5.4, replacing Q_ϵ with the corresponding grasp metric (Q_{iso} , Q_v , Q_{uni}). Tab. 5.3 also contains the mean outcomes obtained on each case.

	Mustard bottle			Mug			Power drill		
	50% (seg)	Not reached	Mean	50% (seg)	Not reached	Mean	50% (seg)	Not reached	Mean
Q_{iso}	12.52	0/10	0.689	21.31	0/10	0.619	27.16	3/10	0.533
Q_ϵ	31.00	3/10	0.657	32.93	4/10	0.466	38.95	6/10	0.406
Q_v	55.46	5/10	0.476	46.13	5/10	0.456	38.15	6/10	0.393
Q_{uni}	18.77	0/10	0.716	14.94	0/10	0.677	12.02	0/10	0.763

Table 5.3: Full model simulation: metrics convergence

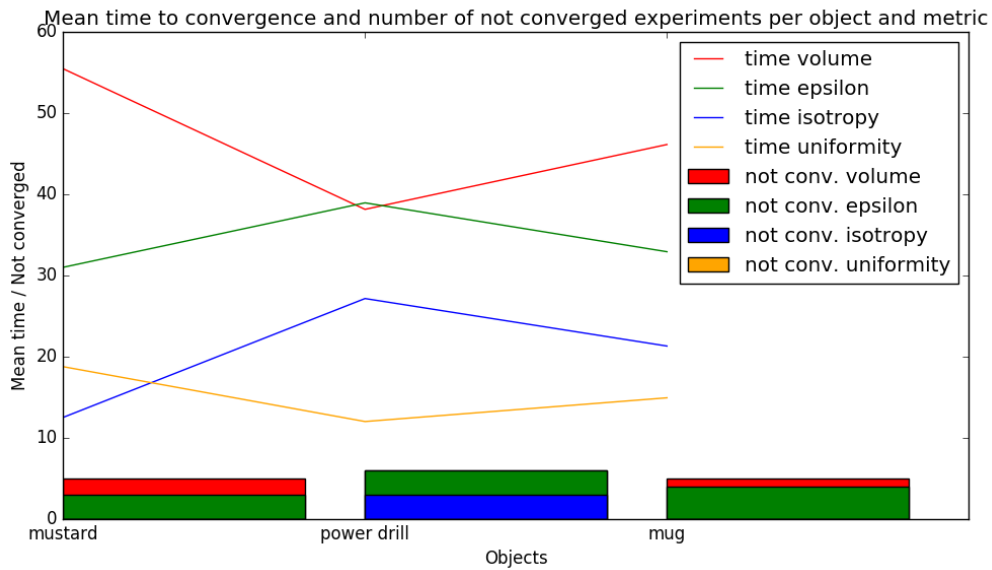


Figure 5.3: Mean time to convergence and number of not converged experiments per object (mustard bottle, mug, and power drill) and metric (Q_{iso} , Q_ϵ , Q_v , Q_{uni}).

The blue and yellow lines in Fig. 5.3 show the average convergence time obtained with Q_{iso} and Q_{uni} respectively, of much lower values than the times obtained with Q_ϵ and Q_v . In the vertical bars, it can be seen how these last two metrics did not reach 50% of the maximum outcome a greater number of times, being Q_{iso} the only case in which this value was reached in the totality of the experiments. The mean outcome values shown in Tab. 5.3 are consistent with these results, with Q_v being the metric with the worst performance in obtaining an optimal grasp on each object. These results have their logic in the definition of each metric. As explained in section 2.3, Q_{iso} and Q_{uni} are ratios of the minimum and maximum singular values of their respective matrices and are therefore unitless. On the other hand, Q_ϵ and Q_v are absolute measures extracted from the GWS, with their respective units of force and force times distance cubed. When normalizing these metrics, the range of values obtained for Q_{iso} and Q_{uni} will be lower and with values closer to 1 if the differences between the maximum and minimum singular values on each case remain similar. On the other hand, the variation in the absolute values of Q_ϵ and, especially, Q_v can be more accentuated because of the importance that small variations in the position or direction of the forces exerted on the object have in these metrics.

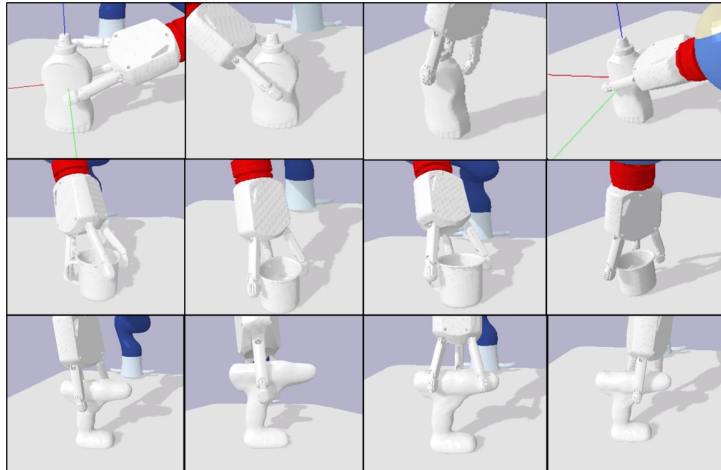


Figure 5.4: Some of the best grasps obtained with each metric for the mustard, the mug, and the power drill. From left to right: Q_{iso} , Q_ϵ , Q_v and Q_{uni} .

Fig. 5.4 shows some of the best grasps obtained by the optimization of Q_{iii} with each of the metrics in the considered objects. These results are clear examples of the difficulty for the method of finding the grasps defined in the object design, such as the mug handle or the power drill handle, especially if these grasps are close to the table, as the table may come into collision with the arm or hand.

As indicated at the end of section 2.2, during these experiments it was observed that in objects with more complex shapes and less good configurations as the power drill, the initial samples proposed by the LHS method were very decisive for the convergence. In order to conform a well-informed prior, they should contain some grasps allowing the probabilistic model to represent areas in which force-closure grasps exist.

5.3 Real world experiments

The following section presents the experiments performed in real robotic systems, from preliminary experimentation in ReFlex TakkTile hand to experimentation in the target environment shown in Fig. 3.4.C, which includes the workbench, the LWR robotic arm, and the CLASH robotic hand.

5.3.1 Experiment 0: Pipeline test

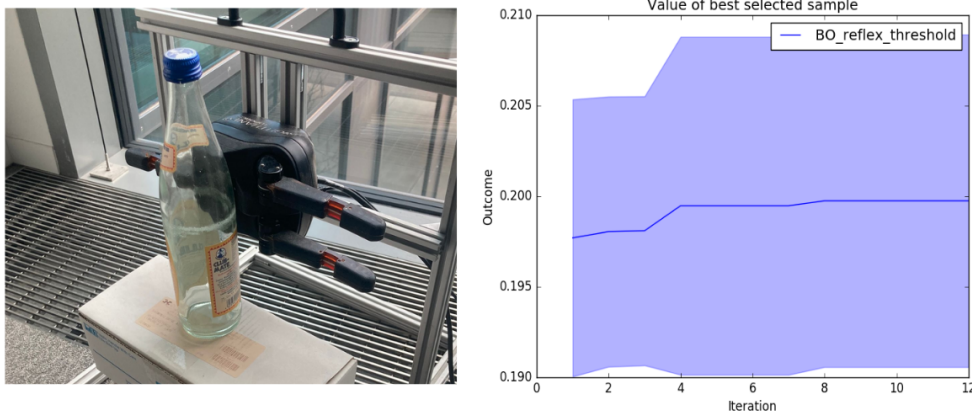


Figure 5.5: Left: experimental setup including the ReFlex TakkTile hand in a static position and a water bottle to grasp. Right: Evolution of the outcome when optimizing the force exerted by the ReFlex TakkTile fingers on the bottle.

The first runs of active grasping on a real robotic system were carried out in the setup shown on the left side of Fig. 5.5. This setup consists of the ReFlex TakkTile hand mounted on a static support, together with a water bottle as the grasping object. Due to the static position of the hand, the force that each finger exerted on the object was used as the parameter to optimize, to obtain conclusions about the implementation of the

iterative process in LN explained in section 4.1.2, and also about the implementation of the computation of the grasp metrics on the real robot. Although the graph shown on the right side of Fig. 5.5 is not of numerical interest, it is an example of a result in which the outcome improves after the first iterations when the pressure exerted on the object and therefore the number of sensors in contact with the bottle increases. This type of experiment served as well to add minor features to the pipeline, such as the possibility of commanding the start of each iteration and each part that makes up the grasping process explained in 3.4, to study them separately, correct possible errors in the setup or avoid possible collisions thanks to the preview of the simulation.

5.3.2 Experiment 1: Simulation vs. real environment

The realization of this experiment is focused on establishing the most suitable weights (w_1, w_2, w_3, w_4) for Q_m in Eq. 3.5, analyzing the performance of the method in different cases. To do so, different grasping poses were obtained in simulation using each of the implemented metrics and three different objects from the object set. Seventy iterations (50 BO + 20 initial samples) were performed for each optimal grasp, with a total of 5 experiments for each combination of metric and object. The goal was to check the portability of the optimal grasps computed in simulation to the real environment. As quoted in [11]: "We need to guarantee that the optimal grasp can be repeated in the presence of noise with sufficient quality". For this purpose, the number of successful grasps in the 5 experiments performed with each combination metric-object, and the mean outcome of each optimization were taken into account. To test the effectiveness of the grasp, the object was lifted several centimeters from the table after being grasped and held in that position for a certain period of time. A binary score (success or fail) was recorded for each optimal grasp pose. A grasp execution was considered a failure if the object fell or slipped during its lift motion, touching the workbench. The results obtained in this experiment are shown in Tab 5.4.

	Pringels can		Mug		Lego	
	Mean outcome	Success over 5	Mean outcome	Success over 5	Mean outcome	Success over 5
Q_{iso}	0.8023	3/5	0.8946	2/5	0.8623	3/5
Q_ϵ	0.6991	3/5	0.7526	4/5	0.6537	2/5
Q_v	0.8193	1/5	0.6972	3/5	0.4590	1/5
Q_{uni}	0.7722	3/5	0.7354	4/5	0.7974	2/5

Table 5.4: Experiment 1: Mean outcomes and number of successful grasps obtained using different metrics in the real environment.

The table shows mean outcomes of the optimal grasps higher than 50% in most of the cases. This is especially interesting in the case of Q_v , which only obtained 33.3% effective grasps in the total evaluated. Both Q_v and Q_ϵ are metrics based on GWS properties, which makes them very sensitive to positional errors introduced by the real robot, as demonstrated in the work of Weisz et al. [51]. However, Q_ϵ is much more restrictive and effective, as the result reflects, since it focuses on the maximum force that the grasp is able to resist and not on the whole wrench space.

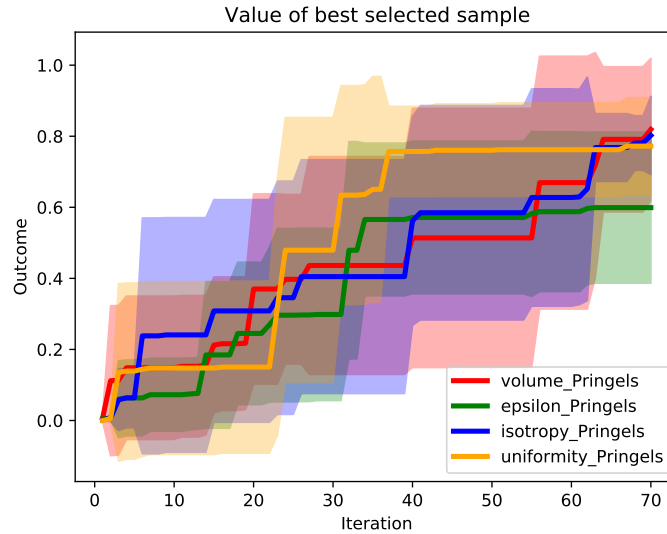


Figure 5.6: Comparison of the evolution of the quality of the grasps obtained using different grasp metrics to grasp the Pringles can.

Although Fig. 5.6 shows very similar learning in all metrics when grasping the Pringles can, highlighting a faster convergence in Q_{uni} , in the optimal grasps obtained by each of them and shown in Fig. 5.7 we can observe certain trends related to the computation of each metric. While the other metrics focused on exploring the top, Q_{uni} found side grasps in which the hand grasped the can very robustly (Fig. 5.7.3). The main reason this happens is that CLASH only has tactile sensors on the fingertips. When the Pringles are grasped from above, the contacts surround the can and the direction of the forces results in a stable grasp according to Q_v , Q_{iso} and Q_ϵ . However, in lateral configurations, a grasp with the hand wrapping around the can has very similar contact points to a grasp that only touches the can from the side. Not having contact points on the proximal phalanges causes information to be lost in the analysis. On the other hand, Q_{uni} does take into account the hand configuration and is able to

obtain this type of grasp pose.

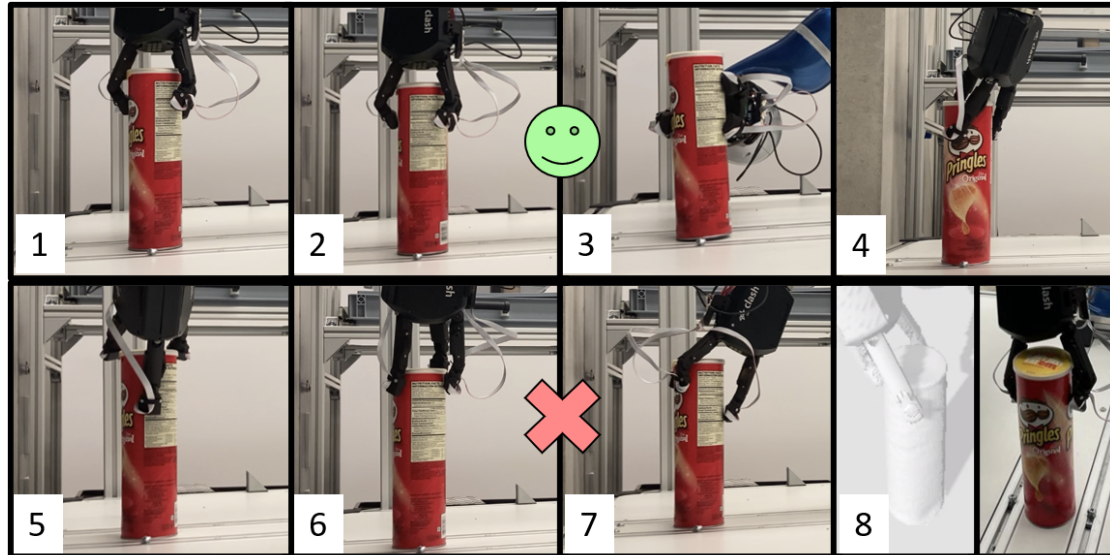


Figure 5.7: Some of the grasps obtained with each metric for the Pringles can. From left to right: Q_ϵ , Q_{iso} , Q_{uni} and Q_v . The images at the top show successful grasps, while those at the bottom show failed grasps.

As already stated in section 4.2.2, the configuration of the pretension in the tendons of the fingers and the third degree of freedom of the thumb allow CLASH to adapt with greater stability to grasp on curved surfaces as shown in Fig. 5.7.4. A reduced pretension allows the hand to adapt to the shape of the object, exerting less pressure on its surface to prevent it from slipping. However, curved surfaces also give rise to other evidence. In grasps such as the one shown in Fig. 5.7.8, the static simulation of the object allows the contact points to hold firm when pressure is exerted, resulting in a GWS that can be considered stable. However, when executing this type of grasp on a real non-static object, the curved surface causes the contact points to slip and the grasp fails.

Fig. 5.8 shows the outcome per iteration obtained with each metric when grasping the mug. It can be observed that the increase in the complexity of the object slows down the convergence of the method, although all the metrics reach an average optimal outcome higher than 50% of the maximum outcome, and the percentage of effective grasps is 65% of the 20 performed on the mug. The blue line corresponding to Q_{iso} reaches the highest outcome in this case, however, only 2 of the 5 grasps performed succeeded in lifting the mug. This is a clear indication that a more symmetrical distribution of forces

and torques around the object, as in Figs. 5.9.2 and 5.9.6, is not always a sufficient condition to obtain an optimal grasp when faced with the uncertainty introduced by positional errors and the unknown friction coefficient μ of the object.

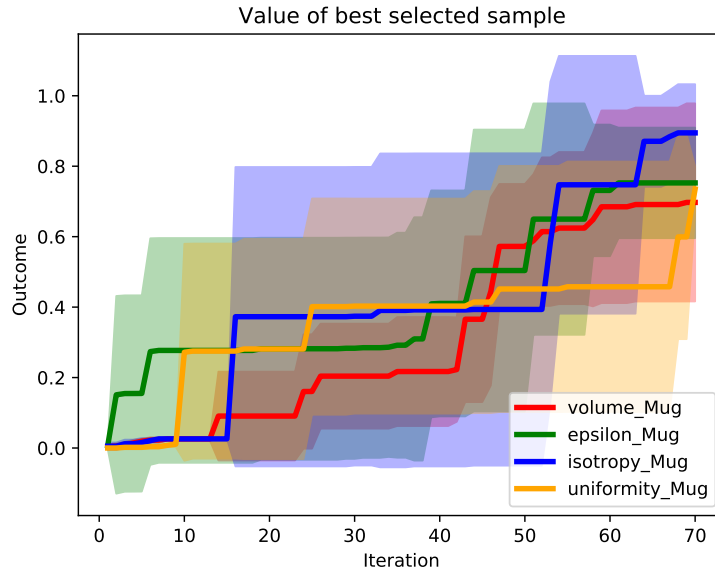


Figure 5.8: Comparison of the evolution of the quality of the grasps obtained using different grasp metrics to grasp the mug.

As in the case of the Pringles can, Figs. 5.9.6 and 5.9.8 show grasps in which the object slips from the hand when pressure is exerted. One way to classify the grasps to avoid these cases could be tracking the section of the fingertip where the contact occurs. In the case of the XELA sensors used in this work, if contact occurs on the 8 sensors closest to the distal, the grasp could be considered more robust than if it occurs on the 8 furthest from the distal, as this would be an indication that the object is closer to the palm. Figs. 5.9.6 and 5.9.7 show cases where the hand is located far away from the object and the force closure calculation is wrong. On the other hand, Fig. 5.9.1 shows a case in which the movement of the object when exerting pressure favors the grasp since the mug becomes completely introduced inside the hand. Fig. 5.9.3 is another example of how the result obtained by Q_{uni} wraps around the mug to a greater extent, as it is not based on contacts on the fingertips. All the images in this figure show how the method avoids those grasps where the arm is close to the workbench, resulting in grasps from above.

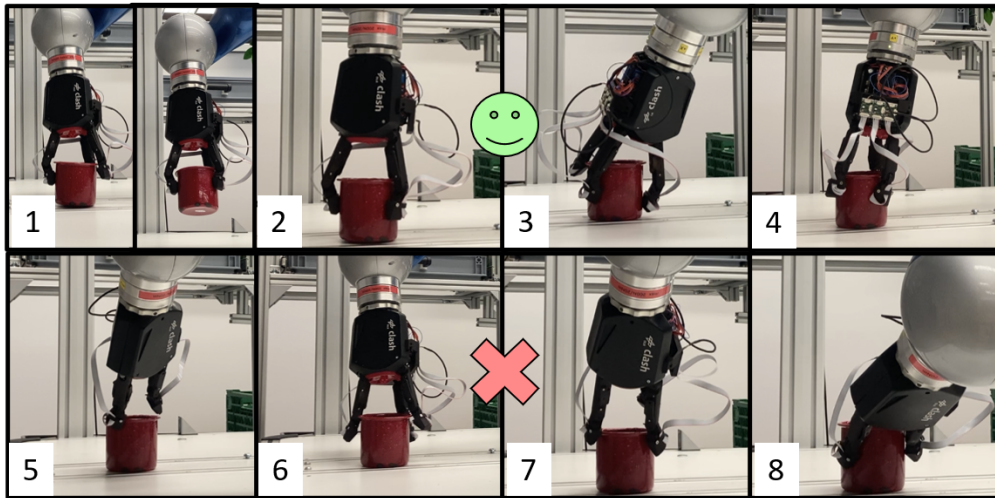


Figure 5.9: Some of the grasps obtained with each metric for the mug. From left to right: Q_e , Q_{iso} , Q_{uni} and Q_v . The images at the top show successful grasps, while those at the bottom show failed grasps.

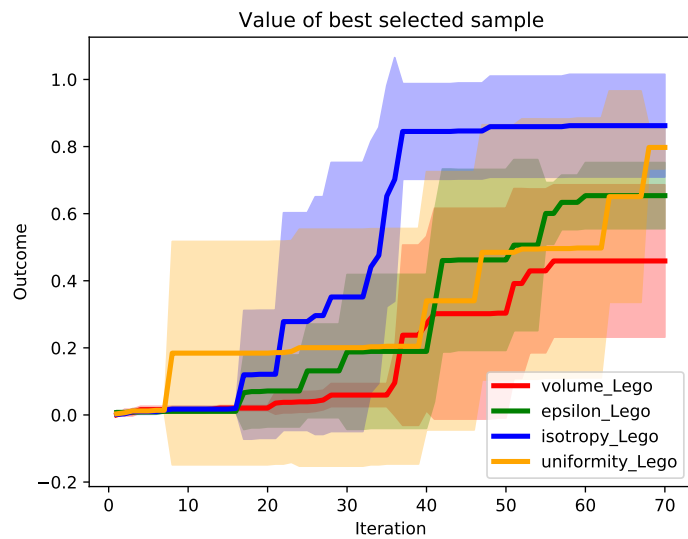


Figure 5.10: Comparison of the evolution of the quality of the grasps obtained using different grasp metrics to grasp the Lego toy.

Both the results shown in Tab. 5.4, as well as the graph in Fig. 5.10, show a considerable decrease in the performance of the method when confronted with an object of higher difficulty such as the Lego toy. However, in this case, the most determinant difficulty is given by a characteristic that is not taken into account by any of the metrics or heuristics used in the optimization, which is the type of surface of the object. In the images of Fig. 5.11 it can be seen how the whole object had to be covered with masking tape, to avoid the slipping of the fingertips, as it occurred in most of the grasps. Once this change has been made, two types of successful grasps can be observed. On the one hand, the ones in which one or more of the fingers are stuck on the toy's head (more distinguishable in the image of Tab. 4.1). Examples of these grasps are those shown in images 1 and 3 of Fig. 5.11. On the other hand, those grasps in which the fingers make contact on the edges of the object as shown in Figs. 2 and 4 of Fig. 5.11. Figs. 5.11.5 to 5.11.7 show failed grasps in which the object slips due to insufficient friction between the silicone of the fingertips and the flat surface of the object. In the case of Fig. 5.11.8, the lack of precision in the computation of the force closure condition causes a spurious grasp. Except for Q_e , none of the metrics considered guarantees the force closure condition in the grasp. Since Q_{iso} and Q_{uni} by their definition give rise to more symmetric contact point distributions surrounding the object, Q_v is the one with the highest risk of obtaining high values for this type of ineffective grasps where errors introduced by the real environment or by assumptions about the center of mass or the contacts can lead to a false force closure.

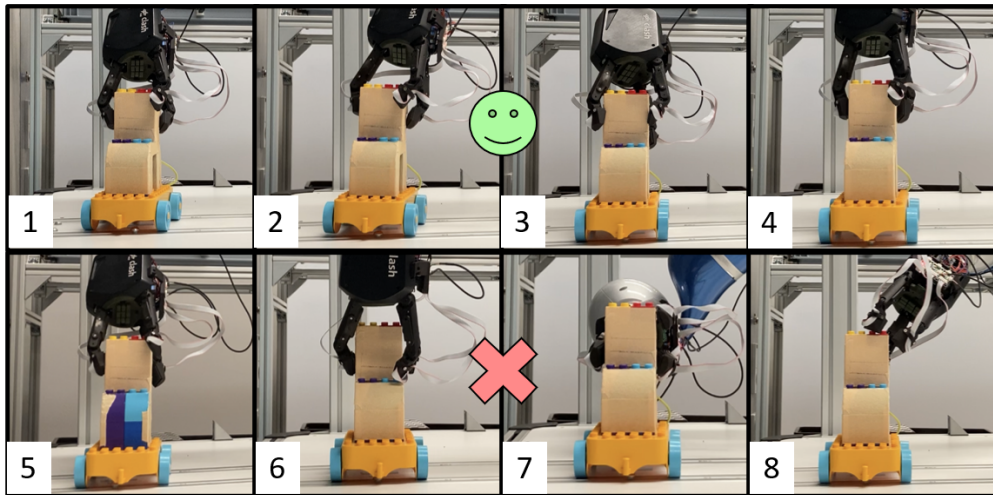


Figure 5.11: Some of the grasps obtained with each metric for the Lego toy. From left to right: Q_e , Q_{iso} , Q_{uni} and Q_v . The images at the top show successful grasps, while those at the bottom show failed grasps.

The configurations at the top of Fig. 5.11 grasp the flat surfaces and edges of the toy in a pincer-like manner. For this type of grasp to be successful it is necessary to increase the pretension of the tendons and the torque threshold that the fingers can exert, to obtain a greater momentum at the proximal joints. Although the section 4.2.2 indicates that CLASH is capable of exerting forces of 10 N and 20 N on the differential fingers and thumb respectively, these measurements may vary depending on how the tendon pretension behaves when grasping an object. In heavier objects, such as the mustard bottle, CLASH was unable to exert enough force on the fingertips to grasp the object in a clamp-like grasp (Fig. 5.12.1). In the case of the power drill, CLASH managed to lift the object, in those grasps where the fingers completely surrounded the tool head (Fig. 5.12.3), and the force was more evenly distributed. This difficulty caused up to 4 failures in the servomotors of the hand, due to their overexertion, so it was decided to continue with the experimentation on lighter objects such as the Pringles can or the Lego toy.



Figure 5.12: Some of the grasps performed with CLASH on mustard bottle and power drill. 1: failed clamp-like grasp on the mustard; 2: failed clamp-like grasp on the power drill; 3: successful grasp surrounding the power drill head.

5.3.3 Experiment 2: Optimization in real environment

In the following experiment, the behavior of the Bayesian optimization was tested by performing the iterations directly on the real robot. The pose in two spatial axes was optimized. X and Z axes for the Pringles can, keeping Y fixed, and X and Y axes for the mug and the Lego toy, maintaining the Z axis fixed. Roll angle was optimized as explained in section 3.3.

It is important to remember that although the collisions and the reachability of the position are previously checked in simulation for each iteration, the obtained grasp pose is evaluated directly in the real environment, obtaining the quality of the grasp

from the sensors and measurements provided by the robotic system. The set of objects used in this experiment was also the same as in Experiment 1. Five optimizations were performed for each object, obtaining the mean outcome in each iteration to graphically represent the learning in 30 iterations (25 BO + 5 initial samples).

	Mean outcome	Best outcome	Success over 5
Pringles	0.6436	0.8423	3/5
Mug	0.7189	0.8579	4/5

Table 5.5: Experiment 2: Mean outcomes and number of successful grasps

Tab. 5.5 shows the number of successful grasps in the 5 optimizations and the outcomes obtained when evaluating the grasp according to Eq. 3.5 and Eq. 3.6. The combination of the metrics aims to obtain an optimal result according to the aspects evaluated by each one (resistance to disturbances, stability, and evenness of the distribution of forces), to compensate metrics with worse convergence with those with better convergence, and to compensate those metrics more influenced by the uncertainty of the real environment. The weights of Eq. 3.5 were calculated as the ratio between the number of successful grasps obtained by each metric and the number of total successful grasps in Experiment 1. This expression is defined in Eq. 5.6, where $i = \{1, 2, 3, 4\}$ and $Q = \{Q_{iso}, Q_{uni}, Q_{\epsilon}, Q_v\}$.

$$w_i = \frac{(\text{successful grasps})_Q}{\text{total successful grasps}} \quad (5.6)$$

In the first optimizations, the importance of the initial samples in the final result can already be observed, especially when the number of total iterations is reduced. In the case of the Pringles can, when a high outcome side grasp is found in the initial samples, the search is focused on that exploration area, more related to greater outcomes in Q_{uni} , resulting in a grasp like the one shown in Fig. 5.13.1. If the initial samples find a high outcome in the upper area of the Pringles can then the rest of the search continues to focus on that type of grasp, more related to high outcomes in Q_{ϵ} or Q_{iso} , resulting in optimal grasps like the one shown in Fig. 5.13.2. Although the search space is unconstrained and the algorithm itself maintains the balance between exploitation and exploration, addressing the learning with heuristics focused on improving the prior can be a good solution for the convergence of the method in areas of interest.

Despite adding a threshold to improve its accuracy, the force closure computation proved to be insufficient to deal with the kinematic and dynamic inaccuracies of the robot, as well as with the assumptions defined for working with the real environment.

Several errors in predicting good grasps were observed throughout the iterations. Two of these false force closures, which also resulted in high outcomes, are shown in Figs. 5.13.3 and 5.13.4. Once again, it was observed that Q_v was the metric most affected by false force closures throughout the iterations, driving the acquisition function toward this type of grasps.

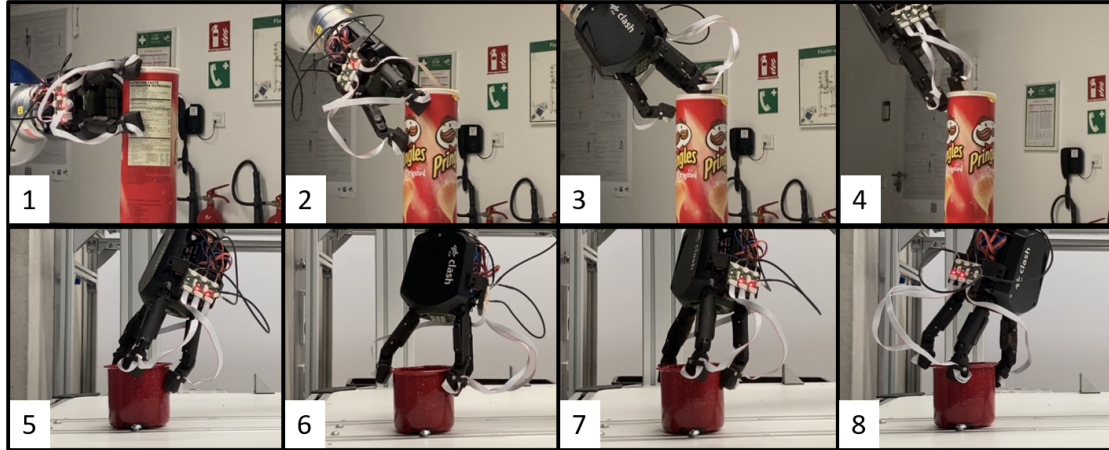


Figure 5.13: Some of the grasps performed with CLASH on Pringles can and mug.

Fig. 5.14 shows faster learning, with higher outcomes for the mug case. The size of the Pringles can, especially in its Z-axis reduces the convergence of the method, however, exploring this axis allows to obtain lateral clamp-like grasps. Although the Z-axis could have been set at a distance closer to the mug, the exploration in the X and Y axes gave results in successful grasps as those observed in Figs. 5.13.5 to 7, with more significant variations in the roll angle. On the other hand, Fig. 5.13.8 shows a non-effective grasp defined as optimal by the method, qualitatively less robust than the other grasps explored, but with a higher outcome.

The last experiment, intended to test the effectiveness of the method on the Lego toy, could not be completed. Constant experimentation and exposure of the tendons to stress for long periods resulted in the overloading of the servomotors. This type of unforeseen event gives evidence of the need to contemplate in the method technical details of the robotic system itself such as the underactuated design of the hand or the variability of the stiffness in its joints and also the difficulty of not knowing the weight distribution of the object, to adapt the grasps to the characteristics of the gripper. Once seen the influence of the tendon pretension in the obtained grasps, a possible solution would be to study the optimization of this pretension as a fundamental parameter of

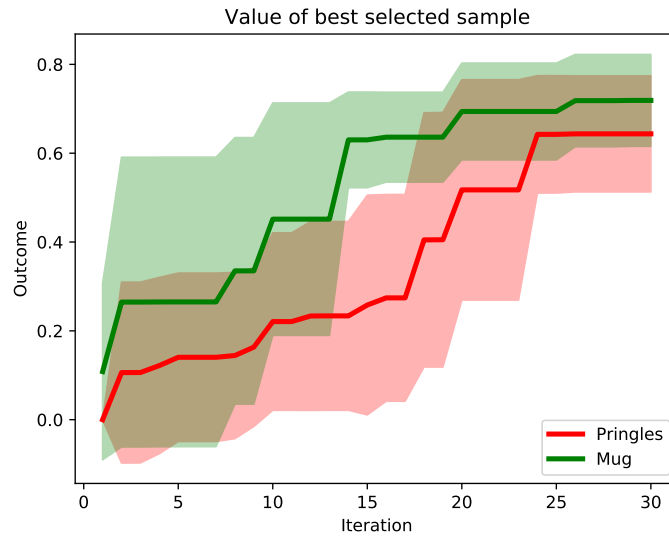


Figure 5.14: Comparison of the evolution of the quality of the grasps obtained grasping the Pringles can and the mug with the real robot.

this type of hand.

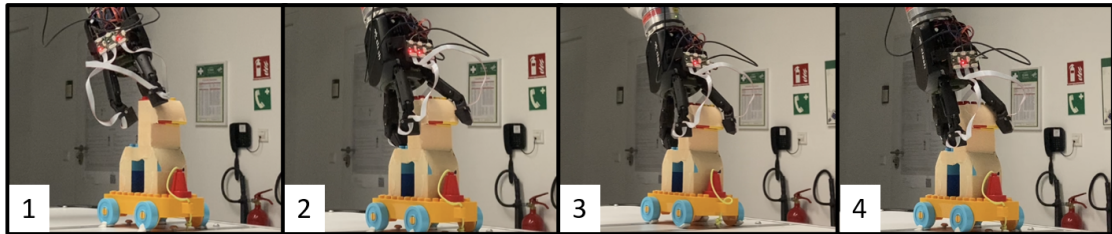


Figure 5.15: Some of the grasps performed with CLASH on

Fig. 5.15 shows, in iterative order, some of the grasps tested during the grasp optimization process on the Lego toy, from a qualitatively less robust grasp (5.15.1), to a grasp with greater penetration of the object inside the hand (5.15.4). In both, this result and those obtained in the mug and Pringles can, the method shows a notable ability to learn effective grasps even in the real environment and using less number of iterations, with 70% of successful grasps in the total obtained, as indicated in Tab. 5.5.

6 Conclusion

At the end of this thesis, **the research has successfully achieved its goal of testing the effectiveness of Bayesian optimization** in the context of grasping diverse and previously unknown objects in a robust and secure manner, using haptic exploration and bridging the gap between analytic and data-driven approaches. By conducting comprehensive experiments in both simulated and real robotic environments, we have gained **valuable insights into the adaptability and performance of this approach under real-world conditions**, accounting for uncertainties and technical complexities. Furthermore, our **investigation into various grasp quality metrics** has shed light on their practical utility when estimated from a real system, including a three-fingered hand with tactile sensors on the fingertips, and how their behavior affects the results obtained in Bayesian optimization.

One of the main advantages observed in this method is that it is based on the optimization of a general black-box function. This has allowed the **addition of different heuristics to the grasp evaluation function** to make the method capable of dealing with diverse objects, end-effectors, and environmental conditions, and the addition of the Light Weight Robot to the robotic system, **without compromising the number of iterations needed to optimize the grasp**. Among these heuristics, a variation to the existing **Collision Penalty CP** has been proposed to force the search algorithm to move away from possible collision configurations, which has led to a **noticeable improvement in the method**. A **Contact Reward CR** has also been developed, to bring the search closer to object grasping configurations, and other considerations based on robot kinematics have been included, **accelerating the convergence of the method** even when working on the real system. This has given evidence of the variable generalization of the method, able to adapt to different environments depending on the evaluation function, being computationally light and without the need for training data.

Furthermore, the experimental process has allowed us to obtain a number of interesting conclusions in several aspects. The comparison of the performance of each metric in the evaluation function carried out in simulation gave rise to various insights about their ability to classify the grasps and obtain an optimal one. It has been shown that Q_{iso} and Q_{uni} **obtain a higher range of values than Q_c and Q_v due to how they are**

defined, favoring the convergence of Bayesian optimization. By testing optimal grasps obtained in simulation on the real system, it has been possible to verify the behavior of these analytic grasp metrics in the presence of noise and uncertainty. Evidence has been given of the **limitations of the force closure condition** to determine the validity of a grasp in the presence of positional errors and when facing a dynamic environment with an unknown object. The sensitivity to this force closure error of metrics such as Q_ϵ and especially Q_v has also been shown, leading to failed grasps with a spurious high outcome. In particular, the use of **the grasp wrench space GWS to classify the grasp has been proven to have a considerable error due to the assumptions** chosen about the center of mass of the object and the uncertainty of the measurements taken for its calculation. Even so, **Bayesian optimization has been shown to deal positively with these difficulties**, achieving a **70% success** rate in the optimization of the grasp performed completely on the real robot. The advantages and disadvantages of each of the metrics have been compensated by the **use of a combined metric** in this final phase of the experimentation, being able to observe a greater variety in the obtained grasps in objects such as the Pringles can. This experiment has also highlighted the **importance of the initial samples to obtain a good result**, being essential to study methodologies to improve this preliminary step in order to enhance the Bayesian optimization.

The type of hand chosen for the experimentation has also had interesting consequences on the results obtained. The metrics computed from the contact points in the CLASH fingertips have resulted in **grasps with less penetration of the object inside the hand** and more symmetrical distribution of the contacts, while Q_{uni} has obtained **grasps with a larger contact surface** in the fingers, reflecting the **loss of information due to not considering the contacts in the proximal phalanges**. In addition, when working with a hand of variable stiffness, conventional **analytical metrics become more inaccurate** as they do not take into account positional errors due to differences in tendon pretension. On the other hand, the **advantage of using a variable stiffness hand in the generalization of the method** has been observed, since its great adaptability allows them to avoid slipping and to grasp curved surfaces robustly.

This work provides practical knowledge and a first approach to the application of Bayesian Optimization to tactile exploration of grasps in a real environment, so that the techniques applied for its operation in an arm-hand system, as well as the conclusions drawn from the experimentation, provide a novel contribution to the state of the art in this study.

6.1 Limitations and future work

Given the research nature of the thesis and the promising results achieved by the proposed methodology, showing a great performance even in the presence of noise and uncertainty introduced by the real environment, the lines open for future work are broad. There are still challenges to overcome in order to bring Bayesian optimization to real applications and daily living or industrial environments. To address these challenges some of the limitations experienced in this work and the proposed future work to study them are included in this list:

- Assuming unit forces at the contacts, which facilitates the use of the selected hands for experimentation, reduces the precision of metrics based on grasp wrench space, especially Q_e , which loses part of its meaning as it relies on force magnitude. Future work should involve the calibration and utilization of sensors capable of measuring force magnitude, not just direction, being distributed across the entire finger to avoid information loss in grasping.
- Assuming the grasped object is static restricts the method's adaptability to the real environment. A potential improvement in future work involves incorporating object pose detection and tracking through computer vision techniques.
- The method does not consider the variable stiffness of the hand or its ability to adapt to different grasps. In future work, grasp parametrization should be reconsidered, taking into account critical parameters like tendon pretension to enhance adaptability to objects with varying characteristics and prevent hand damage.
- Similar to the parametrization, the metrics used do not account for important hand characteristics such as its underactuated design and variable stiffness. The study of metrics considering underactuated joints and including the tendon pretension in their calculations should be addressed in future work.
- The force closure condition has significantly reduced the method's performance. Future work needs to investigate this condition to develop classification algorithms that consider hand stiffness and underactuated joints, reducing dependence on factors like the location of the center of mass. Supervision during optimization may also be explored to prevent such errors.
- Another path to follow in future research is exploring the integration of this work with the state-of-the-art Bayesian optimization-based shape reconstruction techniques. This could lead to a more comprehensive method with outstanding

performance and adaptability in real-world environments, also addressing the limitation introduced by the lack of knowledge of the center of mass of the object.

- As stated during the experimental results it would be interesting to investigate the computation of initial samples of the optimization and their influence on method convergence, potentially guiding the method towards obtaining optimal grasps for specific tasks.
- Finally, to demonstrate the validity of the method in future work, it would be important to conduct an experiment optimizing the grasp of different objects in a real environment using a six-dimensional unrestricted search space (position and orientation).

List of Figures

2.1	An example of using Bayesian optimization on a toy 1D design problem. The figures show a GP approximation of the objective function over four iterations of sampled values of the objective function. The figure also shows the acquisition function in the lower-shaded plots. The acquisition is high where the GP predicts a high objective (exploitation) and where the prediction uncertainty is high (exploration). Note that the area on the far left remains unsampled, as while it has high uncertainty, it is (correctly) predicted to offer little improvement over the highest observation [9].	8
2.2	Radius of the largest ball centered at the origin of the wrench space and fully contained in \mathcal{P} (Epsilon measure, adapted from [23])	14
2.3	Comparison between the Largest minimum resisted wrench Q_e (left) and the Volume of the GWS Q_v (right) metrics of a grasp with 4 contacts [23].	15
3.1	Graphical representation of the search space (bounding box).	18
3.2	Optimization process scheme. Integration of the grasping process in each iteration.	19
3.3	Bounding box limits. Difference between considering the end-effector pose and the hand-palm pose.	20
3.4	Experimental setups: A) Hand in free space; B) Full robot model simulation; C) Real environment.	21
3.5	Left: Capability map for the Kuka LBR arm, with similar kinematics to DLR's LWR [34]; Right: 3D representation of the hand palm (dark gray) in a reachable position (R: 0.92) inside the capability map	22
3.6	Example of arm motion from pre-grasp pose to target pose.	25
3.7	Force closure condition adapted from [37]. The wrench set $W = \{\omega_1, \omega_2, \omega_3\}$ (CH represented in discontinuous lines) is non-FC. The wrench set $W^* = \{\omega_1, \omega_2, \omega_4\}$ (CH represented in continuous lines) is a FC grasp with d_{min} represented in red.	27
4.1	PyBullet example of a complex simulation environment [39].	29
4.2	General scheme of the active grasping process implemented in Links and Nodes.	33

List of Figures

4.3	ReFlex TakkTile by Right Hand Robotics (taken from [41])	34
4.4	A diagram showing the location of the ReFlex TakkTile’s sensor concerning the base of a finger. [42]	34
4.5	CLASH (Compliant Low-cost Antagonistic Servo Hand) (taken from [45])	35
4.6	Kinematic optimization of the length of palm and finger segments (taken from [44]).	36
4.7	Mechatronic structure of the CLASH 3F hand (taken from [44]). (right) Thumb module. (left): Differential finger module.	37
4.8	XELA uSkin Patch 4x4 sensor scheme	38
4.9	3D representation of the XELA uSkin Patch 4x4 (taken from [47]). A) Silicone membrane; B) Force direction in the 3 Cartesian axes.	39
4.10	Light Weight Robot III on the laboratory bench.	40
4.11	LWR III joint unit (taken from [49]): joint components scheme (left), real joint unit in carbon fiber (right).	41
4.12	YCB objects set (taken from [50]).	42
5.1	Evolution of the quality of the grasps obtained in the mustard bottle and power drill. Left: Grasps obtained with CLASH. Right: Grasps obtained with ReFlex TakkTile.	46
5.2	Comparison of the evolution of the quality of the grasps obtained using different implementations of the grasping process (Q_{grasp}).	48
5.3	Mean time to convergence and number of not converged experiments per object (mustard bottle, mug, and power drill) and metric (Q_{iso} , Q_{ϵ} , Q_v , Q_{uni}).	49
5.4	Some of the best grasps obtained with each metric for the mustard, the mug, and the power drill. From left to right: Q_{iso} , Q_{ϵ} , Q_v and Q_{uni}	50
5.5	Left: experimental setup including the ReFlex TakkTile hand in a static position and a water bottle to grasp. Right: Evolution of the outcome when optimizing the force exerted by the ReFlex TakkTile fingers on the bottle.	51
5.6	Comparison of the evolution of the quality of the grasps obtained using different grasp metrics to grasp the Pringles can.	53
5.7	Some of the grasps obtained with each metric for the Pringles can. From left to right: Q_{ϵ} , Q_{iso} , Q_{uni} and Q_v . The images at the top show successful grasps, while those at the bottom show failed grasps.	54
5.8	Comparison of the evolution of the quality of the grasps obtained using different grasp metrics to grasp the mug.	55
5.9	Some of the grasps obtained with each metric for the mug. From left to right: Q_{ϵ} , Q_{iso} , Q_{uni} and Q_v . The images at the top show successful grasps, while those at the bottom show failed grasps.	56

List of Figures

5.10	Comparison of the evolution of the quality of the grasps obtained using different grasp metrics to grasp the Lego toy.	56
5.11	Some of the grasps obtained with each metric for the Lego toy. From left to right: Q_c , Q_{iso} , Q_{uni} and Q_v . The images at the top show successful grasps, while those at the bottom show failed grasps.	57
5.12	Some of the grasps performed with CLASH on mustard bottle and power drill. 1: failed clamp-like grasp on the mustard; 2: failed clamp-like grasp on the power drill; 3: successful grasp surrounding the power drill head.	58
5.13	Some of the grasps performed with CLASH on Pringles can and mug. .	60
5.14	Comparison of the evolution of the quality of the grasps obtained grasping the Pringles can and the mug with the real robot.	61
5.15	Some of the grasps performed with CLASH on	61

List of Tables

4.1	Objects of the YCB object and model set used for the experimentation. Object IDs are consistent with Calli et al. [50].	43
5.1	Free hand simulation: Mean and maximum outcomes	45
5.2	Full model simulation: Mean and maximum outcomes	48
5.3	Full model simulation: metrics convergence	49
5.4	Experiment 1: Mean outcomes and number of successful grasps obtained using different metrics in the real environment.	52
5.5	Experiment 2: Mean outcomes and number of successful grasps	59

Bibliography

- [1] M. T. Mason, "Manipulator grasping and pushing operations," 1982.
- [2] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, IEEE, vol. 2, 2003, pp. 1824–1829.
- [3] 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.
- [4] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *2015 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2015, pp. 1316–1322.
- [5] S. Joshi, S. Kumra, and F. Sahin, "Robotic grasping using deep reinforcement learning," in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, IEEE, 2020, pp. 1461–1466.
- [6] 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.
- [7] S. Dragiev, M. Toussaint, and M. Gienger, "Gaussian process implicit surfaces for shape estimation and grasping," in *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 2845–2850.
- [8] J. Mahler, S. Patil, B. Kehoe, J. Van Den Berg, M. Ciocarlie, P. Abbeel, and K. Goldberg, "Gp-gpis-opt: Grasp planning with shape uncertainty using gaussian process implicit surfaces and sequential convex programming," in *2015 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2015, pp. 4919–4926.
- [9] E. Brochu, V. M. Cora, and N. De Freitas, "A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," *arXiv preprint arXiv:1012.2599*, 2010.
- [10] M. Leco, T. McLeay, and V. Kadiramanathan, "A two-step machining and active learning approach for right-first-time robotic countersinking through in-process error compensation and prediction of depth of cuts," *Robotics and Computer-Integrated Manufacturing*, vol. 77, p. 102345, 2022.

- [11] J. Nogueira, R. Martinez-Cantin, A. Bernardino, and L. Jamone, "Unscented bayesian optimization for safe robot grasping," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2016, pp. 1967–1972.
- [12] J. Castanheira, P. Vicente, R. Martinez-Cantin, L. Jamone, and A. Bernardino, "Finding safe 3d robot grasps through efficient haptic exploration with unscented bayesian optimization and collision penalty," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 1643–1648.
- [13] M. Toussaint, "The bayesian search game," in *Theory and Principled Methods for the Design of Metaheuristics*, Springer, 2013, pp. 129–144.
- [14] J. Garcia-Barcos and R. Martinez-Cantin, "Fully distributed bayesian optimization with stochastic policies," *arXiv preprint arXiv:1902.09992*, 2019.
- [15] C. De Farias, N. Marturi, R. Stolkin, and Y. Bekiroglu, "Simultaneous tactile exploration and grasp refinement for unknown objects," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3349–3356, 2021.
- [16] I. Herrera Seara, J. Garcia-Lechuz Sierra, J. Garcia-Barcos, and R. Martinez-Cantin, "Optimizacion bayesiana multisolucion para la exploracion eficiente de agarres roboticos," in *XLIII Jornadas de Automatica*, Universidade da Coruña. Servizo de Publicacions, 2022, pp. 714–720.
- [17] J. K. Salisbury and J. J. Craig, "Articulated hands: Force control and kinematic issues," *The International journal of Robotics research*, vol. 1, no. 1, pp. 4–17, 1982.
- [18] 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, 1988.
- [19] C. Ferrari and J. F. Canny, "Planning optimal grasps.," in *ICRA*, vol. 3, 1992, p. 6.
- [20] C. Borst, M. Fischer, and G. Hirzinger, "A fast and robust grasp planner for arbitrary 3d objects," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, IEEE, vol. 3, 1999, pp. 1890–1896.
- [21] Y.-H. Liu, "Qualitative test and force optimization of 3-d frictional form-closure grasps using linear programming," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 1, pp. 163–173, 1999.
- [22] 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)*, IEEE, vol. 2, 2001, pp. 949–956.
- [23] M. A. Roa and R. Suárez, "Grasp quality measures: Review and performance," *Autonomous robots*, vol. 38, pp. 65–88, 2015.

- [24] B. León, C. Rubert, J. Sancho-Bru, and A. Morales, "Characterization of grasp quality measures for evaluating robotic hands prehension," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 3688–3693.
- [25] A. K. Goins, R. Carpenter, W.-K. Wong, and R. Balasubramanian, "Evaluating the efficacy of grasp metrics for utilization in a gaussian process-based grasp predictor," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2014, pp. 3353–3360.
- [26] R. Krug, A. J. Lilienthal, D. Kragic, and Y. Bekiroglu, "Analytic grasp success prediction with tactile feedback," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 165–171.
- [27] C. Rubert, D. Kappler, J. Bohg, and A. Morales, "Predicting grasp success in the real world—a study of quality metrics and human assessment," *Robotics and Autonomous Systems*, vol. 121, p. 103274, 2019.
- [28] A. Morales, E. Chinellato, A. H. Fagg, and A. P. Del Pobil, "Using experience for assessing grasp reliability," *International Journal of Humanoid Robotics*, vol. 1, no. 04, pp. 671–691, 2004.
- [29] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global optimization*, vol. 13, no. 4, p. 455, 1998.
- [30] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.
- [31] R. Martinez-Cantin, "Bayesopt: A bayesian optimization library for nonlinear optimization, experimental design and bandits.," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3735–3739, 2014.
- [32] C. Rubert, B. León, A. Morales, and J. Sancho-Bru, "Characterisation of grasp quality metrics," *Journal of Intelligent & Robotic Systems*, vol. 89, pp. 319–342, 2018.
- [33] A. M. Sundaram, O. Porges, and M. A. Roa, "Planning realistic interactions for bimanual grasping and manipulation," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, IEEE, 2016, pp. 987–994.
- [34] O. Porges, T. Stouraitis, C. Borst, and M. A. Roa, "Reachability and capability analysis for manipulation tasks," in *ROBOT2013: First Iberian Robotics Conference: Advances in Robotics, Vol. 2*, Springer, 2014, pp. 703–718.
- [35] S. R. Buss and J.-S. Kim, "Selectively damped least squares for inverse kinematics," *Journal of Graphics tools*, vol. 10, no. 3, pp. 37–49, 2005.

- [36] D. Q. Huynh, "Metrics for 3d rotations: Comparison and analysis," *Journal of Mathematical Imaging and Vision*, vol. 35, pp. 155–164, 2009.
- [37] M. A. Roa and R. Suárez, "Geometrical approach for grasp synthesis on discretized 3d objects," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2007, pp. 3283–3288.
- [38] *Documentation | bullet real-time physics simulation*, <https://pybullet.org/wordpress/index.php/forum-2/>, Accessed: 2023-06-29.
- [39] erwincoumans, *Pybullet_robots*, https://github.com/erwincoumans/pybullet_robots, Accessed: 2023-06-29.
- [40] F. Schmidt, *Links_and_nodes master documentation*, https://rmc-github.robotic.dlr.de/pages/common/links_and_nodes/, Accessed: 2023-06-29.
- [41] *Reflex hand*, <https://www.labs.righthandrobotics.com/reflexhand>, Accessed: 2023-06-26.
- [42] K. Meyere, "Model making and haptic controller design for the reflex tactile," M.S. thesis, University of Twente, 2018.
- [43] Y. Tenzer, L. P. Jentoft, and R. D. Howe, "Inexpensive and easily customized tactile array sensors using mems barometers chips," *IEEE R&A Magazine*, vol. 21, no. c, p. 2013, 2012.
- [44] W. Friedl and M. A. Roa, "Clash—a compliant sensorized hand for handling delicate objects," *Frontiers in Robotics and AI*, vol. 6, p. 138, 2020.
- [45] *Dlr / clash*, <https://www.behance.net/gallery/70399591/DLR-clash>, Accessed: 2023-09-13.
- [46] *Uspa 44 | patch sensor series*, <https://www.xelarobotics.com/sensor-collection/uspa-44>, Accessed: 2023-06-26.
- [47] *Tattile sensors: Our technology*, <https://www.xelarobotics.com/tactile-sensors>, Accessed: 2023-09-13.
- [48] G. Hirzinger and A. Albu-Schaeffer, "Light-weight robots," *Scholarpedia*, vol. 3, no. 4, p. 3889, 2008.
- [49] *Lwr iii*, <https://www.dlr.de/rm/en/desktopdefault.aspx/tabid-12464/#gallery/29165>, Accessed: 2023-06-27.
- [50] 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," *IEEE Robotics and Automation Magazine*, vol. 22, no. 3, pp. 36–52, 2015. doi: 10.1109/MRA.2015.2448951.

Bibliography

- [51] J. Weisz and P. K. Allen, "Pose error robust grasping from contact wrench space metrics," in *2012 IEEE international conference on robotics and automation*, IEEE, 2012, pp. 557–562.