Sample-Efficient Search for Reactive Grasping using Fingertip Force/Torque Sensors

Master's thesis in Systems, Control and Mechatronics

ZEID AL IDANI

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2022
www.chalmers.se

MASTER'S THESIS 2022

Sample-Efficient Search for Reactive Grasping using Fingertip Force/Torque Sensors

ZEID AL IDANI

Department of Electrical Engineering
Systems and Control Division
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2022

Sample-Efficient Search for Reactive Grasping using Fingertip Force/Torque Sensors
ZEID AL IDANI

Supervisor: Yasemin Bekiroglu, Department of Electrical Engineering
Examiner: Yiannis Karayiannidis

Master's Thesis 2022
Department of Electrical Engineering
Systems and Control Division
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2022

Sample-Efficient Search for Reactive Grasping using Fingertip Force/Torque Sensors
ZEID AL IDANI
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Success of a grasping and manipulation system depends on various factors, starting from the perception and planning stage, to the end of a task completion. Robustness during grasp execution is crucial for successful systems, which can be highly effected by many sources of uncertainty inherent in real-world settings, such as imperfect pose estimation, unknown friction or deformability properties of objects. Most grasping approaches mainly only apply predefined fixed forces during grasp execution based on simplifying assumptions on the objects, i.e., that the forces would not cause damage or failure. However, this cannot be guaranteed in the case of unknown objects. This thesis addresses the issue of grasping objects with unknown features and is focused on increasing robustness in grasp execution using real sensory data, i.e., force/torque readings.

The proposed sample-efficient approach to robust grasp execution includes a self-learning controller with an updating reference for increasing grasp success rates. Bayesian optimization is used for sample-efficient search, which allows for finding good grasp control parameters in a small number of trials using a real robot. The experiments were performed on a real robot with four different objects with different weights an the results showed that the proposed approach can be successfully applied to grasp execution, where unknown objects can be grasped without any damage or slippage while passing stability tests.

Keywords: pick and place, grasping, force/torque sensors, bayesian optimization.

Acknowledgements

I would like to give thanks and gratitude to my supervisor Yasemin Bekiroglu for her support and for pushing to work harder than I ever thought I was capable of. I will carry this experience with me not just to future work and studies, but to the rest of my life.

I would also like to give a big thanks to Hadi Hadjigari for his help and support during this project. The fun I have had with Hadi during stressful times, discussing robotics, physics, politics and whether barberries are in fact berries, will not be forgotten.

Lastly, I would like to thank Yiannis Karayiannidis for giving great feedback and sharing his expertise.

Zeid Al idani, Gothenburg, June 2022

Contents

List of Figures

List of Tables

1
Introduction

The task of grasping and manipulating objects requires meticulous orchestration of planning, gripping, lifting, moving and placing the object all while keeping the grasp sturdy and safe. This is even more true for grasping novel objects where weight, texture, rigidness and other object properties are not known beforehand. Humans master this ability at a young age and can with great dexterity manipulate an object without prior information while being little aware of underlying mechanisms. It is still a long way before robots can match grasping capabilities of humans. However, robot grasping and object manipulation has improved significantly since the early works in 1980s [1]. With the surge in machine learning, algorithms for grasp planning and grasp execution have become increasingly more successful. This is true not only for grasping familiar objects but also for novel objects [2]. Even so, robot grasping of novel objects still lack robustness. An initially good grasp can still fail due to e.g. imperfect calibration, unknown object friction, weight distribution or deformation properties, interaction dynamics etc. Grasp planning and execution should be performed taking various sources of uncertainties in real systems into account to avoid failures. But in most robotic grasp systems, grasp planners rely heavily on visual input to plan for a grasp. After a suitable grasp is generated, it is executed without feedback. By using other sensory modalities, such as tactile sensors, the stability of a grasp can be assessed during a grasp and adaptations can be applied thereafter to increase system robustness.

The development and incorporation of tactile sensors in robot grasping systems has accelerated the past two decades [3]. There are various approaches that focus on estimating the stability of a grasp with tactile sensory feedback [4], [5], [6]. When a grasp is generated and executed, readings from the sensors in the gripper are used to predict if a grasp is stable or not. This information can in turn be used to generate and execute a new grasp if the current grasp is deemed likely to fail. There are also approaches that applied similar machine learning methods for correcting the grasp once it is predicted as unstable [7], [8], [9]. But choosing a high-quality grasp initially does not guarantee a stable grasp throughout grasp execution. Adapting the grasp continuously with a controller ensures stability during full execution.

However, this approach of including feedback in the system throughout the grasp execution raises the question of how to design the controller for processes that are not known beforehand, have nonlinear dynamics that cannot be ignored or processes that have extensive noise. The challenge is therefore to deal with these issues and design a controller that ultimately increases the robustness of a robot grasp system by making it reactive.

1.1 Aim

The purpose of this thesis is to design and evaluate an approach for dealing with uncertainty in robotic grasping of never-before seen objects. The approach is, more specifically, concerned with the inclusion of an automatically tuned force controller. It is implemented and tested in a real-life robotic grasping system which includes force/torque sensors for tactile measurements.

1.2 Problem description

The starting point of this work is a scenario where a planned grasp configuration is executed for a novel object. In order to have a robust system where the grasp execution leads to a successful manipulation such as lifting up an object without the object slipping, there needs to be 1) detection of any change in grasp stability during grasp execution of the grasp stability and 2) corrective actions when needed.

Detecting disturbances of a stable grasp of an object with only sensory data comes with challenges. Noise in the sensors would make it difficult to differentiate between actual change in the stability of a grasp and non-important signal variations. Considering that we are also dealing with objects with unknown features, the physical characteristics of those objects may cause different levels of noise. For instance, some object may be more reflective than others and using optical sensors to detect contact would give a noisier signal for some objects than others.

Reacting to an unstable grasp of an object with feedback controllers also comes with difficulties. The optimal control-law for achieving grasp stability differs from object to object. The control-law should also be chosen such that it allows for smooth movements to not disturb the object and cause instability. But how is the optimal control input adapted to different object with different characteristics that are unknown a priori? This question brings us to the main questions in the thesis:

- ***How could a feedback controller best be designed for processes that are not known a priori, that have important nonlinear dynamics that cannot be ignored and for processes that have extensive noise?***
- ***How could information about a current unstable grasp be used to take reactive corrective actions with an adaptive controller to increase grasp stability without deforming the object?***
- ***How could the feedback controller be tuned automatically in an efficient way?***

1.3 Related work

A comprehensive review of the research in the field of using tactile sensory information to perform a successful grasping can be found in [10]. In the rest of this section, a presentation of related work in this field is given.

Previous work [11] studied how to design a framework for controlling the position, velocity, torque, and force of a gripper. They simplify the problem by linearizing the model of a Barrett-hand and designing P- and PD-controller to follow a predefined reference trajectory with feedback from tactile sensors and joint angles of the robotic hand. They extend this work in [12] by proposing a method for learning successful grasps from human demonstration. The key idea is to choose the right approach not only using perceptual data but also through experience. The demonstration data is used to construct a map from the recognized object shapes to the matching grasp strategy. Then, a P-controller with feedback from tactile sensors is used to lead the gripper to a stable grasp. [12] mainly focuses on a method for generating a suitable approach vector towards the object based on demonstrations.

Romano et al. [9] draw inspiration from human somatosensory system and proposed an approach to control the grasp during its entire cycle. They filter tactile sensory signals so that they mimic the signals from different touch receptors in human, and then use them to detect disturbances in the grasp. The filtered signals are used to regulate the forces applied on the object. A strategy is devised for controlling the applied forces by empirically tuning the controller parameters for a set of real-world objects. The drawback of their approach is that finding the optimal set of parameters for controlling the grasp of a wide range of real world objects empirically is expensive and time-consuming. Furthermore, in case of novel objects, the applied force can be too weak or too strong, which could lead to either slippage or deformation of the object.

Li et al. [7] apply a probabilistic method to deal with inherent uncertainties of grasping novel objects. They devise a strategy to predict the grasp stability with tactile sensory data and then choose the configuration of a superior grasp. A low impedance controller is used to guide the gripper to a more stable grasp. However, the authors report that the parameters of the controller are hard to tune in practice which could make the movements of the gripper fraught. This could lead to, as they argue, an unstable grasp.

In [13], Hyttinen et al. use a trained model to predict the quality of a grasp attempt based on tactile/proprioceptive data and partially reconstruct 3D model of the object. These features are used to calculate the probability of success for candidates of grasping attempt, then the algorithm chooses the plan with the highest probability of resulting in a stable grasp and executes it. Furthermore, they exploit the tactile/proprioceptive data to fine-tune the action even at the last stage of preparing to grasp the object in order to improve the chance of success. The algorithm is shown to be effective in planning successful grasp of known as well as novel objects. However, the success of the attempts solely depend on the initial decision and planning, and any unforeseen dynamics, such as slippage, and external disturbance during the execution may ultimately result in the failure of the grasp.

Krug et al. [6] propose using data from tactile sensors, joint configuration, and object information to estimate the stability of a grasp before lifting the object. They take inherent uncertainties of the object into consideration by employing probabilistic techniques for classifying the grasp stability. Notably, even after grasping the object if the data from tactile sensors indicate that the grasp may be unstable the planner generates a new candidate and the object is re-grasped. Their study shows that tactile signatures from a grasp carry valuable information which is valuable to decide about grasp stability. The strength and limitations of such a probabilistic method is examined further in [14] where a framework based on long-term memory and reasoning modules is presented. They explore how a robot can take learn from its experiences in a long run and utilize its experience to enhance its performance within a Bayesian Optimization framework.

She et al. [15] use a controller with feedback from tactile sensors for a reactive gripper that successfully follows a cable. They design a PD controller combined with a leaky integrator to adjust the applied force on the cable, and a Linear Quadratic Regulator (LQR) to keep the cable centred in the gripper. They estimate the dynamics of the cable empirically and devise a linear model on which the controllers are based. This is a special-purpose methodology; using it to design controllers for robotic grasping of wide-range of novel objects is impractical. Even a linear model, which is based on a small set of objects, will most likely result in designing a controller that performs poorly in dealing with different novel objects [16].

2
Background

This chapter gives an introduction to robotic grasping systems (section 2.1) with a brief overview of different types of grippers (subsection 2.1.1) and sensors (subsection 2.1.2). A background on Linear Quadratic Regulators is given (section 2.2) with the machine learning techniques that are relevant in this thesis (section 2.3).

2.1 Robotic grasping system

A grasping system designed for a pick and place task will in its simplest form contain a robotic arm and some kind of gripper. In this type of setup, the objects will need to be known before hand and placed in the workspace with the same initial pose every time. Sensory input is required for more sophisticated system designed for more cumbersome tasks. In, for example, bin picking the objects are perhaps unordered and placed in different orientations which may require a new grasp pose in each grasping occasion. Visual input combined with a grasp planning algorithm is often the preferred solution for this type of task. Figure 2.1 shows an example of a grasping system with vision that can be combined with a grasp planner for bin picking.

Figure 2.1: The figure shows an example of a robotic grasping system setup for bin picking. The system includes vision in addition to an arm and manipulator.

Grasp planners are a highly active area of research with state of the art grasp planners being impressively effective at generating high quality grasps for novel objects [17], [18], [19]. However, even if we ignore the presence of noise and uncertainty in the system and assume a perfect initial grasp, the dynamics of the object could still very well cause grasp failure through, for example, slippage or object perturbation.

The performance of a grasping system can be greatly enhanced by adding sensory feedback during grasp execution. Different types of sensors could be combined with different types of grippers leading the grasping system to react to unexpected events, especially when grasping novel objects.

2.1.1 Grippers

A robotic gripper, or end-effector as it is also called, is the device in a grasping system that is equivalent to a human hand and can come in many different sizes and forms for different applications. Some grippers are vacuum based with a suction-cup end that holds an object by using the difference between atmospheric pressure and vacuum. This type of gripper is often used in an industrial setting with rigid and flat surfaced objects, such as boxes, and is generally not suitable for objects with curved or uneven shapes. For applications that require handling of a wide variety of objects, a multi-fingered gripper is generally more suitable. This class of grippers include devices that vary in complexity ranging from the simpler ones, the two fingered grippers, all the way to the more advanced human-like ones with multiple fingers and joints. The more complex ones are suitable for applications that require dexterity and precision and are generally preferred in current research. However, they have the drawback of being more complex to control and highly costly relative to other types of grippers, making them unfavorable for industrial applications. Figure 2.2 shows a complex five-fingered gripper and a simple vacuum gripper.

(a) **(b)**

Figure 2.2: Two examples of robotic grippers. (a) shows the Epick vacuum gripper by Robotiq™ that moves and holds objects by utilizing the difference between atmospheric pressure and vacuum. The more complex SVH gripper by SCHUNK™ is shown in (b). This type of human-like gripper consists of several motors and actuators that gives it 27 DOF which also makes it costly in comparison with other types of grippers.

For applications that require handling of a wide variety of objects, two or three fingered grippers with one or two degrees of freedom will often be sufficient. Popular grippers in this category is the parallel grippers which, as the name suggests, have two fingers in parallel, as shown in Figure 2.3. There is a low trade-off between being able to handle many different objects and being easy to control for this type of grippers which is what makes them popular. They consists of either pneumatic or electric actuators that open and close the fingers and can easily be equipped with sensors at the fingertips for regulating the movement of the fingers.

Figure 2.3: An example of the Robotiq™ 2F-140 parallel gripper. This type of gripper is popular due to its versatility while in the same time being low-cost and easy to control.

2.1.2 Sensors

The most common sensor classes for robotic grasping are identification-, tactile- and haptic-sensors. Identification sensors such as cameras are usually used in the planning phase of a grasp to detect an object and determine how and where it should be grasped. Tactile and haptic sensors, on the other hand, are used when in contact or near contact with an object and are used to gather information about the object and the state of the grasp. The two terms tactile- and haptic-sensors are in the literature often used interchangeably and refer to classes of sensors such as contact-arrays and force/torque sensors but sometimes also proximity sensors, although they do not rely on physical interaction. [20]

Proximity sensors are most commonly optical based sensors that measure distance by emitting light and tracking the time it takes for it to reflect back. These types of sensors have been utilized in robotic grasping to gain more accurate knowledge about an objects position and pose which has subsequently been used to adjust the grasp-pose, gripper position etc. prior to contact to ensure grasp stability [21], [22]. But they have also been used in research for other purposes such as slip detection during grasp execution [23]. This is why proximity sensors are sometimes classified as tactile sensors in grasping systems [3]. However, proximity sensors for tactile sensing requires high resolution signals with low noise, which in practice is often not the case. Contact-based sensors for are generally preferred for tactile sensing.

F/T sensors are tactile sensors that measure the linear and rotational forces that are exerted on them. The sensors utilize the principle of a strain gauge which register the exerted strain via a change in electrical resistance. Combined with a deformable and elastic component, the sensors register changes in pressure when in contact with an object [24]. The pressure is directly proportional to the applied forces and the output is force and torque measurements.

2.2 The LQR Problem

An essential controller in optimal control theory is LQR – the Linear Quadratic Regulator. In contrast to more basic controllers, for instance variations of the PID-controller, a designed LQR controller guarantees stability in a system without the need for pole-placement analysis while being arguably more intuitive to tune for higher-order systems.

To highlight the advantages of this powerful control method, let us review it by first considering the following noiseless and discrete-time dynamical system

$$x_{k+1} = Ax_k + Bu_k \tag{2.1}$$

with state vector $x_k \in \mathbb{R}^{n_x}$ and input vector $u_k \in \mathbb{R}^{n_u}$. Given an initial state vector x^0, an optimal control input u^* can be found by minimizing the function

$$J = \frac{1}{N} \sum_{k=0}^{N} x_k^T Q x_k + u_k^T R u_k \tag{2.2}$$

with the matrix $Q \in \mathbb{R}^{n_x \times n_x}$ and the matrix $R \in \mathbb{R}^{n_u \times n_u}$. Equation (2.2) is referred to as the quadratic cost function and is the LQR problem with matrices Q and R serving as cost matrices. Penalizing certain states or inputs with a higher cost renders a higher cost function value. Therefore, a natural step is to choose a cheaper control input that does not increase the states associated with a higher cost. It is precisely this feature of tuning Q and R that makes this control method powerful and more intuitive to design compared with other control strategies. A variation of the standard LQR is the Linear Quadratic Tracking Problem

$$J = \frac{1}{N} \sum_{k=0}^{N} (x_k^{ref} - x_k)^T Q (x_k^{ref} - x_k) + (u_k)^T R (u_k) \tag{2.3}$$

The difference between (2.2) and (2.3) is the incorporation of the reference state x_k^{ref}. Here, the objective is to minimize the error between the reference and the states of the system, i.e. tracking the reference with a low control effort. It is useful in applications where following a time-variant trajectory is the objective. Solving (2.2) and (2.3) requires finding a solution to the *Riccati Differential Equation* where it is necessary to have a linear and time-invariant model of the system, as in (2.1) [25]. This is of course not always the case. Often times, linearization means neglecting non-linearities that are important for describing the behavior of the system [26].

In the case of grasping novel objects, it is safe to say that it is impossible to know the proprieties of the object beforehand making it infeasible to build a model of such a system. It is therefore clear that an alternative approach for this type of problem is needed.

2.3 Machine learning

Machine learning (ML) is a branch of artificial intelligence that has had an explosive development the past decade. This has led to significant advances in research areas such as computer vision, autonomous control, imitation learning and self-supervised learning, to name a few. In the field of robotics and autonomous systems, these developments have resulted in a whole new class of solutions to problems that were previously thought as unmanageable. [27]

Machine learning algorithms can largely be divided in supervised, semi-supervised and unsupervised learning algorithms [28]. The unsupervised and semi-supervised learning algorithms use no or little example data to build models while the supervised algorithms learn input-output mapping from a so-called training dataset. There are many different supervised ML algorithms for various modelling problems such as classification and regression. A classification problem may include the need to predict which category an input belongs to, e.g. the need to classify whether an object is a chair or not. Regression problems, however, concerns the mapping of an input to a continuous output. An example of the application of a regression model is predicting the velocity of the wheels of a robot using sensory data for avoiding obstacles [29].

2.3.1 Gaussian Processes Regression

A machine learning tool that is quickly gaining popularity is Gaussian process (GP). It is a powerful machine learning tool used for building models that make predictions of data by incorporating prior knowledge. The probabilistic nature of GP can be used for classification and clustering problems but is widely used for regression problems [30]. However, contrary to traditional regression approaches such as the Bayesian approach where inference of a probability distribution over all possible values for parameters in a function is made, Gaussian process regression instead infers a probability distribution over all possible functions that fit the data. Indeed, this means that it is non-parametric and therefore requires no model assumptions. Furthermore, the stochastic properties of a GP allows it to account for the distribution of noise in the observations, making it useful when dealing with noisy data. Formally, a Gaussian process is a collection of random variables that span over a continuous domain such that the joint distribution of all of these random variables is a multivariate Gaussian distribution:

$$f(x) \sim \mathbb{GP}(m(x), K(x, x')) \tag{2.4}$$

where $m(x)$ is the mean and $K(x, x')$ is the covariance function [30]. The covariance function, or the kernel, describes the relationship between two data points. In effect, this means that (2.4) is a distribution over functions with the shape defined by $K(x, x')$ and with the mean determined by $m(x)$. A regression function modelled as (2.4) will take a set of data points and update the distribution of all possible functions that fit that dataset, i.e. updating the mean and kernel. First, with the initial data points, the probability distribution will be spanning over many possible functions. This *prior* distribution is then used together with new data points to render a *posterior* distribution - an updated model based on new data combined with prior knowledge. The procedure is repeated for all datapoints in the training dataset. A rigorous mathematical description of Gaussian process regression can be found in [31]. Figure 2.4 show how experimental data can be used to build a regression model with this approach.

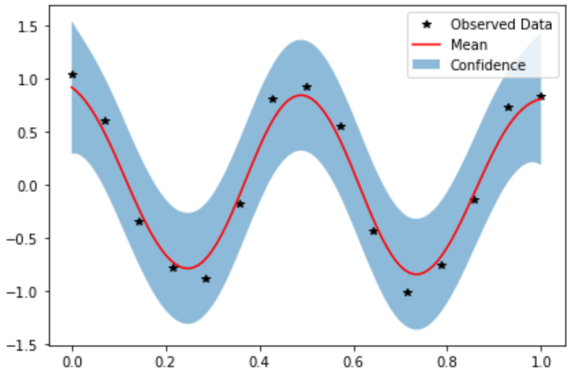


Figure 2.4: A toy example to illustrate how experimental data can be fitted to a Gaussian Process model. The black stars represent observed data points. The blue area represents the confidence of the distribution while the red curve represents the mean value of the distribution.

There are different types of kernels and selecting the right one can be decisive for constructing a useful model. Although there are advantages with other kernels, the most widely used one is the *radial basis function* (RBF):

$$K(G, G'; \sigma, l) = \sigma^2 exp(-\frac{\|G - G'\|^2}{2l^2}) \tag{2.5}$$

with σ^2 being the variance and l is the length scale. The RBF kernel is popular due to its generalization ability and tolerance to input noise [32]. σ^2 and l in RBF are considered as the *hyperparameters* of the *GP* and can be tuned iteratively to fit the training dataset [33].

2.3.2 Bayesian optimization

Bayesian optimization (BO) is an iterative, sample-efficient method for finding the extrema of objective functions that are typically difficult to evaluate. It is applicable in cases where an analytical description of a process is not obtainable but where experimental noisy data is available. The black box objective function, the *surrogate model*, is often times modelled as a Gaussian Process (GP) where BO uses the GP's probabilistic predictions to efficiently localize the optimum of a function [34]. The efficiency comes with the aim of finding possible optima of a function rather than building a complete model of it, as with Gaussian process regression. The algorithm utilizes a so called *acquisition function* to suggests candidates for optimization in an iterative procedure. Figure 2.5 illustrates one iteration in the Bayesian optimization algorithm. The algorithm is commenced by providing an initial best guess which are evaluated with experiments. The output from the experiments is recorded and used to update the surrogate model. With the updated model, a new point of interest is suggested by maximizing the acquisition function. The suggested data point is stored with previous samples and the cycle is repeated. With more and more samples, the surrogate model is improved whereby the likelihood of locating the optima is increased. The algorithm is generally terminated when the distance between the current suggestion and previous suggestion is less than a predefined value. But other termination conditions could also be employed.

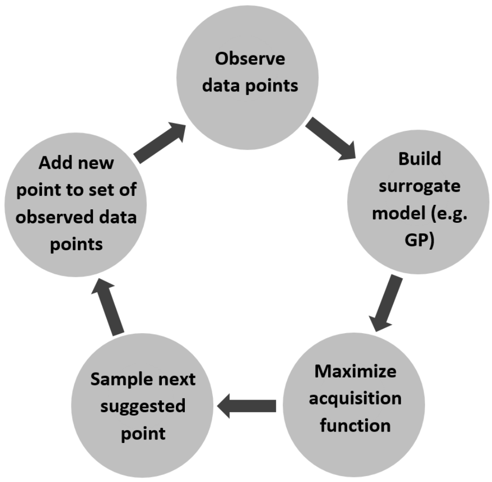


Figure 2.5: The Bayesian Optimization algorithm. The procedure starts with the observation of data points from a black box process. The observed data points are then used to construct a surrogate model, often with Gaussian process regression. This is followed by maximizing an acquisition function to suggest the next input to the black box process. The new input results in a new output which in turn is observed and the cycle starts over.

The acquisition function can be considered as the guide in the BO algorithm. It calculates the probability distribution for the location of the optimal objective function value. Maximizing the acquisition function means locating the next point with the highest probability of corresponding to an optimal objective function value. Figure 2.6 shows an example of a Gaussian process with a corresponding acquisition function in a BO procedure. Different acquisition functions can be selected depending on the strategy for optimization. In some cases, quickly exploring different areas across the objective function is favored. This means that the acquisition function will suggest evaluation points with high uncertainty of residing at the extrema. In other cases, it is more desirable to primarily exploit, meaning that points with high probability of being located at the optima are suggested. In the context of Bayesian optimization, this is known as the *exploring-exploiting trade off* . In practice, choosing the right acquisition function for a specific application is a trial and error process. However, for most scenarios, *probability of improvement* (PI) is a safe first choice of acquisition function. [35], [36]

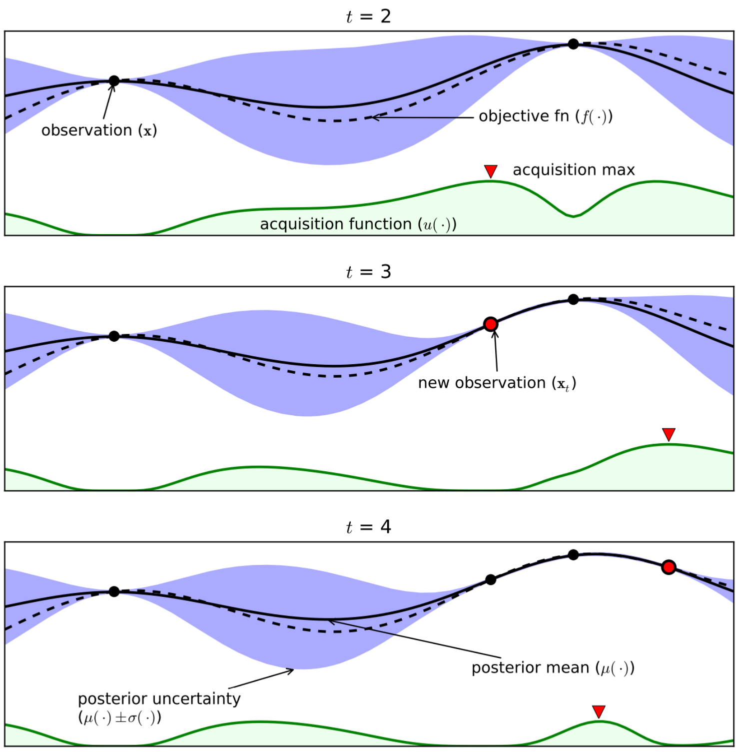


Figure 2.6: A 1-dimensional toy example to illustrate the procedure of the Bayesian Optimization algorithm. The dashed line represents the objective function while the black line represents the mean of the distribution of all possible functions with a Gaussian Process approximation. At t=2, with a distribution constructed with two observations, the optimum of the acquisition function (the lower green area) determines the new input to the function. The output serves as a new observation at t=3, which is used to update the distribution. The procedure is repeated until convergence. [37]

3
Reactive Grasping Solution

In this chapter, a presentation of the proposed approach is given by tying together some of the concepts that are presented in chapter 2. The aim is to address the problem statements highlighted in section 1.3 by first giving an overview of the proposed approach (Section 3.1) and second giving an in-depth presentation of the different aspects of the solution (sections 3.2 and 3.3). The chapter ends by detailing the algorithm for the method in its entirety (section 3.4).

3.1 Outline of the approach

The solution is concerned with a scenario where a grasp pose is determined by a grasp planning algorithm and where the controller is initiated after the gripper is in contact with the object. The proposed approach automatically learns a robust controller using feedback from touch sensors of the gripper. For this, a gripper with torque and force sensors on the tip of its fingers is used. The width and force of the grip of its two fingers are controlled. The proposed solution comprises of two main parts: 1) auto-tuning the controller to deal with novel objects; 2) updating the set-point when the grasp is deemed stable. The diagram in Fig. 3.1 illustrates the proposed solution in full.

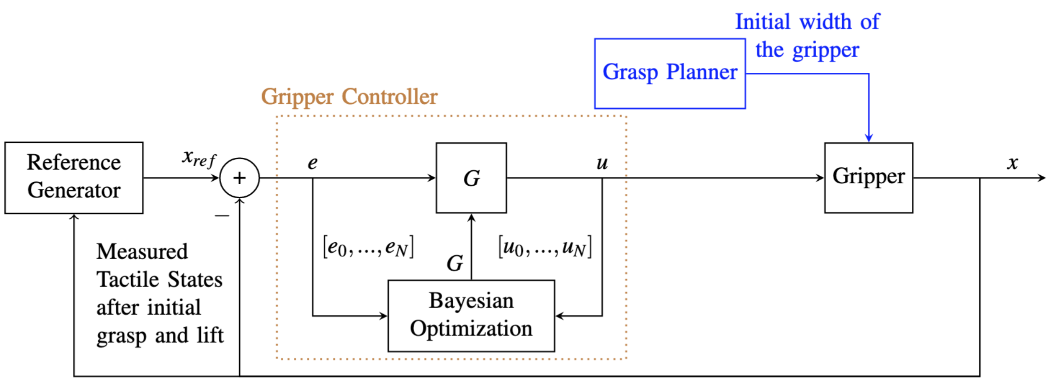


Figure 3.1: Overview of the proposed method for reactive grasping. x is the sensor measurement, and x_{ref} is the set-point. The objective is to minimize the error between the measurements and the calculated set-point. u is the control input. The reference is updated when the grasp is deemed stable. The controller is a linear feedback regulator with gain matrix G. To find the optimal gain matrix in an efficient way, an approach based on Bayesian optimization is proposed.

To auto-tune the controller, a Bayesian optimization (BO) algorithm is utilized together with Gaussian process (GP). The cost function of the tracking problem serves as an objective function modelled as a GP and the objective is thereby to minimize this function. When tuning a controller for a novel object, recording of the measurements in one pick and place iteration is used to evaluate and suggest new parameters for the next iteration. The set of iterations is where the grasping system learns to grasp a novel object. The initial grasp width can be determined by a grasp planner. During grasp execution, measurements from the fingertip tactile sensors, state x in Fig. 3.1, is fed back and compared to a reference x_{ref}. The reference is in turn determined by the stability of the state x. When x is deemed stable in a window of n samples, the reference is updated to be the same as the mean value in the window of x. The error between the reference and the measurements is controlled with a gain matrix G which is optimized with BO.

3.2 Adapting the approach for a parallel gripper with F/T sensors

The reactive grasping approach is adapted for a parallel gripper equipped with six-axis force/torque sensors in each finger, as shown in Figure 3.2, giving a total number of twelve sensor values.

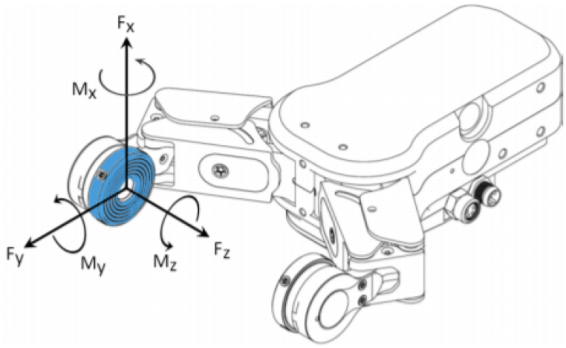


Figure 3.2: A parallel gripper equipped with six-axis force/torque sensors is used for the reactive grasping approach.

To reduce the complexity of the optimization problem, the number of dimensions is decreased by 1) taking the average of the right and left sensor values, 2) combining the forces in x- and y-directions to one tangential force and 3) disregarding the torque around the y- and x-axes, which can be motivated by safely assuming that movement around these axes do not cause grasp failure. This leaves us with a three-dimensional state vector:

$$x = [F_t, F_z, \tau_z] \tag{3.1}$$

where

$$F_t = \sqrt{F_x^2 + F_y^2} \tag{3.2}$$

is the tangential force comprised of the force reading along the $x-$ and $y-axis$. F_z is the normal force and τ_z is the torque around the z-axes. The control input to the parallel gripper is

$$u = width \tag{3.3}$$

where $width$ is the desired gripper width. The dimension of the control input u is thus $n_u = 1$. With the dimension for the state vector x being $n_x = 3$, it gives a dimension for the control parameters G as $n_G = 3$.

3.3 Controller-tuning with Bayesian Optimization

The basis for finding the optimal controller gain for a specific object is the cost function:

$$J = \frac{1}{N}\sum_{k=0}^{N}(x_k - x_k^{ref})^T Q(x_k - x_k^{ref}) + \delta u_k^T R \delta u_k + \rho_k \tag{3.4}$$

which regulates the error between the reference values and the measured states. Here, $x_k \in \mathbb{R}^{n_3}$ is the measurements from the F/T sensors of the gripper, $\delta u_k = u_k - u_{k-1} \in \mathbb{R}$ where u is the control input to the gripper, and Q and R are positive-definite cost matrices. ρ_k is a penalty term that is added whenever the states x_k drop to a mean value of zero, indicating that the object is dropped. Without this penalty, the error between x_{ref} and x may be sufficiently small when dropping some objects and hence give a misleadingly low cost function value. This would of course lead to a poor choice of control action δu_k.

This approach for a reactive grasping solution includes a linear feedback controller of the form

$$\delta u_k = \begin{cases} \epsilon G(x_k - x_k^{ref}), & \text{if } (x_k < x_k^{ref}) \\ G(x_k - x_k^{ref}), & \text{otherwise} \end{cases}. \tag{3.5}$$

where, $G \in \mathbb{R}^{n_G = n_u \times n_x}$ is the control gain matrix and $\epsilon \in [0, 1)$ is a scalar that is included when the error is negative. In practices, this means that the control action for opening the gripper is slower than the action that closes the gripper yielding faster response when perturbation is detected while applying a more conservative ease of the grasp when the object is stable.

The problem is formulated as an optimization problem to minimize the cost function of a quadratic regulator:

$$G = \arg\min_{\tilde{G}} J(\tilde{G}). \tag{3.6}$$

Solving this optimization problem using a classic method requires a model of the system which determines how the input signal, u_k, effects the states of the system, x_k. However, modeling the dynamics of the grasp for fragile and deformable objects is not trivial, as is outlined in section 2.2. To solve this problem, the cost function in Equation (3.4) is modelled as a Gaussian Process (GP):

$$J(G) \sim \mathbb{GP}(m(G), K(G, G'; \sigma, l)), \tag{3.7}$$

where the mean of the process $m(G) = 0$ and where $K(G, G'; \sigma, l)$ is the kernel of the GP. The kernel is chosen as the Radial Basis function described in section 2.3.1. The hyper-parameters are optimized using the Broyden–Fletcher–Goldfarb–Shanno algorithm [38].

If the cost function in Equation (2.3) is modeled as a GP, the control input u and the error between x_{ref} and x are the inputs of the model; and, the output is therefore the value of the cost function J. In the context of Bayesian optimization, the probabilistic model over the cost function J is used for sample-efficient exploration for gain parameters. For this, a suitable acquisition function is needed. By selecting an appropriate acquisition function, the trade-off between exploration and exploitation can be influenced. In this thesis, "Probability of Improvement" (PI) is chosen as the acquisition function to suggest the next set of parameters to evaluate on the robot. The algorithm is initiated by a manual suggestion for the controller parameters G for the first iteration. Then, N samples of x are recorded for a pre-defined pick and place task and use the chosen parameters to evaluate Equation (2.3). The trial provides input to the Bayesian optimization algorithm to render a new set of parameters. The next iteration will be initiated with this set of parameters and will result in a better set according to the optimization cost function. This iterative way of finding optimal parameters will be referred to as the *training session*. The *training session* is terminated after M iterations.

3.4 Updating the Reference

During grasp execution, the measurements from the tactile sensors can show different values depending on the way the object is being grasped; that is if a sudden weight change has occurred when the object is picked up from rest or if changes in the object itself have occurred. Having a fixed reference could lead to a feedback error even if the grasp is stable. For this reason, updating the reference is crucial. However, the controller should also be able to adapt the grasp to ensure that the object is not deformed. Therefore, taking into consideration the need for an error in one or multiple states is also important. When a grasp is stable enough to keep the object from perturbations, the measurements from the tactile sensors will be relatively constant over time. This behavior is exploited by devising a solution for updating the reference for the states that register disturbance. Meanwhile, updating the reference is avoided for the states that register pressure on the object. This would lead to a feedback error for the concerned states and subsequently adapting the grasp applying less forces. The solution for updating the reference is based on linear regression. A line is fitted to a window with n samples of x and the slope is registered. If the slope is less than predefined tolerance, κ, the reference is updated with the average value of the fitted line.

3.5 Procedure summary

The reactive grasping approach is summarized in its entirety in Algorithm 1.

Algorithm 1 Training algorithm for reactive grasping controller.

Require:
Initial controller gain: G_{init}. Weight matrices: Q and R. Number of training iterations: M. Number of samples taken during one training iteration: N. Number of samples to confirm change of the reference values: n. Acquisition function: α. Tolerance for the slope: κ.

repeat
  Close the gripper gradually;
  if x starts changing **then**
    Monitor Sensor Measurements
  end if;
until Sensor Measurements, x, Stabilize;
Initialize:
  $G \leftarrow G_{init}$
  $e_{seq}, \rho_{seq}, \Lambda, \Omega \leftarrow \emptyset$
for iteration = 1 to M **do**
  $X \leftarrow [x_1 \dots x_n]$ ▷ Record window of n measurements
  $x_{ref} \leftarrow average(X)$
  $\rho \leftarrow 0$
  while $length(e_{seq}) < N$ **do**
    $x_{ref} \leftarrow$ UpdateReference(X, x_{ref})
    if $x = 0$ **then**
      $\rho \leftarrow$ Penalty value
    end if
    $u \leftarrow$ update according to Equation (3.5)
    $e_{seq} \leftarrow e_{seq} \cup e$ ▷ Record the error
    $\rho_{seq} \leftarrow \rho_{seq} \cup \rho$
  end while
  $G \leftarrow$ ParameterSearch(e_{seq}, G, ρ_{seq})
end for
function UpdateReference(X, x_{ref})
  slope $\leftarrow$ Linear Regression of X
  if $slope < \kappa$ **then**
    $x_{ref} \leftarrow average(X)$
  end if
  return x_{ref}
end function
function ParameterSearch(e_{seq}, G, ρ_{seq})
  $J \leftarrow \frac{1}{N} \sum_{k=0}^{N} e_{seq,k}^T Q e_{seq,k} + (G \cdot e_{seq,k})^T R (G \cdot e_{seq,k}) + \rho_{seq,k}$
  $\Lambda, \Omega \leftarrow \Lambda, \Omega \cup J, G$
  $\bar{m}, \bar{K} \leftarrow \mathbb{GP}(m(\Omega), K(\Omega), \Lambda)$ ▷ Model posterior
  $\tilde{G} \leftarrow \text{argmin}_G \, \alpha(\bar{m}, \bar{K})$ ▷ Render new controller parameters
  return $\tilde{G}$
end function

4

Experiments and results

This chapter presents details of the setup for the experiments as well as the implementation of the approach presented in chapter 3. The results from the experiments are also presented in this chapter. It includes a discussion where the problem description outlined in section 1.3 being central.

4.1 Experiment setup

The experimental setup consists of a Universal Robot™ UR10 together with Onrobot™ RG2-FT two-finger gripper, shown in Figure 4.1. Onrobot™ RG2-FT is equipped with six-axis sensor values in each finger and takes as input the desired width and force.

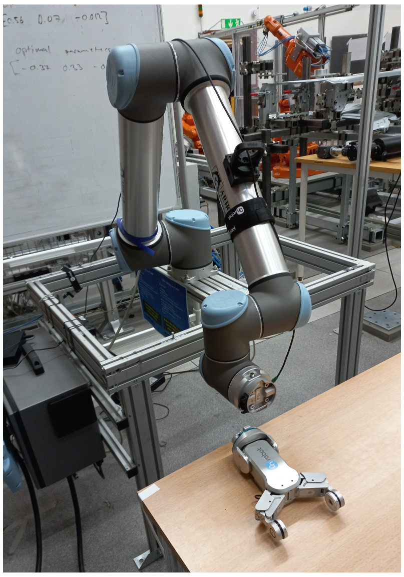


Figure 4.1: The setup employed to perform the training and testing experiments: (left) Universal Robot UR10 and OnRobot RG2-FT gripper. The gripper is equipped with three-axis Force/Torque sensors at its fingertips.

The cost matrices for Equation (2.3) are chosen to be $Q = \text{diag}[10, 1, 100]$ and $R = 1$. Penalizing the error for the tangential force F_t and the torque τ_z higher than the normal force F_z is rooted in preferring not dropping the object over holding it lightly.

The experiments are carried out for four different objects with different sizes, weights, rigidity and friction coefficients. These objects include a small paper cup (310g), a water bottle (600g), a small milk box (660g) and a big milk box (800g), as shown in Figure 4.2. They are filled with different levels of water or chickpeas, giving them an uneven weight distribution during grasp execution.

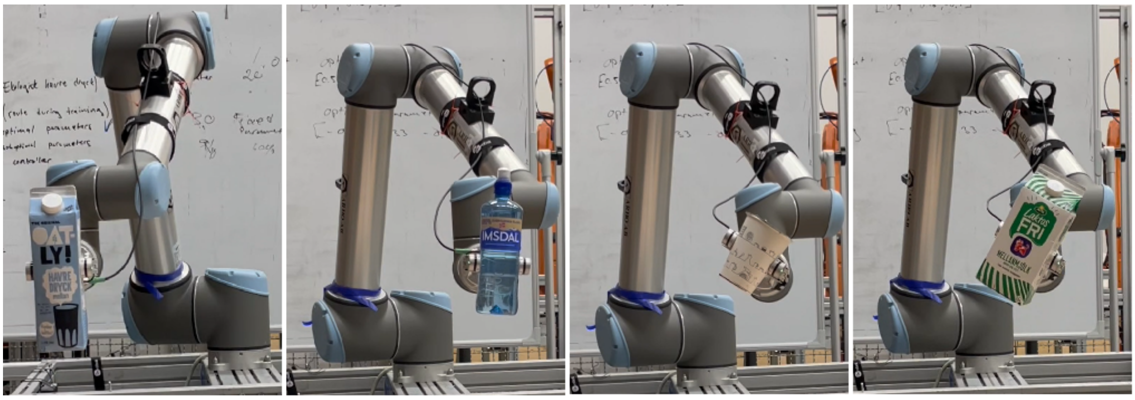


Figure 4.2: The proposed controller is trained to handle four object with different sizes, weights and tactile characteristics: (a) Small milk box half-filled with water (total weight of 660g). (b) Plastic bottle filled with water (total weight of 600g). (c) Paper cup filled with grain (total weight of 310g). (d) Large milk box half-filled with water (total weight of 800g)

A pick-and-place route is devised for the training session, which includes lifting, lightly shaking, rotating and placing the objects during a time period of approximately 30 seconds. When one pick-and-place cycle is completed, the algorithm finds a new set of parameters and the pick-and-place procedure is repeated. One cycle is thus one iteration in the Bayesian optimization algorithm. With a sampling frequency of 70 Hz, one iteration for the training session gives a horizon of around 2100 samples for Equation (2.3). A pick and place cycle starts with an initial predefined grasp plan, including the pose for the grasping attempt and an estimate of the width of the object to be grasped. Once the two fingers of the gripper are positioned around the object at the grasping pose, the robot attempts to pick up the object. The initial width is chosen such that, without reactive grasping, the grasp execution will fail most of the time.

When the training session for an object is finished, the optimal controller parameters are evaluated in a pick-and-place scenario and a shaking scenario. The pick-and-place route for testing differs from training, both in terms of duration and the order of the movements. However, the testing incorporates the same challenging movements as during the training. The shaking experiment involves fast and rough movement of the objects in a vertical path to impose a grasping scenario that differs from the training route but that is still challenging. Each testing scenario is performed 10 times for each object.

The size of the search space for the controller parameters is restricted by taking into account the resolution of the Onrobot™ RG2-FT sensors and the speed of its actuators. The minimum value that can be registered by the sensors when grasping an object multiplied with a gain outside of the search space would result in overshooting for any object. This means that the gripper would open and close much to fast with a high gain.

Furthermore, the BO algorithm is initiated with parameters in the center of the search space, G_{init}, to allow for equal distance in all directions in the beginning of the exploration. The optimized parameters, G_{opt} are then compared to G_{init} to demonstrate if any improvements are made and how much that can be gained by employing the BO algorithm. Finally, a manual search for optimal parameters is made for one object. This is to evaluate the effectiveness of the BO algorithm and demonstrate if it is worth wile automatically tuning the controller.

4.2 Results

Table 4.1 shows the success rates for the optimal controller parameters G_{opt} vs. the initial controller parameters G_{init}. G_{opt} outperforms the initial parameters for all objects in both testing scenarios. The biggest improvements were for the water bottle and the big milk box in the pick-and-place scenario, which went from 0% success rate to 100% success rate. For both objects, the failures occurred early in the pick-and-place attempt, due to the weight and the stiffness of the object. For stiffer and heavier objects, having parameters that lead to overshooting when following the reference means as soon as the gripper width becomes too large, slippage occurs which could lead to grasp failure.

	Pick and Place		**Shake**	
Object	G_{init}	G_{opt}	G_{init}	G_{opt}
Paper Cup (wgt. 310g)	40%	90%	60%	80%
Water Bottle (wgt. 600g)	0%	100%	40%	100%
Small MilkBox (wgt. 660g)	30%	100%	60%	100%
Big MilkBox (wgt. 800g)	0%	100%	30%	100%

Table 4.1: Success rates for the controllers tuned with an expert. Initial controller parameters G_{init}, and for the controllers with optimal gains G_{opt} obtained with Bayesian optimization. These rates are calculated based on 10 trials for each case.

Figure 4.3 shows an unsuccessful pick-and-place attempt with G_{init} in the scenario with the water bottle. The controller manages to compensate for an unfirm grasp in the beginning of the pick-and-place attempt, and the water bottle is picked up successfully. However, when the object is stabilized, the controller eases the grasp too quickly, which results in the object being dropped. This can be clearly seen from the measurements in Figure 4.4 with the big milk box. The controller decreases the width during the first 2 seconds of the lift resulting in a stable grasp. After around 2.5 seconds, the controller starts increasing the width due to an error

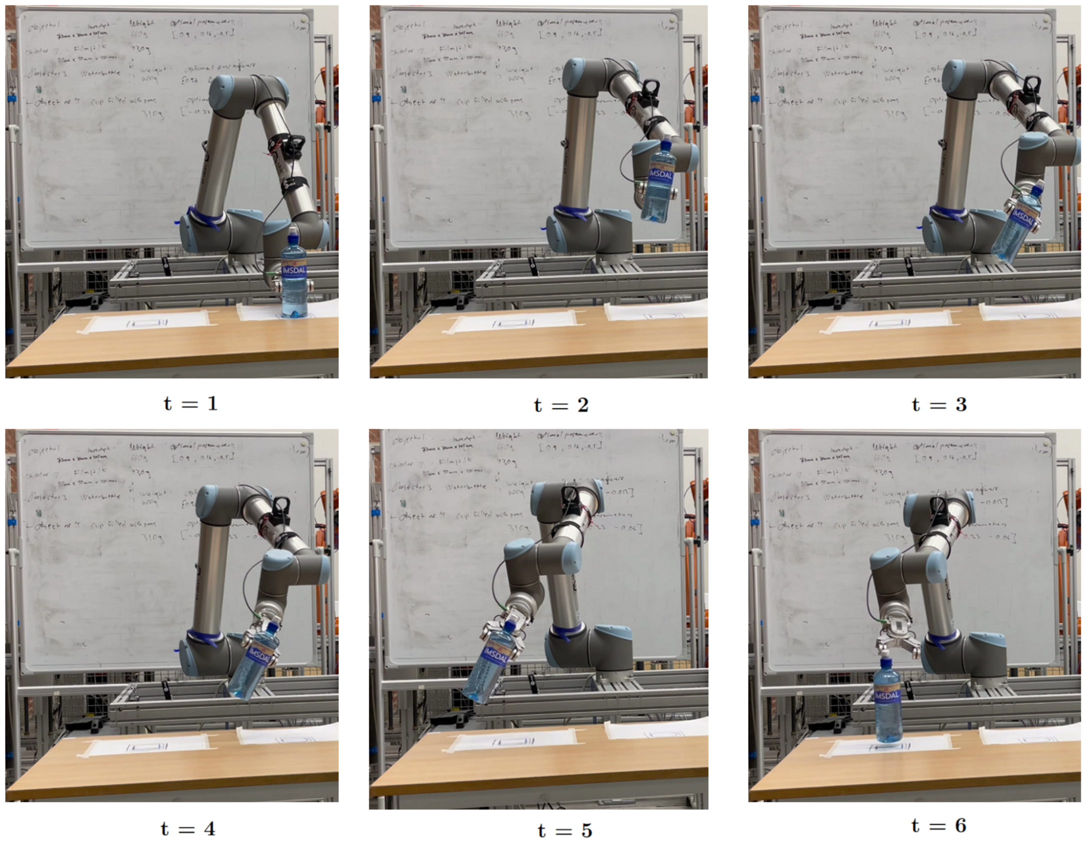


Figure 4.3: Unsuccessful attempt to grasp and move a water bottle with sub-optimal parameters. At time instance (t) equal to 1, the water bottle is successfully picked up with a stable grasp. At t=2, the controller is easing the grasp to ensure stability without excessive force. At t=3, t=4 and t=5, the quick action of the controller is instead leading to a more unstable grasp which ultimately results in dropping the water bottle (t=6).

between the reference and the measurements of the normal force. This leads to opening the gripper too quickly and subsequently not reacting fast enough when the object is slipping. In contrast, when the reactive grasping controller runs with optimal parameters, it can be seen in Figure 4.5 that the milk box is firmly grasped throughout the pick-and-place motion. The algorithm recognized during the training session that easing the grasp for the big and heavy milk box would only result in grasp failure, thereby favoring parameters that have a low gain for regulating the error in the normal force.

In the case of a less rigid and more fragile object like the paper cup, the algorithm manages to find controller parameters that lets the gripper grasp the paper cup firmly yet without damaging it. It can be seen in Table. 4.1 that the optimal controller parameters G_{opt} outperforms G_{init} for both the pick-and-place and shake experiments. However, although the controller performs better with G_{opt}, there were three attempts in total which we deemed as failures due to unstable grasps. In these three cases, the paper cup was moving around the z-axes of the gripper without

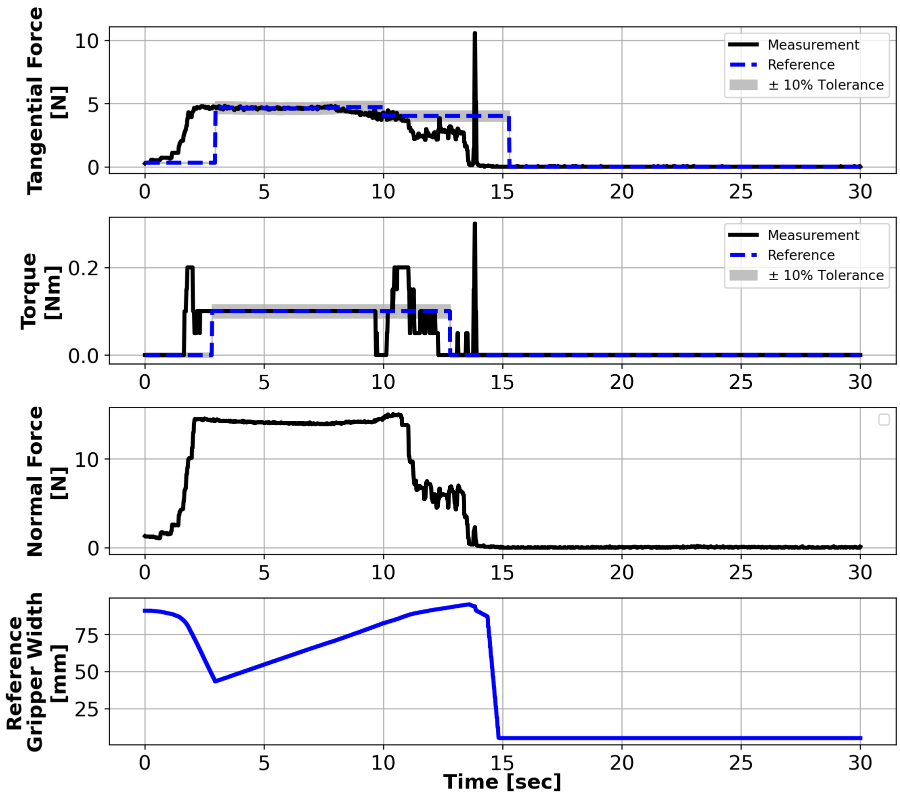


Figure 4.4: Sensor values and references for the two forces and torque with the desired width during a Pick and Place attempt for the big milk box. The milk box is successfully picked up with the width decreasing steadily during first 2 seconds. When the grasp is stable enough, the error for the tangential force and the torque is brought to zero. However, the error in the normal force remains and width is therefore increased again. With sub-optimal controller parameters, the width was increased too quickly for the controller to react to the slippage that occurred around 10 seconds in the attempt and the milk box is subsequently dropped.

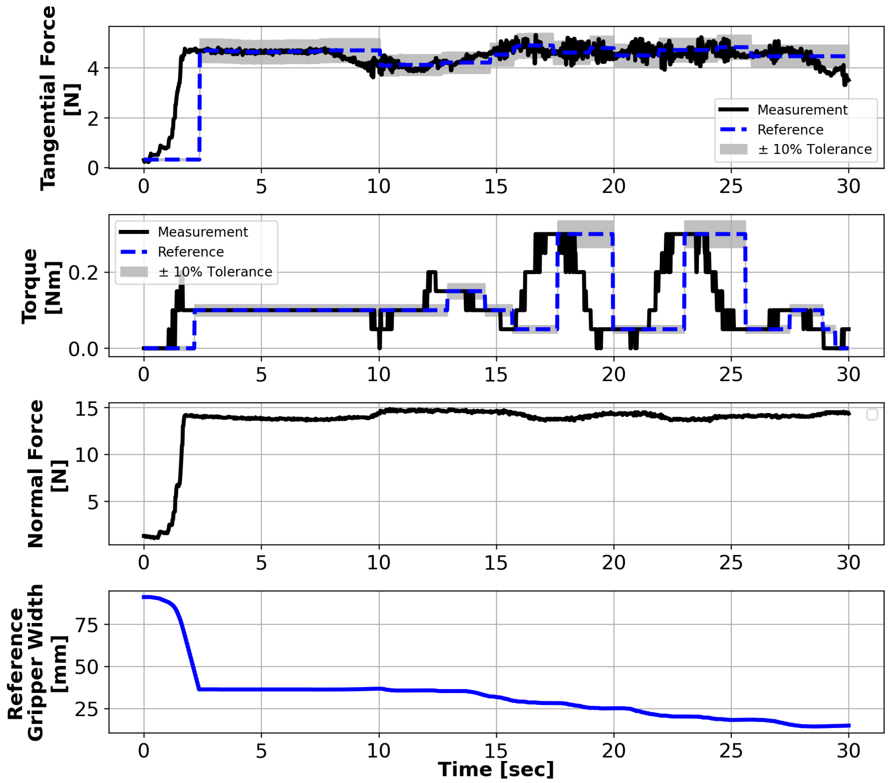
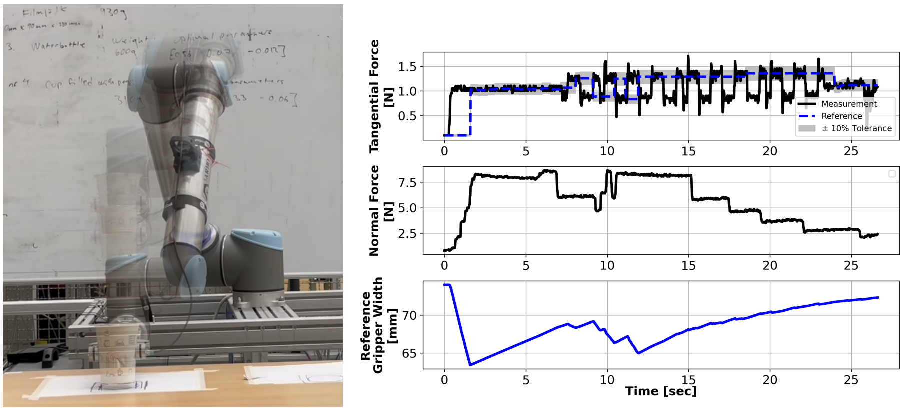


Figure 4.5: Sensor values and references for the two forces and torque with the desired width during a Pick and Place attempt for the big milk box with optimal parameters G_{opt}. The Pick and Place attempt is performed successfully with a firm grasp throughout the motion. The gain for the error in the normal force is so small compared to the gain for the error in tangential force and the torque, leaving the gripper grasping the object firmly throughout the motion without increasing the width.

being detected by the sensors. This is most likely due to the shape of the paper cup which makes it difficult to engage the sensors in some cases. But in the cases where the sensors were excited by movements of the paper cup during an entire grasp execution, the controller performed as intended. Figure 4.6 shows the results of a shaking experiment with the paper cup, with optimal parameters. As it can be seen, the controller handles the challenging motion by updating the reference values and regulating the states. At the beginning, once the system detects the cup is slipping away, it tightens the grasp. However, once the grasp is stabilized the controller starts increasing the width. At the end, the values are converging to stability. It is evident from Figure 4.6 that the width is efficiently controlled, showing that the reactive grasping controller regulates the grasp by decreasing the width of the gripper when instability is detected while easing the grasp when the object is stable.



(a) (b)

Figure 4.6: Sensor measurements and reference input width in the experiment to hold and shake a paper cup filled with grain (weight equal to 310g). In this experiment the cup is shaken up and down along the z-axis of the gripper. In the beginning the cup starts slipping. The controller reacts by tightening the grip. However, after the grip is stabilized, the width is relaxed so that the gripper does not squeeze the cup, thereby avoiding to deform it.

Figure 4.7 shows the exploration for optimal controller parameters for the water bottle over 30 iterations. Here, a comparison is made against manually tuned controller parameters as a baseline depicted as a black star in the search space. The baseline was found with extensive manual search and was included to test the effectiveness of the algorithm. In the first iteration which starts of with parameters in the center of the search space, the object is picked up successfully but dropped soon after. After 5 iterations, the suggested parameters get even worse by dropping the water bottle even sooner. After 9 iterations, the algorithm locates parameters close to the baseline that outperforms all previous parameters and successfully carries out the entire Pick and Place motion with minimal slippage. When the algorithm is terminated after 30 iterations, an entire area of well performing parameters is located near the baseline, showing that the algorithm successfully tunes the reactive grasping controller for optimal performance much more efficiently than manual-tuning.

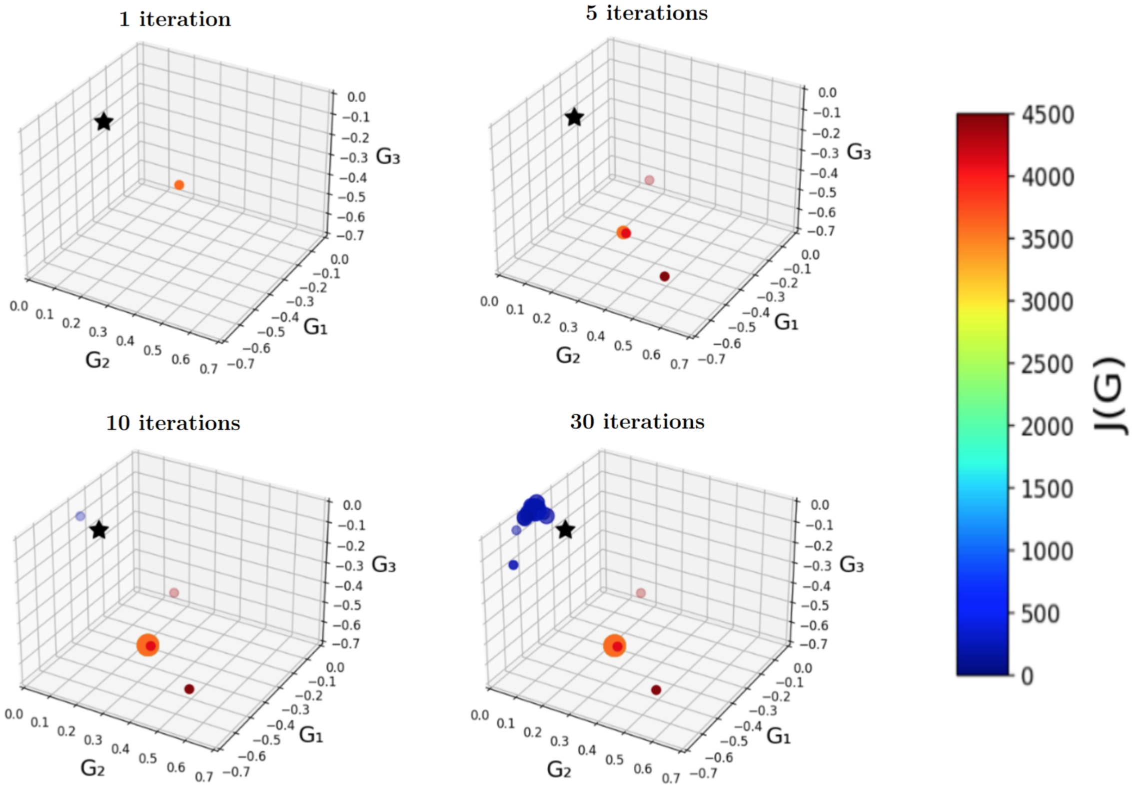


Figure 4.7: Evolution of control parameters for the water bottle as the number of iterations increase in the Bayesian optimization loop. A single point shows the suggested parameters after one iteration while the size of the point shows more than one suggestion in the same place. Red color indicates poor controller performance, i.e. a high value of Equation (2.3), while the blue color indicates better controller performance. The black star is a baseline which is found manually through experiments. The algorithm is initiated with parameters in the center of the search space and after 5 iterations, the suggested parameters result in poor grasp execution with the water bottle being dropped. After 10 iterations, parameters closer to the baseline are suggested which yield better controller performance. When the algorithm has run for 30 iterations it locates a neighborhood near the baseline with multiple parameters with good performance.

5
Conclusion

The aim of this thesis was to design and evaluate an approach for reactive grasping of novel objects. The core of the presented method is a feedback controller based on LQR which is auto-tuned with Bayesian optimization. Connecting this back to the problem description outlined in section 1.3, the first and second central questions were:

- ***How could a feedback controller best be designed for processes that are not known a priori, that have important nonlinear dynamics that cannot be ignored and for processes that have extensive noise?***
- ***How could information about a current unstable grasp be used to take reactive corrective actions with an adaptive controller to increase grasp stability without deforming the object?***

Bayesian optimization was used together with Gaussian process to find optimal parameters for the feedback controller. The experiments included a real robotic setup with a parallel gripper equipped with F/T sensors. The results showed that the designed controller was highly effective at responding to changes in stability during grasp execution. A comparison between optimal and sub-optimal controller parameters showed that the performance was greatly improved by optimizing the feedback controller. At the same time, the approach proved to be suitable for handling fragile objects, such as a paper cup. The controller was able to react to instability when grasping a cup but without applying excessive force.

The third central question in this thesis was:

- ***How could the feedback controller be tuned automatically in an efficient way?***

When the parameters of the BO tuned controller for one object were compared to manually found and well-performing parameters, it was shown that the BO algorithm indeed could locate an entire area of high quality parameters. This was done in just a fraction of the time in comparison to manually tuning the controller, making it truly efficient when auto-tuning.

The main questions in this thesis were all shown to be appropriately addressed with the proposed reactive grasping approach. However, despite the success of the method shown with the experiments designed in this thesis, more thorough evaluation is needed. Future work should include expanded experiments to further test the robustness of the proposed approach by, for instance, designing pick and place tasks with different grasp poses and object orientations. This should be done to establish the generalizability of the method, i.e. to ensure the robustness of the controller in multiple different grasping scenarios. Furthermore, additional objects with various characteristics could be added for extensive evaluation of the proposed approach.

Future work could also investigate how the controller performs together with a grasp planner in a pipeline for grasping novel objects. It may also include the expansion of the method by incorporating a memory bank over several learned controllers. In practice, this could mean storing the posterior distributions of all objects learned during training and using them for similar objects to learn even faster.

Bibliography

[1] V.-D. Nguyen, “Constructing stable grasps in 3d,” in *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, vol. 4, 1987, pp. 234–239.

[2] Y. Li, Q. Lei, C. Cheng, G. Zhang, W. Wang, and Z. Xu, “A review: machine learning on robotic grasping,” 03 2019, p. 54.

[3] D. Floreano, P. Husbands, and S. Nolfi, *Springer Handbook of Robotics*, 01 2008.

[4] Y. Bekiroglu, J. Laaksonen, J. A. Jorgensen, V. Kyrki, and D. Kragic, “Assessing grasp stability based on learning and haptic data,” *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 616–629, 2011.

[5] H. Dang and P. K. Allen, “Learning grasp stability,” in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 2392–2397.

[6] 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)*, 2016, pp. 165–171.

[7] M. Li, Y. Bekiroglu, D. Kragic, and A. Billard, “Learning of grasp adaptation through experience and tactile sensing,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 3339–3346.

[8] H. Dang and P. K. Allen, “Grasp adjustment on novel objects using tactile experience from similar local geometry,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 4007–4012.

[9] J. M. Romano, K. Hsiao, G. Niemeyer, S. Chitta, and K. J. Kuchenbecker, “Human-inspired robotic grasp control with tactile sensing,” *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1067–1079, 2011.

[10] Q. Li, O. Kroemer, Z. Su, F. F. Veiga, M. Kaboli, and H. J. Ritter, “A review of tactile information: Perception and action through touch,” *IEEE Transactions on Robotics*, vol. 36, no. 6, pp. 1619–1634, 2020.

[11] J. Tegin and J. Wikander, “A framework for grasp simulation and control in domestic environments,” *IFAC Proceedings Volumes*, vol. 39, no. 16, pp. 490–495, 2006, 4th IFAC Symposium on Mechatronic Systems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667015342142

[12] J. Tegin, S. Ekvall, D. Kragic, J. Wikander, and B. Iliev, "Demonstration-based learning and control for automatic grasping," *Intelligent Service Robotics*, vol. 2, 01 2009.

[13] E. Hyttinen, D. Kragic, and R. Detry, "Learning the tactile signatures of prototypical object parts for robust part-based grasping of novel objects," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4927–4932.

[14] M. Petit, E. Dellandrea, and L. Chen, "Bayesian optimization for developmental robotics with meta-learning by parameters bounds reduction," 2020.

[15] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson, "Cable manipulation with a tactile-reactive gripper," *The International Journal of Robotics Research*, vol. 0, no. 0, pp. 1–17, 2021. [Online]. Available: https://doi.org/10.1177/02783649211027233

[16] F. Veiga, J. Peters, and T. Hermans, "Grip stabilization of novel objects using slip prediction," *IEEE Transactions on Haptics*, vol. 11, no. 4, pp. 531–542, 2018.

[17] X. Huang, M. Halwani, R. Muthusamy, A. Ayyad, D. Swart, L. Seneviratne, D. Gan, and Y. Zweiri, "Real-time grasping strategies using event camera," *Journal of Intelligent Manufacturing*, vol. 33, 02 2022.

[18] L. Bertoni, D. Torielli, Y. Zhang, N. G. Tsagarakis, and L. Muratore, "Towards a generic grasp planning pipeline using end-effector specific primitive grasping actions," in *2021 20th International Conference on Advanced Robotics (ICAR)*, 2021, pp. 722–729.

[19] H. Zhang, J. Peeters, E. Demeester, and K. Kellens, "A cnn-based grasp planning method for random picking of unknown objects with a vacuum gripper," *Journal of Intelligent Robotic Systems*, vol. 103, 12 2021.

[20] M. R. Cutkosky, R. D. Howe, and W. R. Provancher, *Force and Tactile Sensors*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 455–476. [Online]. Available: https://doi.org/10.1007/978-3-540-30301-5_20

[21] K. Sasaki, K. Koyama, A. Ming, M. Shimojo, R. Plateaux, and J.-Y. Choley, "Robotic grasping using proximity sensors for detecting both target object and support surface," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 2925–2932.

[22] K. Hsiao, P. Nangeroni, M. Huber, A. Saxena, and A. Y. Ng, "Reactive grasping using optical proximity sensors," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 2098–2105.

[23] K. Shimonomura, H. Nakashima, and K. Nozu, "Robotic grasp control with high-resolution combined tactile and proximity sensing," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 138–143.

[24] A. K. Hossein Akbari, “Improving the coupling errors of a maltese cross-beams type six-axis force/moment sensor using numerical shape-optimization technique, measurement,” in *Proceedings. IEEE International Conference on Robotics and Automation*, vol. 126, 1987, pp. Pages 342–355.

[25] M. Rajab, *Linear Quadratic Regulator (LQR) Controller Design for DC Motor Speed Using Matlab Application.* UMP, 2008. [Online]. Available: https://books.google.se/books?id=HVisAQAACAAJ

[26] H. Khalil, *Nonlinear Systems*, ser. Pearson Education. Prentice Hall, 2002. [Online]. Available: https://books.google.se/books?id=t_d1QgAACAAJ

[27] M. Mitra, “Editorial on advances in machine learning and robotics,” vol. 3, pp. 1–5, 03 2019.

[28] T. Qin, *Machine Learning Basics.* Singapore: Springer Singapore, 2020, pp. 11–23. [Online]. Available: https://doi.org/10.1007/978-981-15-8884-6_2

[29] N. Vakada, A. Chunduri, K. Manne, V. L. Meda, and K. Sailaja, “Autonomous obstacle avoidance robot using regression,” EasyChair Preprint no. 3581, EasyChair, 2020.

[30] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*, 01 2008, p. 266.

[31] E. Schulz, M. Speekenbrink, and A. Krause, “A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions,” *bioRxiv*, 2017. [Online]. Available: https://www.biorxiv.org/content/early/2017/10/10/095190

[32] H. Yu, T. Xie, S. Paszczynski, and B. M. Wilamowski, “Advantages of radial basis function networks for dynamic system design,” *IEEE Transactions on Industrial Electronics*, vol. 58, no. 12, pp. 5438–5450, 2011.

[33] J. Wainer and P. Fonseca, “How to tune the rbf svm hyperparameters?: An empirical evaluation of 18 search algorithms,” 2020. [Online]. Available: https://arxiv.org/abs/2008.11655

[34] P. I. Frazier, “A tutorial on bayesian optimization,” 2018. [Online]. Available: https://arxiv.org/abs/1807.02811

[35] 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, 2016.

[36] J. T. Wilson, F. Hutter, and M. P. Deisenroth, “Maximizing acquisition functions for bayesian optimization,” 2018. [Online]. Available: https://arxiv.org/abs/1805.10196

[37] 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, 2016.

[38] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se