

# **Active Grasp Synthesis for Grasping Unknown Objects**

Berk Çallı



# **Active Grasp Synthesis for Grasping Unknown Objects**

PROEFSCHRIFT

ter verkrijging van de graad van doctor  
aan de Technische Universiteit Delft,  
op gezag van de Rector Magnificus Prof. dr. ir. K.C.A.M. Luyben,  
voorzitter van het College voor Promoties,  
in het openbaar te verdedigen op 5 Januari 2015 om 12:30 uur  
door

Berk ÇALLI

Master of Science in Mechatronics Engineering  
Sabancı University

geboren te Izmir, Turkije.

Dit proefschrift is goedgekeurd door de promotor:

Prof. dr. ir. P.P. Jonker

Copromotor: Dr. ir. M. Wisse

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof. dr. ir. P.P. Jonker	Technische Universiteit Delft, promotor
Dr. ir. M. Wisse	Technische Universiteit Delft, copromotor
Prof. dr. ir. J. Hellendoorn	Technische Universiteit Delft
Prof. dr. R.C. Veltkamp	Universiteit Utrecht
Prof. dr. ir. E.J. van Henten	Wageningen Universiteit en Researchcentrum
Prof. dr. O. Brock	Technische Universität Berlin
Dr. W. Caarls	Universidade Federal do Rio de Janeiro
Prof. dr. F.C.T. van der Helm	Technische Universiteit Delft (reservelid)

The research leading to the results of this thesis has received funding from the Dutch Ministry of Economic Affairs.

Copyright © 2014 by B. Çalli

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without the prior permission of the author.

ISBN 978-94-6186-410-9

Author email: berkc@sabanciuniv.edu

*To my family...*



# Contents

<b>Summary</b>	<b>v</b>
<b>Samenvatting</b>	<b>vii</b>
<b>0 Preface</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Motivation . . . . .	5
1.2 Thesis goals . . . . .	7
1.3 Proposed Framework . . . . .	7
1.4 Thesis Structure . . . . .	9
<b>2 Fundamentals of Grasp Synthesis</b>	<b>11</b>
2.1 Basic Concepts in Grasping Literature: . . . . .	11
2.1.1 What is a Grasp? . . . . .	11
2.1.2 Contact Wrench and Wrench Space . . . . .	12
2.1.3 Force Closure vs. Form Closure . . . . .	12
2.1.4 First and Second Order Mobilities . . . . .	12
2.1.5 Grasp Synthesis . . . . .	13
2.1.6 Measures of Grasp Quality . . . . .	13
2.2 Factors that Affect Grasp Synthesis Process . . . . .	15
2.2.1 The Grasp Scenario and the Effect of the Surroundings . .	15
2.2.2 Grasp Types . . . . .	16
2.2.3 Gripper Types . . . . .	17
2.2.4 2D Objects vs. 3D Objects . . . . .	19
2.2.5 Grasping considering the task and its formulation . . . . .	19
2.2.6 Object being known/familiar/unknown . . . . .	20

<b>3 Grasp Synthesis Literature</b>	<b>23</b>
3.1 Grasp Synthesis Algorithms for Known Objects . . . . .	24
3.2 Grasp Synthesis Algorithms for Familiar Objects . . . . .	27
3.3 Grasp Synthesis Algorithms for Unknown Objects . . . . .	27
3.4 Conclusions . . . . .	31
3.4.1 Remarks on the Algorithms for Known Objects: . . . . .	31
3.4.2 Remarks on the Algorithms for Familiar Objects: . . . . .	31
3.4.3 Remarks on Algorithms for Unknown Objects: . . . . .	32
<b>4 Grasping of Unknown Objects via Curvature Maximization using Active Vision</b>	<b>35</b>
4.1 Introduction . . . . .	36
4.2 Modeling with EFD . . . . .	38
4.3 Obtaining Grasping Points . . . . .	39
4.4 Filtering Grasping Points . . . . .	40
4.5 Visual Servoing Rule . . . . .	42
4.6 Simulation Results: . . . . .	44
4.7 Experimental Results . . . . .	46
4.8 Conclusion . . . . .	46
<b>5 Comparison of Extremum Seeking Control Algorithms for Robotic Applications</b>	<b>51</b>
5.1 Introduction . . . . .	52
5.2 Analog Extremum Seeking Control Methods . . . . .	53
5.2.1 Sliding mode ESC . . . . .	54
5.2.2 Neural network ESC . . . . .	55
5.2.3 Approximation based ESC . . . . .	56
5.2.4 Perturbation based ESC . . . . .	57
5.2.5 Adaptive ESC . . . . .	58
5.3 Simulations . . . . .	59
5.3.1 Unidimensional Simulations: . . . . .	59
5.3.2 Two dimensional Simulations: . . . . .	64
5.4 Analysis . . . . .	64
5.5 Experimental Results . . . . .	66
5.6 Conclusion . . . . .	68
<b>6 Grasping Unknown Objects with Active Exploration via Online Viewpoint Optimization</b>	<b>71</b>
6.1 Introduction . . . . .	72
6.2 Overview . . . . .	73
6.2.1 Viewpoint optimization problem . . . . .	73

6.2.2	Proposed framework . . . . .	75
6.3	Grasp Synthesis . . . . .	76
6.3.1	Grasping strategy . . . . .	76
6.3.2	Modeling using Elliptic Fourier Descriptors . . . . .	78
6.3.3	Calculating the best grasp . . . . .	79
6.4	Objective Value Calculation . . . . .	80
6.4.1	Grasp Quality Component . . . . .	80
6.4.2	Workspace Constraints Component . . . . .	80
6.4.3	Optimizer specific component . . . . .	81
6.5	Extremum Seeking Controller . . . . .	81
6.6	Experimental Results . . . . .	84
6.7	Discussion . . . . .	90
6.8	Conclusion . . . . .	92
<b>7</b>	<b>Viewpoint Optimization for Grasp Synthesis via Supervised Learning</b>	<b>93</b>
7.1	Introduction . . . . .	94
7.2	Global Viewpoint Optimization Methodology . . . . .	96
7.2.1	The Viewpoint Optimization Scheme . . . . .	96
7.2.2	The Training Process . . . . .	97
7.2.3	Automated Training Data Generation Procedure . . . . .	99
7.3	An Implementation of the Viewpoint Optimization Methodology . . . . .	102
7.3.1	Grasp Synthesis Algorithm . . . . .	102
7.3.2	State Modelling . . . . .	104
7.4	Simulation Results . . . . .	104
7.5	Conclusions and Future Work . . . . .	112
<b>8</b>	<b>Active Object Recognition for Service Robots with Extremum Seeking Control</b>	<b>113</b>
8.1	Introduction . . . . .	114
8.2	Related Work . . . . .	115
8.2.1	The effect of the structured noise . . . . .	116
8.2.2	Inefficiency caused by discrete search . . . . .	118
8.2.3	The effect of imperfect training data . . . . .	118
8.3	Active Object Recognition with Extremum Seeking Control . . . . .	118
8.3.1	System Architecture . . . . .	119
8.3.2	Object Recognition . . . . .	119
8.3.3	Objective Value Calculation . . . . .	120
8.3.4	Extremum Seeking Control . . . . .	121
8.4	Experimental results . . . . .	122
8.5	Conclusion . . . . .	128
8.6	Acknowledgements . . . . .	128

<b>9 Conclusion</b>	<b>131</b>
<b>Bibliography</b>	<b>137</b>
<b>Acknowledgments</b>	<b>153</b>
<b>About the Author</b>	<b>157</b>

# Summary

## **Active Grasp Synthesis for Grasping Unknown Objects**

Manipulation is a key feature for robots which are designed to work in daily environments like homes, offices and streets. These robots do not often have manipulators that are specialized for specific tasks, but grippers that can grasp the target object. This makes grasping a crucial ability that enables many manipulation tasks.

Robotic grasping is a complex process with various aspects: design of the gripper, detecting grasping points/regions that lead to a stable grasp (grasp synthesis), avoiding surrounding objects while executing the grasp (obstacle avoidance), detecting task related features of the object, altering the pose of the object to free up graspable regions (pre-grasp manipulation) are some of these aspects. In order to maintain a robust grasping system, all these aspects should work in harmony, aid each other and preferably cover each others mistakes.

Among these aspects, vision based grasp synthesis for unknown objects forms a large portion of the robotic grasping literature. These algorithms deal with the problem of detecting grasping points or regions on a target object without an object shape model supplied *a priori*; instead they utilize visual information provided by the robot's sensors. The majority of these algorithms use one single image of the target object for grasp synthesis, and make implicit or explicit assumptions on the missing shape information of the target object. The missing information is a function of the shape of the object as well as the viewpoint of the vision sensor. So far in literature, there is no reliable grasp synthesis algorithm that can cope with the missing shape information and provide successful grasp synthesis for a large variety of objects and viewpoints.

This thesis proposes a novel framework in which grasp synthesis process is coupled with active vision strategies in order to relax the assumptions on the viewpoint of the vision sensor and increase grasp success rate. Unlike prior

work which considers grasp synthesis as a passive data analysis process that uses only the provided image of the target object, the proposed framework introduces strategies to improve the quality of the data by leading the sensor to viewpoints by which the grasp synthesis algorithms can generate higher quality grasps. With such a strategy, the burden of the grasp synthesis algorithms is shared with an active vision stage which boosts their success rates.

Within the framework two novel methodologies are presented each of which utilizes a different active vision strategy. In the first methodology, local viewpoint optimization methods are analyzed; an extremum seeking control based optimization method is utilized to optimize the viewpoint of the sensor locally by maximizing the grasp quality value continuously. This methodology is easy to implement as it does not necessitate any prior training, but it has a risk of getting stuck at local optima. With this method up to 94% success rate has been achieved for power grasps. However, it is observed that, noise on the grasp quality value and not being able to avoid local optima affect the performance negatively. In the second methodology, supervised learning algorithms are used to obtain an exploration policy. This strategy has a lower risk of getting stuck at local optima, but requires a training process. Furthermore, with this strategy, the information acquired during the process can be fused, and assumption on the missing object shape data can be relaxed significantly. The experimental results show that the strategy is superior than heuristic based and random search techniques in terms of both success rate and efficiency.

With the proposed framework, we hope to encourage a new way of thinking about the grasp synthesis problem by introducing the use of active vision tools. We believe such an approach can have significant contribution for solving this challenging robotics problem.

*Berk Çallı*

# Samenvatting

## **Actieve greepsynthese voor het vastpakken van onbekende objecten**

Het manipuleren van objecten is een belangrijke taak voor robots die in dagelijkse omgevingen moeten werken, zoals thuis, op het kantoor, of op straat. Zulke robots hebben vaak geen gespecialiseerde manipulatoren, maar grijppers om objecten vast te pakken. Vastpakken is dus van cruciaal belang bij veel manipulatietaak.

Het vastpakken van objecten door robots is een complex proces. Enkele aspecten zijn: het ontwerp van de grijper, het detecteren van aangrijppunten die een stabiele greep opleveren (greepsynthese), het omzeilen van andere objecten in de buurt bij het uitvoeren van de greep (obstakelvermijding), het detecteren van taakgerelateerde eigenschappen van het object, en het positioneren van het object om betere grepen mogelijk te maken (manipulatie voor het pakken op zich). In een robuust systeem moeten al deze aspecten in harmonie samenwerken, waarbij fouten van de een door de ander worden hersteld.

Onder de genoemde aspecten vormt beeldgebaseerde greepsynthese voor onbekende objecten een belangrijk deel van de literatuur. Zulke algoritmes houden zich bezig met het detecteren van aangrijppunten of -regio's op een object zonder een vooraf gegeven model; in plaats daarvan gebruiken ze visuele informatie op basis van sensoren. De meeste varianten gebruiken een enkel beeld van het object voor greepsynthese, en vullen de rest in op basis van impliciete of expliciete veronderstellingen. De ontbrekende informatie hangt af van zowel de vorm van het object als de positie van de beeldsensor. Tot nu toe bestaat er geen betrouwbaar algoritme voor greepsynthese dat goed kan omgaan met deze ontbrekende informatie en zo een goede greep kan vinden voor een groot aantal verschillende objecten en sensorposities.

Dit proefschrift beschrijft een nieuw raamwerk waarin greepsynthese wordt gecombineerd met actieve waarnemingsstrategieën om zo de veronderstellingen over het standpunt van de beeldsensor af te zwakken en meer succesvolle grepen

te vinden. In tegenstelling tot eerder werk wordt greepsynthese op deze manier niet gezien als een passieve procedure, maar als een actief proces waarin de sensor naar posities wordt geleid waarop het greepsynthesealgoritme een betere greep kan vinden. Met zo'n strategie wordt de last van greepsynthese gedeeld met een actieve waarnemingsstrategie om zo het slagingspercentage te verhogen.

Binnen het raamwerk worden twee nieuwe methoden gepresenteerd, op basis van verschillende strategieën. In de eerste methode worden lokale technieken voor het optimaliseren van het gezichtspunt geanalyseerd. Een optimalisatiemethode gebaseerd op extremen-zoekend regelen (extremum seeking control) wordt gebruikt om voortdurend het gezichtspunt te optimaliseren door het maximaliseren van de voorspelde greepkwaliteit. Deze methode is makkelijk uit te werken en heeft geen training nodig, maar kan vast komen te zitten in een lokaal optimum. Met deze methode is een slagingspercentage van 94% bereikt voor omsluitende grepen, maar ruis en lokale optima hebben een negatief effect op de prestaties.

In de tweede methode worden patroonherkenningsalgoritmes gebruikt om een verkenningsstrategie te bepalen. Deze strategie heeft minder kans om in een lokaal optimum vast komen te zitten, maar moet eerst getraind worden. Een voordeel is dat met deze strategie alle informatie die wordt verkregen gedurende het grijpproces kan worden gecombineerd, zodat de veronderstellingen over de vorm van het object sterk kunnen worden afgewakt. Uit experimenten blijkt dat de methode beter werkt dan willekeurig en heuristisch zoeken.

Met het voorgestelde raamwerk hopen we een nieuwe manier van denken te bewerkstelligen door tools uit het veld van actieve waarneming in het onderzoek naar greepsynthese te introduceren. We zijn er van overtuigd dat een dergelijke aanpak een grote bijdrage kan leveren aan het oplossen van dit uitdagende probleem in de robotica.

*Berk Çallı*

# Chapter 0

## Preface

*“On mechanical slavery, on the slavery of the machine, the future of the world depends.”*

says Oscar Wilde in his book “The Soul of Man under Socialism”. He pictures a world in which the role of machines is beyond the production lines:

*“Machinery must work for us in coal mines, and do all sanitary services, and be the stoker of steamers, and clean the streets, and run messages on wet days, and do anything that is tedious or distressing.”*

Wilde had a vision that machines serve humans everywhere, in almost every aspect of life. He thought that, if every human equally benefits from the advantages that the machinery can supply, this will be the ultimate solution to poverty, and humanity will only be busy with *“making beautiful things, or reading beautiful things, or simply contemplating the world with admiration and delight”*. Therefore, according to him, the scientific effort should be directed to realizing this future instead of the temporary and ineffective solutions to poverty:

*“And when scientific men are no longer called upon to go down to a depressing East End and distribute bad cocoa and worse blankets to starving people, they will have delightful leisure in which to devise wonderful and marvellous things for their own joy and the joy of everyone else. There will be great storages of force for every city, and for every house if required, and this force man will convert into heat, light, or motion, according to his needs.”*

Wilde agrees with people who think this is a Utopian, and adds:

*“A map of the world that does not include Utopia is not worth even glancing at, for it leaves out the one country at which Humanity is always landing.”*

As humanity we are progressing quite fast in solving the technical difficulties of this beautiful Utopian. Each technological advancement in robotics has a potential to be used as a step towards a better future. Let's hope that we make the right choices while utilizing this potential, so that we can all live in prosperity and peace.

# Chapter 1

## Introduction

Research efforts are increasingly directed to realize the usage of robots in houses, offices and streets. Various service robots are under development in research centers all around the world (Figure 1.1). Despite this effort, the role of robots is still limited to being workers in factories and large warehouses. The current robot technology is simply not good enough yet to make them operational in daily human environments. However, it is also apparent that soon this will change thoroughly; robots will not only impact the social life with their products, but they will also be the subjects of the social life where humans will directly interact with them, speak with them, command them.

This new role of robots is getting more and more necessary for several aspects of human life, e.g. for healthcare, especially for the countries with population aging like the Netherlands and Japan. As reported in [34], around 20% of people aged 65 or above receive either homecare or residential care services in the Netherlands. This took 3.8% of the GDP in 2012 which brings discussions to the sustainability of the quality of the healthcare program [147]. Moreover, according to the estimations of Dutch Central Agency for Statistics, the percentage of people above the age 65 will increase from around 16% to 21.7% in 2025, and this ratio is estimated to reach 25.3% by 2035 [1]. Furthermore, labor shortage in healthcare is in an increasing trend [164]. Developing technologies to enable robotic systems in human daily life would provide cheap and reliable solutions to this healthcare problem. In addition, this technology can bring new possibilities for better and cheaper healthcare for both developed and developing countries.

The core of the problem that prevents robots appearing in daily life is that, unlike factories which are the current niche for robots, the daily environments are uncontrolled, unstructured and dynamic. These properties of the daily environments make the use of conventional offline programming methods inapplicable, since these methods rely on the full knowledge of the robot's and environment's state. This is also valid for the robotic grasping problem: Grasping enables many



**Figure 1.1:** Several service robots from today: (a) Care-o-bot [70] by Fraunhofer Institute for Manufacturing Engineering and Automation, (b) Roby by Delft University of Technology, (c) Amigo by Eindhoven University of Technology, (d) Armar III [2] by Karlsruhe Institute of Technology, (e) PR2 [15] by Willow Garage, (f) REEM by PAL Robotics.

fundamental manipulation tasks for the robots such as fetching/replacing an object, pouring liquids, using tools etc. All these abilities are crucial for service robots.

For making a robot grasp a target object, suitable points/regions (so called grasping points/regions) should be detected on that object. This procedure is called grasp synthesis. If the 3D model of the target object is available, grasp synthesis can be formulated as an optimization problem over the object's surface for the locations of the grasping points/regions, and this problem can be solved by the existing methods in the literature ([108, 18, 120, 115, 80]). However, in daily environments, the variety of objects is huge which makes it infeasible to create a model database of all possible objects. Therefore, manipulation algorithms that use 3D models of the objects are not directly applicable, and algorithms for manipulating unknown objects are needed.

## 1.1 Motivation

This thesis proposes the use of robot's vision sensors actively in the grasp synthesis process unlike the algorithms in the literature. In that context, the grasp synthesis problem for unknown objects is analogous to the task-based navigation problem for mobile robots, and it is useful to analyze these two problems next to each other in order to explain the motivation behind this thesis. For robot navigation in daily environments, path planning using previously loaded maps are not reliable since the environment changes over time considerably, or the robot may find itself in a totally new environment. In these cases, algorithms are needed for generating maps of the unknown environments and updating them continuously. For this purpose, Simultaneous Localization and Mapping (SLAM) [49, 163, 38] algorithms are developed. By continuously synthesizing the data acquired from the robot's sensors, these algorithms generate the map of an unknown environment while simultaneously determining the position of the robot in this map. However, for most of the tasks, SLAM by its own is not enough since the map is usually a medium to achieve a task, not the task itself. For example, if a robot is looking for a specific object in an unknown environment, the map is necessary but not enough since the robot also needs to plan its path within the generated map (that is why the SLAM algorithms are also named as *passive* SLAM algorithms since they passively observe the environment, and do not involve any decision process [162]). That is where *active* SLAM algorithms [102, 37, 62, 166] come into the picture in which SLAM is coupled with online path generation algorithms so that the robot is led to its goal (possible locations of the target object/person, locations to maximize a certain information etc.) together with the map generation and localization processes.

Similar to the task-based navigation problem, the grasp synthesis algorithms for unknown objects should be able to cope with the missing information of the object while planning the manipulation, and a strategy proposed in the literature

for this purpose is to generate a full 3D model of the object prior to manipulation (e.g. [174, 7, 16]). After a full model is obtained, the model based techniques are applied for grasp synthesis. An analogous approach for the object search problem in an unknown environment would be creating a full map of the unknown environment first, and then searching for the target object. Clearly, this strategy is far from efficient. Moreover, generating a full model of the object may not always be possible, because some of the viewpoints of the object which are necessary for model generation may be occluded by other objects (for example, if the target object is on a shelf), or robot's workspace may not be large enough to take all the necessary images for the model generation. These reasons make the usage of model based methods unreliable. Therefore, algorithms that do not use full 3D models are needed.

The algorithms that aim to generate a grasp without using the full shape information of the target object utilize various sensors and data processing techniques, but a common characteristic of all these algorithms is being passive. For instance, there are algorithms that fit a model to the partial shape information (e.g. [107, 158]), algorithms that use 2D features (e.g. [140, 142, 110]) or 2<sup>1/2</sup>D features (e.g. [130, 143, 82]) for grasping. All these algorithms assume that the robot passively observes the target object, and they use the data available for them which is almost all the time only one single image (either color image, depth image or combined color and depth information i.e. an RGB-D image) of the target object. This makes them rely on some assumptions on the viewpoint of the sensor, the shape of the object and the missing data about the object's shape. In other words, the performance of these algorithms are viewpoint dependent, and in addition to that they can only work for the objects where the assumptions on the object shape hold.

All these difficulties bring the necessity of an active grasp synthesis algorithm where the robot contributes the synthesis by its motion. Not using the robot actively in grasp synthesis makes the problem much harder than it actually is; for the object search problem, this approach is similar to having one image of a room and trying to detect a target object without moving the robot and actively searching for it in the room. Just like searching a room for the target object, the robot can actively explore the object for a good grasp, and just like the passive SLAM algorithms form a basis for the active SLAM algorithms, the passive grasping algorithms can very well be a building block for the active grasp synthesis algorithms where the robot improves the quality of the data continuously. Of course, having a random exploration, a random search for a better grasp, hardly leads to efficient results. The exploration should be systematic and should lead the robot in such a way that the quality of the synthesized grasp improves over time.

The power of using the robot actively has been acknowledged in the grasping literature. In [128], a biologically inspired grasping process is investigated, and systematic exploration is detected as the missing link that can bridge the gap between the state of the art algorithms and the biological models of grasping.

Nonetheless, neither in that work nor in any other work in the grasping literature, the methods of exploration are addressed.

## 1.2 Thesis goals

This work aims to fill the gap of active algorithms in the grasp synthesis literature by introducing a viewpoint optimization framework. It presents a novel formulation of the grasp synthesis problem which includes the motion of the robot in the process, and solution strategies are provided within the proposed framework. With this framework mainly two goals have been targeted:

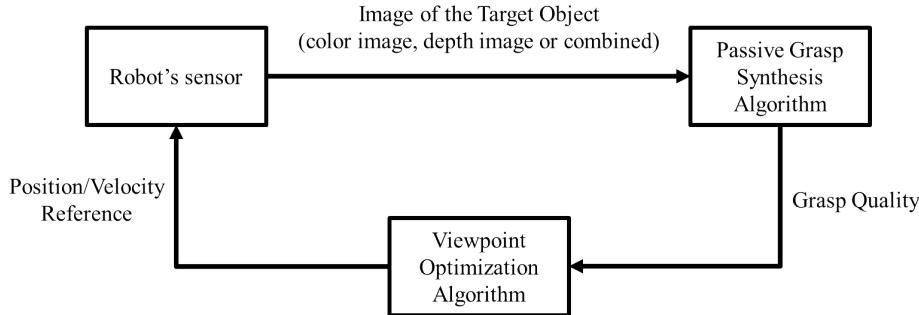
1. Improving the success rate of the current passive grasping algorithms by eliminating the assumptions on sensor viewpoint and object shape by providing tools for changing the viewpoint of the sensor systematically.
2. Increasing the possibilities of solution strategies for the grasping unknown objects field with the ability of the viewpoint optimization. With this ability, the grasp synthesis problem does not have to be considered only as a data analysis problem, but a complete procedure which also deals with the quality of the data.

## 1.3 Proposed Framework

The core idea of the proposed active grasp synthesis framework can be seen in Figure 1.2. In essence, the whole active grasp synthesis process can be viewed as a loop that is made up of two cascaded optimization procedures. The first optimization is the passive grasping algorithm that generates the best grasp for the acquired data. The quality of this best grasp is sent to the second optimization procedure, the viewpoint optimization algorithm, which aims to improve the grasp quality by changing the viewpoint of the sensor.

In this work, within the proposed active grasping framework, two types of viewpoint optimization methods are designed and examined: Local search methods and global search methods. The local search methods are easy to implement, and provides efficient results. However, they have the risk of getting stuck at local optima. On the other hand, global search solutions require an offline training procedure, but they are more likely to escape from local optima and provide results that are closer to the global optimum. In both of these types, the loop is run continuously in a constant sampling rate (varies between 5 and 10 Hz in our experiments). In other words, in a constant rate a new image is acquired from the robot's sensor, the loop continues and new references are supplied to the robot. By continuous looping, efficient utilization of data is aimed.

There are the following requirements for the usage of the proposed active grasping framework: Naturally, this framework is for robots with movable sensors;



**Figure 1.2:** Scheme of the active grasp synthesis framework.

if the robot uses some fixed sensors in the environment, this framework is not applicable. Also, the passive grasp synthesis algorithm that is used in the loop should be able to supply a grasp quality value for a given data. This is usually the case, since most passive grasp synthesis algorithms in the literature are also optimization procedures that aim to detect the best grasp for a given data. In addition, in order to conduct an efficient continuous viewpoint optimization process, a fast passive grasp synthesis stage is preferable.

Of course, the grasping procedure does not end with the detection of the grasping points/regions using the visual sensors. For the realization of the synthesized grasp, the robot should approach to the detected grasping points/regions and provide a force balance between the object and its gripper. For this purpose, force control techniques are widely studied in the grasping literature (e.g. [76, 93, 146]). An alternative solution is designing compliant grippers that can mechanically obtain this force balance [92]. These solutions form a crucial component of the whole grasping procedure. However, this work is strictly concentrated to the grasp synthesis stage of grasping, and the later stages of the grasping process is outside the scope of this thesis.

It is important to note that, depending on the type of grasp (power grasp, precision grasp etc. [33]), the mechanical properties of the gripper (number of fingers, whether it is underactuated or not etc.) and the task at hand (fetching, pouring, tool using etc.) the desired locations of the grasping points/regions differ. All these properties of the grasp and the gripper put constraints on the passive grasp synthesis algorithm. However, for the viewpoint optimization algorithm, all these properties are irrelevant, since in this optimization only the output of the passive algorithm is used. Therefore, the viewpoint optimization schemes presented in this work are applicable for all the abovementioned cases. On the other hand, the workspace constraints of the robot imposes constraints on the viewpoint optimization. The ways of including these constraints to the optimization process are discussed in detail.

## 1.4 Thesis Structure

This thesis is organized as follows: First, the fundamentals of grasp synthesis are explained in Chapter 2. In that chapter, the basic concepts of grasping synthesis and the factors that affect the grasp synthesis process are clarified. Following that, a comprehensive literature study is presented in Chapter 3. There is a lack of a recent thorough literature study in the grasp synthesis field, and that chapter aims to fill that gap and explain why active algorithms are crucial to solve the grasping synthesis problem for unknown objects. In the forth chapter a simple local viewpoint optimization method is examined. With this simple implementation the basic usage of the framework is demonstrated. Following that, a more elaborate approach for local viewpoint optimization that uses extremum seeking control methods has been proposed. For that purpose, first a comparative study on extremum seeking control methods is presented in Chapter 5. Then, the most suitable extremum seeking control method for our local viewpoint optimization problem is chosen and implemented within the proposed framework in Chapter 6. In Chapter 7, a global viewpoint optimization method is presented. This method aims to generate an exploration policy by using supervised learning techniques. The steps of the training process are explained in detail, and the resulting policy is analyzed. Chapter 8 presents an application of the proposed framework to a different field i.e. active object recognition. The algorithms for active object recognition (e.g. [40, 79, 151]) deal with a similar problem to the active grasp synthesis: Instead of trying to recognize the object from one single image, these algorithms aim to lead the robot to more discriminative viewpoints which can increase the recognition rate. Unfortunately, the active object recognition algorithms in literature are not robust to structured noise (i.e. occlusions and variations in lighting conditions), and therefore, are not suitable for robots that operate in unstructured and dynamic environments. In that chapter it is shown that efficient viewpoint optimization can be conducted and high robustness against structured noise can be achieved by applying the framework proposed in this thesis. Finally in Chapter 9, the outcomes of the proposed methods and their potential improvements are discussed.



# Chapter 2

## Fundamentals of Grasp Synthesis

In this chapter, the basic concepts of grasp synthesis and their use within the context of this thesis are explained. These concepts will frequently be referred to in Chapter 3 which summarizes the grasp synthesis literature. Following that, the factors that put constraints on the grasp synthesis process are examined. The influence of each factor on the viewpoint optimization stage is also discussed.

### 2.1 Basic Concepts in Grasping Literature:

#### 2.1.1 What is a Grasp?

According to Oxford English dictionary, a grasp is “a firm hold or a grip”. This definition implies that, the (robotic) hand constraints the motion of the object by providing a force and torque balance, and that it can resist forces and torques in certain directions. It is interesting to note that there is a nuance of meaning in Cambridge dictionary. According to that dictionary, the verb grasp means “to *quickly* take something in your hand(s) and hold it firmly”. This means, according to the human perception, the grasping action is expected to be fast, and if the grasp is too slow it loses a part of its meaning.

In the grasping literature, the computation time of the grasp synthesis algorithms is considered as an important measure of success together with immobilization ability of the synthesized grasp. In this work, since the viewpoint optimization is introduced as a part of the grasp synthesis process, the efficiency of the viewpoint optimization procedure is a major concern, and the methods of systematic and efficient viewpoint optimization are targeted.

### 2.1.2 Contact Wrench and Wrench Space

Contact wrench and wrench space are important concepts which are used in the calculations of force/torque balance in the grasping literature. A contact wrench is the set of forces and torques applied by a contact point. In 2D case, a wrench is three dimensional with two force components and one torque component. In 3D case, a wrench is six dimensional with three force and three torque components. A set of all possible wrenches forms the wrench space. Therefore, each wrench can be considered as a point in the wrench space. These concepts form a basis for the force closure/form closure calculations which will be explained next.

### 2.1.3 Force Closure vs. Form Closure

Force closure is a widely used method for analyzing balance of forces and torques acting on a target object. A grasp is a force closure grasp if any arbitrary force and torque can be applied on the object via the contact points [113]. Equivalently, a force closure grasp is able to resist forces and torques in any arbitrary direction. This implies that, for a grasp to be force closure, the contact wrenches of this grasp should positively span the whole wrench space. While calculating force closure, friction of the contact surfaces are taken into account, and the frictional forces are usually represented by friction cones [113].

On the other hand, form closure is immobilizing the object without relying on the contact surface friction [11], and it is harder to achieve. In other words, it is force closure without friction.

It is important to note that, an accurate calculation of force closure necessitates the friction coefficient of the object surface to be known. Unfortunately this is usually not the case, and most of the time only an approximate force closure analysis can be conducted. In literature, there are approximations of force closure (e.g. the approximations used in [99, 138]) which are preferred for their simplicity and calculation efficiency. For the same reasons, these approximations are used in this thesis in Chapters 6 and 7.

### 2.1.4 First and Second Order Mobilities

The mobility definitions are core concepts while deciding for a grasp. These concepts are presented in [131, 132]. The first order mobility is about the force and torque balance on the object and can be covered with the conventional force and form closure concepts. It does not infer any shape properties of the object. The second order mobility is related to the surface curvature of the object and the gripper. It is shown in [132] that, grasping the object from the concave regions increases the stability of the grasp. Furthermore, [114] shows that the effect of local curvature can be approximated by a stiffness matrix whose combination with contact stiffness makes up the stiffness of the grasp. By this formulation, it is concluded that, selecting the grasp from the concave regions of the grasp

enhances the grasp stability significantly, and the reverse is valid for convexities. Therefore, considering the effects of both first and second order mobilities leads to more accurate analysis while deciding on a grasp, since considering only force/torque balance may mean neglecting some stabilizing/destabilizing effects that are caused by the surface properties of the contact.

In this thesis, the grasp synthesis methods that are used together with the viewpoint optimization algorithms consider both the first and second order mobilities, so that force closure grasps are aimed to be obtained by avoiding the convex surfaces of the object.

### 2.1.5 Grasp Synthesis

Grasp synthesis is the procedure of detecting or calculating points/regions on a target object that lead to a desired grasp according to a given set of criteria. These points/regions are called grasping points/regions.

Grasp synthesis algorithms do not necessarily provide information about how to reach the grasping points/regions. Therefore, most of the time, they need to be coupled with a path planning algorithm in order to execute the grasp. However, some grasp synthesis algorithms supply a direction for approaching the target object, i.e. an *approach vector*, which determines the orientation of the gripper.

Within the context of this thesis, we make a distinction between passive and active grasp synthesis methods. With few exceptions, the grasp synthesis algorithms in the literature analyze the data that is provided to them, and this data is mostly a single image of the target object. We call these algorithms *passive grasp synthesis* algorithms. On the other hand, with the viewpoint optimization techniques provided by this work, the viewpoint of the sensor can be altered for aiding the grasp synthesis process. Such an approach, which couples data analysis with active vision, is referred to as *active grasp synthesis* in this thesis.

### 2.1.6 Measures of Grasp Quality

Grasp quality measures are not only necessary for analyzing a resulting grasp, but they also serve as optimization criteria for grasp synthesis. These measures are particularly important for this thesis as it is a natural choice to use the grasp quality measures as viewpoint optimization criteria since the viewpoint optimization aims to increase the quality of the synthesized grasp. The viewpoint optimization framework proposed in this work can be used with any of the grasp quality measures or the combination of them. In this thesis, the measures related to force closure, surface curvature and orthogonality are used in Chapters 4, 6 and 7.

A detailed survey on grasp quality measures can be found in [155]. In this section some frequently used grasp quality measures will be pointed out.

### **Measures based on force closure approximation**

A practical way of calculating the quality of a two finger force closure grasp is presented in [99, 138]. In this method, a grasping line is defined as the line that connects the two grasping points. The magnitudes of the angles between the grasping line and force application directions determine this quality measure which is aimed to be low.

### **Measures related to surface curvature**

As mentioned previously, the concavity of the contact surface affects the stability of the grasp positively. The methods that take this effect into account uses surface curvature of the contact surface as a grasp criterion. An example can be found in [135].

### **Measures related to grasp polygon**

In wrench space, contact wrenches can be represented as points, and these points form a polygon which is called a grasp polygon. As the area of this polygon gets larger, resistance to external forces increases. By this property, grasp polygon area is used as a maximization criterion in grasp synthesis literature (e.g. [108]). The location of the polygon on the object is another grasp quality measure: As the center of the polygon gets closer to the center of mass of the object, the effect of the inertial forces on the grasp decreases. Therefore, the distance between the polygon center and the center of mass of the object can be aimed to be minimized as in [123]. The shape of the polygon is also an important factor, since it determines in which direction the grasp is more resistant to forces and torques.

### **Measures related to finger forces**

While obtaining a stable grasp, the finger forces can be aimed to be optimized. Maximum finger force and total finger force are used in the grasp synthesis literature as criteria to minimize (e.g. [108]).

### **Measures related to gripper kinematics**

As the positions of the gripper fingers are being calculated for a grasp, it is important to avoid singularities of the gripper. For this purpose, maximum singular value or combination of all the singular values can be used as a minimization criteria. An example can be found in [31].

### **Orthogonality**

The orthogonality measure is studied comprehensively in [10, 9]. In those works it is shown that, humans generally prefer to grasp an object around its major



**Figure 2.1:** An example of push grasp presented in [45]: The target object and the occluded object are separated by the pushing action to free the occluded surface of the target object.

axis, and combining this measure with other grasp quality measures increase the grasp success significantly.

## 2.2 Factors that Affect Grasp Synthesis Process

### 2.2.1 The Grasp Scenario and the Effect of the Surroundings

Especially for robots that operate in daily environments, there can be various grasping scenarios each of which introduces different constraints on the grasping process. Perhaps the most simple scenario is having one single object on a planar surface. In this case, the target is clear, segmentation of the object (if necessary) is easier and there are no occlusions by other objects. However, some grasping poses may not be applicable due to the occlusion by the supporting plane. This scenario is the most widely studied one in the literature. A more complex scenario is having one target object together with other objects in the scene. In this case, grasping the target object from certain directions may be precluded by the other objects. In both of these cases, the options of the possible grasps are restricted by the surrounding objects. These restrictions are often treated as constraints in grasp synthesis algorithms. In the grasping literature, there are manipulation strategies in the literature which can aid the grasping process and help relax these constraints. If the objects which occlude a desired grasp are movable, the “push grasp” algorithm presented in [45] can be employed. In that work, a method is proposed for separating the target object from the other objects by pushing during the reaching phase. An example of this procedure can be seen in Figure 2.1. For some thin planar objects, all the grasping options may be blocked by the table. For these cases, manipulation before grasping becomes inevitable. The *pre-grasp manipulation* strategy presented in [83] is for enabling grasps in these kind of cases. In that work, the robot pushes the target object towards the edge of the table in order to free up the occluded regions. This procedure can be seen in Figure 2.2. (Here, it is interesting to note that, sliding the object to the edge of the table is usually the last resort for humans. Humans either use their finger tips or nails, or apply force to one edge of the object to elevate the other edge.)



**Figure 2.2:** Example of a pre-grasp strategy presented in [83]: The planar object is moved to the side of the table to free the graspable region and then grasped.

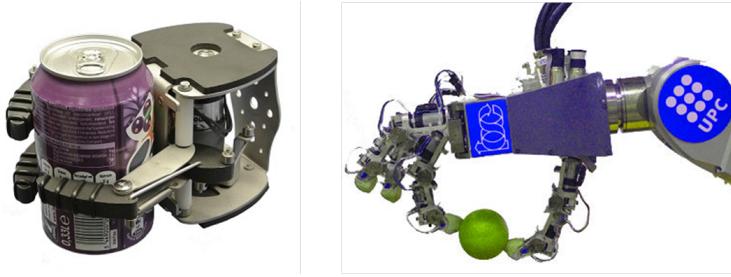
Another scenario is having multiple objects in the scene all of which are aimed to be grasped. An example is the task of emptying a box filled with objects as covered in [61]. In that scenario, the objects are not necessarily on a plane anymore, and they heavily occlude each other. On the other hand, the order of picking is not important, and it is allowed to grasp multiple objects at the same time. All these properties of the scenarios and available manipulation strategies should be considered while designing a grasp synthesis algorithm.

In this thesis, we used the scenario with a single object on a plane for testing our viewpoint optimization methods. Besides this case, in scenarios with multiple objects, the viewpoint optimization techniques can avoid viewpoints with occlusions during the optimization process by either introducing penalty terms to the objective value calculation stage (if local optimization techniques are used) or by marking the unexplored regions (if global optimization techniques are used). These implementations are discussed in Chapter 9.

## 2.2.2 Grasp Types

A taxonomy of grasp types is presented in [33]. In that work, grasps are categorized into two main groups which are power grasps and precision grasps. In power grasps, both palm and fingers are used. This type of grasps have higher contact area between the object and the gripper, and they are preferred when high stability is needed. On the other hand, precision grasps use finger tips, and they are preferable when high sensitivity is necessary for the grasp location and/or a precise manipulation of the object is required. Examples of these grasps can be seen in Figure 2.3.

Both power and precision grasps are divided into many subcategories depending on the finger and palm positions. Perhaps, the most commonly used subcategory for precision grasps is planar grasps in which all the contact points lie on the same plane. This grasp type is used for grasping planar (2D) objects. The other subcategories are rarely studied in the literature (an exception can be found in [51]), and grasp synthesis algorithms are generally designed for obtaining either a power or a precision grasp.



**Figure 2.3:** Power grasp vs. precision grasp: At left, the Fetch Hand [92] by Lacquey Robotic Grasping Solutions performing a power grasp, at right the MA-I hand [154] performing a precision grasp.

### 2.2.3 Gripper Types

Kinematics and actuation mechanism of a gripper are important parameters for the grasp synthesis and execution stages. Shape of a gripper can vary from complex anthropomorphic hands to simple parallel jaw grippers or even to grippers without fingers. Anthropomorphic hands are grippers with five or four fingers whose configuration is similar to the human hand. Two examples, the Shadow Dexterous Hand and the IH2 Azzurra Hand can be seen in Figures 2.4a and 2.4b respectively. Examples of simpler three finger grippers, the Servo-electric 3-Finger Gripping hand (SDH) and the Fetch Hand, are presented in Figures 2.4c and 2.4d. A commonly used gripper type, a 2-jaw parallel gripper which is made up of two fingers closing to each other, is given in Figure 2.4e. The Universal Gripper in Figure 2.4f stands out with its unique design without fingers. This gripper is a compliant ball that squeezes the object.

Considering the actuation mechanism, the Shadow Dexterous Hand is a fully actuated gripper, so that each joint of each finger can move independently. Despite having a similar look, the IH2 Azzurra Hand has an underactuated mechanism with one actuation per finger. The fingers of this gripper can only contract towards the palm and relax. The Fetch Hand is also an example of an underactuated gripper: It has only one actuator to contract all fingers. The 2-jaw parallel gripper is an underactuated gripper as well, since the parallel plates move towards each other with a single actuator.

Grippers with different kinematics and actuation mechanisms have different grasping capabilities. Grippers with full actuation have more freedom while positioning the fingers on the object, thus provide more possibilities for precision type grasps. On the other hand, they necessitate force control algorithms to provide force and torque balance on the object. For underactuated grippers, the motion of the fingers has constraints which makes them more suitable for power grasps rather than precision grasps. However, force balance can be maintained by the mechanism itself without even using a force control algorithm (e.g. for the IH2



**Figure 2.4:** Several service robots from today: (a) Shadow Dexterous Hand [87] by Shadow Robot Company, (b) IH2 Azzurra hand by Prensilia s.r.l., (c) Servo-electric 3-Finger Gripping Hand (SDH) by SCHUNK, (d) Fetch Hand [92] by Lacquey Robot Grasping Solutions, (e) a 2-jaw parallel gripper by SCHUNK, (f) Universal Gripper [21] by University of Chicago, Cornell University, iRobot, and Defense Advanced Research Projects Agency DARPA

Azzurra Hand and the Fetch Hand) which makes the grasp execution process easier.

All these abilities/restrictions of grippers should be considered while designing a grasp synthesis algorithm. Some algorithms in the literature prefer to provide solutions for a specific type of gripper whereas, some algorithms provide methodologies that can be applied to various types of grippers.

Here, it is important to note that, in the grasping literature, the importance of compliance is stressed in many publications [114]. The compliance can be maintained by the materials used in the contact points, actuation mechanisms or force control algorithms. Although high compliance may bring some precision issues, it increases the possibility of a successful grasp significantly.

#### 2.2.4 2D Objects vs. 3D Objects

Although there is no explicit definition in the literature, the objects which are thin (one dimension is significantly small) are generally referred to as 2D objects. In most cases, these objects are assumed to lie on a plane, and planar precision grasps are aimed to be synthesized. On the other hand, the objects which have considerable volume in all three dimensions are called 3D objects. In this thesis both 2D and 3D objects are used in the experiments.

#### 2.2.5 Grasping considering the task and its formulation

Grasping is never a sole goal by itself; it is just a medium in order to achieve a higher level task. In grasp synthesis process, the task at hand is a major deterministic factor. In the grasping literature, the most commonly addressed tasks are lifting and fetching the object (actually they are so common that, many papers in literature do not even mention the task if it is lifting or fetching). If the task is one of these two, then the forces to resist are gravitational force and inertial forces due to the robot's motion. However, there can be many different tasks for a service robot like pouring the liquid contained in an object, opening/closing the lid of an object or using an object as a tool e.g. using a hammer, using a key etc. Each of these tasks requires the grasp to be located in certain regions of the object. For example, if the task is pouring, the robot should leave the opening of the object free; if the robot wants to use a hammer, it should grasp it from its handle, etc. All these restrictions and preferences for grasp locations can be interpreted as regional constraints by the grasp synthesis algorithm. A method for modeling and learning such constraints is presented in [150].

Apart from the regional constraints, tasks also necessitate the grasps to resist forces and torques in certain directions. For instance, if a robot is needed to use a hammer, then the grasps should resist the forces and torques expected on the hammer's head. The work presented in [19] directly addresses this problem. It claims that, each task implies a custom polygon in the wrench space, and this polygon can be used as an optimization criterion.

It should be noted that for the case of grasping an unknown object, the possible tasks are quite limited. For example, the robot cannot synthesize a grasps by considering how to use a tool without knowing what it is grasping. For this case, the potential tasks are reduced to fetching and lifting. Still such an ability enable the robot to tidy up a room, unload a basket, fetching a requested object to the user, removing debris in a disaster area etc.

In the implementations of this thesis, only the lifting task is aimed while synthesizing a grasp. However, the proposed viewpoint optimization framework is not task dependent: Task constraints affect the passive grasp synthesis stage by limiting the set of possible grasps. The viewpoint optimization uses only the output of the passive grasp synthesis stage, and is not affected by the internal decision process of the passive stage.

### **2.2.6 Object being known/familiar/unknown**

The concepts “known object”, “familiar object” and “unknown object” are related with the amount of a priory information about the target object. Although, they sound like simple and straightforward concepts, the meanings change depending on the stage of the grasping process.

A known object in grasping context is an object whose full model is available to the robot prior to the process. The robot may even already know how to grasp the object. In that case, the grasping process is reduced to path planing and object stabilization, and no grasp synthesis is needed. If the robot does not know how to grasp the object, grasp synthesis algorithms for known objects are utilized to detect the grasping points/regions.

In the case of an unknown target object, a full model of the object is not available to the robot prior to the grasping process. There are basically two options in this case: The first option is to generate a full model of the object by acquiring images from several viewpoints. As the model is acquired, grasp synthesis algorithms for known objects can be used. The second option is to use grasp synthesis algorithms for unknown objects which do not rely on a full model of the object; they use partial shape information obtained by its sensors. In the first option, although the object is unknown considering the whole grasping process, it is a known object from the grasp synthesis point of view since the model is available. In the second option, both for the whole grasping process and the grasp synthesis stage, the object is unknown. The framework presented in this thesis is designed for this second case.

For grasping 2D objects, the grasp synthesis algorithms use the object boundary that is extracted from a single image captured from the top. This boundary can be considered as a full model of the object since the object is 2D. Therefore, all the grasp synthesis algorithms for 2D objects can be considered as algorithms for known objects even though the robot has no information about the object shape prior to the grasping process.

The definition of a familiar object lies between the known and unknown object

concepts. Here the word familiar implies that, the shape of the target object is similar to an object which the robot can successfully synthesize a grasp: For example, the robot may have trained to grasp mugs. When it aims to grasp a mug whose shape is different than the shapes in the training set, it can still achieve a successful grasp if the training algorithm can generalize the synthesis procedure good enough.



# Chapter 3

## Grasp Synthesis Literature

This chapter gives a comprehensive overview of the grasp synthesis algorithms in the literature. It aims to clarify which portions of the grasp synthesis problem are solved, and which issues still necessitate further research. This overview will help to see why active grasp synthesis algorithms are needed, and how they can contribute to the solution of the robotic grasping problem.

As can be seen from Chapter 2, robotic grasping is a complex problem with many aspects: Shape of the object, a priori information about the shape, state of the environment, abilities of the robot, abilities of the gripper and task-at-hand are all important factors that influence the grasping process and in particular the grasp synthesis stage. The algorithms in the literature address this challenging problem by employing various strategies and grasping criteria, and, most of the time, by also making assumptions on object shape, gripper type and/or grasp type. While analyzing an algorithm in this chapter, it is explicitly stated which grasping criteria are used and what kind of assumptions are made. Knowing the assumptions are essential for determining in which conditions the grasp synthesis algorithm is valid. Besides these aspects, the majority of the algorithms deal with synthesizing a grasp for the case of having one single object on a plane and for the task of lifting or fetching. Therefore, the scenario and task at hand are not mentioned explicitly, unless they are different from these cases.

In this chapter, the grasp synthesis algorithms are examined in three main categories: Algorithms for known objects, familiar objects and unknown objects. Even though the framework presented in this thesis is only for the grasp synthesis algorithms for unknown objects, an insight on the algorithms for known and familiar objects is also useful since they form a basis for some of the algorithms for unknown objects. After the analysis of the literature, the chapter is concluded with a discussion about the difficulties that the current algorithms face and the role of the active grasp synthesis algorithms for solving these difficulties.

### 3.1 Grasp Synthesis Algorithms for Known Objects

The earlier algorithms in the grasp synthesis literature dealt with synthesizing a grasp when a full shape model of the target object is available, and these algorithms formed a basis to the harder problem, the grasp synthesis problem for unknown objects. Knowing a model of the target object helps substantially while solving the grasp synthesis problem. By using models, various optimization techniques can be utilized to search for the optimum grasp for given criteria. A survey on the theoretical developments in this area can be found in [11].

Perhaps the simplest objects to synthesize a precision grasp are 2D polygonal objects. Assuming the edges of the polygon are extracted with a vision sensor, four-finger form closure grasps and two-finger force closure grasps can be synthesized by applying the procedures presented in [113]. For four-finger grasps, a parallelogram based method is used, and for two-finger grasps friction cones are utilized in the grasp stability calculations. For the case of a curved 2D object, an algorithm is proposed in [58] for two-finger force closure precision grasps. In this method, the boundary of the object is modelled part by part with parametric curves, and each combination of these parts are recorded as grasp candidates. Then, stable grasps are chosen with the same friction cone based method with [113]. Another method that uses parametric representation of the object boundary is given in [135]. In that work, the whole boundary is modelled using Elliptic Fourier Descriptors (EFD), and grasping point locations are optimized considering the local curvature of the model. Local curvature is also considered in [81] for two-finger precision grasps. In that work, the boundary curve is decomposed into concave and convex regions, and each region pair is analyzed for detecting stable grasps. Similarly, a method for two finger grasp that uses surface curvature is presented in [26] which defines and optimizes an energy function for the grasping point locations. In that formulation, stable grasps for both convex and concave regions can be found by searching for the critical points of the energy function. For multi-finger force closure grasps, an approach is presented in [60]. That work proposes methods to optimize the total and maximum finger forces. There are also methods that use heuristics such as [138] in which the grasp is synthesized according to the surface curvature, distance to the centroid and angle between grasping points. A method that uses desired wrenches to resist as optimization criteria is given in [108] for multi-finger force closure precision grasps. Such a formulation is especially useful for task based grasp synthesis where resistance in certain directions are needed for the task at hand. An algorithm for two finger precision grasps that also considers the holes of the object and can design *expansion grasps* is given in [110]. All these algorithms synthesizing a precision grasp for 2D objects provide reliable solutions for different kinds of grippers and optimization criteria. Naturally, the abilities of these algorithms are quite limited when they are applied to grasping of 3D objects. In this thesis, the EFD based 2D grasp synthesis algorithm in [135] is coupled with a viewpoint optimization stage in Chapters 4 and 6, and it is shown that, even with such a 2D algorithm which

has limited abilities for grasping 3D objects, high success rates can be obtained when good viewpoints are supplied to the algorithm. This shows the importance of the viewpoint optimization for improving the abilities of the offline grasp synthesis methods.

When the target object is 3D, the grasping process becomes more challenging. First, the grasp synthesis problem becomes harder due to the added dimension. Second, the accuracy of the model at hand influences the grasp synthesis thus the success of the grasp significantly. Third, after the grasp synthesis, the relative pose of the object with respect to the robot should be estimated in order to execute the grasp, and the errors in this estimation affect the success of the grasp.

Assuming that the object model and the pose estimation are accurate, the above-mentioned algorithms [26, 60, 108] can be extended for grasping 3D objects by adapting the same principles to 3D. Besides these methods, several other techniques are proposed in the literature. A common technique that is used in grasp synthesis algorithms is model simplification. In [107], the object models are simplified with shape primitives like spheres, cylinders, cones and boxes. Then, the grasps are determined on those simplified shapes by heuristics. Another method that is based on simplification of the object model is presented in [80]. In that method, the object model is decomposed into sub-components each of which are approximated with a box. Then, a grasp is chosen via a set of heuristics using the faces of the boxes. In [66], the object is similarly decomposed into sub-components and each component is modelled using superquadrics. Then, many grasps are generated by some heuristics using the Graspit! simulator [106], and best grasp is chosen. The abovementioned methods based on model simplifications do not necessitate an accurate object model. However, as the object shape becomes more complex, the performance of the algorithm is affected negatively. Among these methods, the method in [66] generates more detailed models, since it has both shape decomposition and more accurate modeling of sub-components comparing to the whole body approximation or the box based approximation methods.

A framework that explicitly deals with model and pose estimation inaccuracies in the grasp synthesis stage is presented in [54]. In that work, some gripper preshapes are learnt by human demonstration via a glove. These preshapes reduce the possible grasps, therefore make the optimization of the grasping point locations easier by reducing the search space. Then, the grasp that are more likely to be successful under model and pose estimation inaccuracies are synthesized. Another method that is concerned with model accuracy is given in [104]. In that case, a parallel architecture is presented in which model refinement and grasp synthesis are conducted simultaneously. In that work planar force closure grasps are synthesized on 3D objects.

Learning algorithms are also utilized for the grasp synthesis problem of known objects. In [120], the grasp synthesis algorithm is trained with synthetic superquadric object models, and by this way, the grasp parameters (gripper pose) are associated with the model parameters. Therefore, as a new object is aimed to be grasped, a grasp can be synthesized directly from its superquadric model. In

[137], an active learning algorithm is employed for learning the approach vector for grasping by again using superquadric representations of the object. In [63], the shape properties (mainly principle axes) are mapped with an approach vector using neural networks. The data necessary for the training is obtained from experiments with humans.

Learning frameworks are also helpful for embedding task requirements to the grasp synthesis process. [25] provides a framework for combining task representations with object representations. This method uses discrete views of the object model (namely front, back, left, right, top and bottom), represents these views with simple shapes and combines them to task representations with neural networks. In [55], similar to the previously mentioned algorithm [66], a method that decomposes the object model and generates superquadric models of the sub-components is proposed. However, different from [66], which sub-component to grasp is learnt from manually labeled synthetic data. The synthetic data is labeled by asking humans to choose a sub-component to grasp for using the object in a task. Therefore, the output of the system is more likely to generate grasps that are suitable for a given task. On the other hand, different tasks are not considered separately in this work, since labels are generated without giving a task definition.

Apart from the abovementioned algorithms, an approach for four-finger grasps is proposed in [115]. In this work, the object model is sampled and searched via a heuristic for force closure grasps using surface normals. In this case, the model should be accurate enough to calculate surface normals reliably. In [126], the object model is represented with a grid of spheres. Then, principle component analysis (PCA) is applied on this model to detect the symmetry axes. After that stage, grasps are synthesized using this information and the stability of them is checked with a force closure analysis. A method for power grasps is presented in [103]. In that method, virtual cords are wrapped on the object model and analysed for detecting stable grasps. The advantage of this method is its efficiency since it does not require model preprocessing. A methodology for handling the pose and shape uncertainty using partially observable Markov decision processes (POMDPs) is presented in [78]. Although presenting an interesting methodology, the results of this paper can be considered as preliminary since very simple objects are considered.

As mentioned previously, for the case of an unknown object, some algorithms propose the use of automated modeling techniques to generate a full 3D model of the object prior to grasp synthesis. Since a full model is acquired before the grasp synthesis stage, these algorithms should be considered in the know object category when the grasp synthesis process is considered. Examples of a such strategy are presented in [100, 168, 175]. In [100], after the modeling stage, the sum of finger forces are optimized using a generic algorithm. In [175], the optimal grasp is found by minimizing an objective function which is made up of contact area, gravitational effect manipulability and motion cost components. [168] minimum resistible wrench is maximized by using a simulation environment.

Next, the grasp synthesis algorithms for familiar objects are examined.

## 3.2 Grasp Synthesis Algorithms for Familiar Objects

The grasp synthesis algorithms for familiar objects aim to generate grasps without relying on a full model by utilizing shape similarities of the target object to the previously examined objects. An example of such algorithms is presented in [13, 14]. In that work, shape context approach is used to represent the global contour of the object. Using this representation the system is trained via supervised learning using synthetic images. The goal of the training is to obtain a grasp synthesis policy for a given shape context model.

In [68], a data driven approach is proposed. In this approach, the Columbia Grasp Database [67], which is a huge database of object models associated with desired grasps, is used. By a partial range image of a target object, the most similar object in the database is found, and associated grasp is used.

A template based approach is presented in [74]. In this approach, template models of the objects are generated by using depth images, and grasp are associated with the models using a programming by demonstration technique. As a new object is aimed to be grasped, the template model is generated and the best match is found from the template database. After the matching, the demonstrated grasp is executed.

Another method which uses sub-components of the object is presented in [43]. First using teleoperation, some prototype grasps are taught to the robot. The grasps are associated with the models of the sub-components that the grasp is executed at. When a new object is targeted to be grasped, the sub-components are searched on the depth image of the object, and grasps are applied to the detected parts.

For the success of these kind of approaches, there are two major issues. First, the shape representation should be descriptive enough to discriminate between the objects that need to be grasped differently. Second issue is the amount of training data. As the training data covers more and more cases, the possibility of generating successful grasps for higher variety of objects increases. By choosing a good descriptive model and having a large training data set, these kind of algorithms can be used for grasping unknown objects.

## 3.3 Grasp Synthesis Algorithms for Unknown Objects

Grasp synthesis algorithms for unknown objects aim to generate desired grasps without relying on a full model of the target object. By employing various strategies, these algorithms aim to cope with missing shape information and noisy sensor data while synthesizing a grasp. Commonly used sensors are cameras, depth image sensors (e.g. laser scanner, time-of-flight camera) and RGB-D sensors which supply colored images combined with depth information (e.g. Kinect, stereo camera).

One of the strategies in the literature is to simplify the grasp synthesis problem

by considering only a certain set of grasps. The method presented in [129] adopts such a strategy and aims to grasp the target objects from their top surfaces. In that work, the scenario of having multiple target objects on a table is considered, and two-finger precision grasps are synthesized using a depth image. The first stage of the algorithm is the segmentation of the target objects. Then, top surfaces of each object are extracted, and the grasping points locations are determined to be close to the center of mass of these surfaces (alternatively, the abovementioned 2D grasp synthesis algorithms can also be utilized). This method is extended for considering the handles and top openings of the target objects in [130]. In addition to top grasps, the work presented in [76] also considers side grasps and high point grasps (for bowl-like objects that have opening on the top). In that work, an estimation of the object's principle axes is calculated using a depth image. Then a grasp is selected from the defined grasp set and executed using some heuristics. The algorithm is also supported with a contact-reactive method that stabilizes the grasp using tactile information. This method is applied to two finger precision grasp of the target objects. However, by defining the grasp set and heuristics accordingly, it can also be applied to other types of grippers.

Data simplification (like [107] for known objects) is another strategy to approach to this grasp synthesis problem. [64] followed this strategy by fitting shape primitives to a depth image of the target object. Then, previously defined grasps that are assigned to the corresponding shape are executed. Another simplification based method is given in [32] in which a knowledge base is utilized. For building the knowledge base, an automated procedure is followed by using the GraspIt! simulator: 3D models of the training objects are loaded to the simulator and oriented bounding boxes are fit to those models. Then, grasps are generated in the simulator using some heuristics. The knowledge base is built up with bounding box models and associated high quality grasps. When a new object is aimed to be grasped, an oriented bounding box is fit to the object's depth data, and the best grasp that corresponds to a similar bounding box is selected from the database. In these methods, since grasps are synthesized using the simplified model of the object, the error between the simplified model and the actual shape affects the performance of such algorithms negatively.

On the other hand, some methods make explicit assumptions on the shape of the object, so that missing shape information can be completed. A method for symmetric objects is given in [12]. By using an RGB-D image of the scene, the missing data of the object is completed by symmetry. Then, a grasp and a collision free path to the grasping pose are generated using a simulator called OpenGrasp [101].

For some scenarios, modeling the whole scene can be more useful than dealing with each object separately. For the scenario of emptying a box, such an approach is proposed in [61]. By using a depth image, this method models the whole scene by using Height Accumulated Features (HAF). The system is trained with support vector machines (SVMs) for identifying good grasps for a given HAF model. The system is capable of grasping the objects in complex scenarios, but objects used

in the test set are rather simple.

Feature based strategies make up a significant portion of the grasping literature. An approach that uses 2D image features is proposed in [141, 142]. In that work, image segments that correspond to good grasping regions on the object are aimed to be extracted. For that purpose a set of features are defined, and a feature vector is constructed. Using this vector and labeled synthetics images (images of the objects with marked grasping regions), the system is trained via supervised learning. This approach does not necessitate segmentation of the target object. The output of this algorithm is several regions on the image that are suitable for grasping, and the 3D locations of these regions are calculated using triangulation. Designing a grasp using these regions is necessary to execute the grasp. Furthermore, gripper kinematics are not considered in grasp synthesis. The algorithm is applied to the task of lifting the target object as well as unloading a dishwasher. A similar approach with RGB-D images is presented in [143]. In that work two classifiers are used: one for the success probability of a grasp which is calculated using the depth features, and the second for the probability of executing this grasp successfully with the robot's gripper. The combination of these probabilities serves as a quality measure for the grasp synthesis, and the system is trained with this measure using supervised learning in order to come up with a policy that bridges features with grasps. The output of this algorithm is both the grasping regions and the full gripper pose. Therefore, less work is left for the planning of the synthesized grasp comparing to the previous work. Another feature based method is presented in [99]. In that work, several features are defined using both depth and intensity image information. A weighted sum of these features is used as the quality measure for detecting the grasping regions, and the weights are determined using supervised learning. Another learning based method is presented in [127] for the case of having multiple target objects in the scene. By using supervised learning, an algorithm that can differentiate between graspable and ungraspable regions is trained. This system is then used on the segmented object data. A 3D feature based method that does not require training is presented in [86]. In that approach, the depth image is searched for the object segments that are suitable to the opening of a two-finger gripper. The criteria for the search are the alignment of the contact normal vectors and having sufficient grasp depth and spread. This algorithm can work in a scenario with multiple target objects with occlusions. A similar method that searches for suitable gripper opening is presented in [82]. In that work, supervised learning is used for the search of this 3D feature. An approach that combine 2D and 3D features for two finger grasps is presented in [125]. In that work, four grasp types are defined. Using co-planarities and co-colority, the edges of the object are detected and grouped. Then, suitable predefined grasps are executed for the grouped edges. This approach is further extended in [124] for considering also the surface based grasps together with the edge based grasps.

Reinforcement learning is also proposed as a solution of this grasping problem. [8] presents a reinforcement learning based methodology which uses tactile inform-

ation rather than visual data. In that work, an action set is defined as translations and rotations in x, y and z axes together with the alteration of the spread angle and execution of the grasp. Using these actions and detecting whether a contact is occurred or not, an action sequence that leads to a successful grasp is learnt via training. This work is only implemented for simple objects, and can be considered as a preliminary work.

All the abovementioned algorithms except [76] do not propose a methodology for using the robot motion in order to increase the probability of the grasp success. The approach in [76] is different in this sense, since it is supported with a contact-reactive method by which the gripper pose is altered using tactile information in order to achieve a stable grasp. Another contact-reactive approach is presented in [144]. In that work, The object is aimed to be grasped around the major axis. The principle axes of the object is calculated roughly from camera images. The gripper is then aligned with the major axis and the execution of the grasp begins. During the execution, if an unwanted contact is detected, the gripper retreats, corrects its pose, and approaches again. These correction movements continue until a desired grasp is achieved. (In the same work, robot is also used actively for the segmentation of the object. In order to verify the image based segmentation result, the object is pushed by the robot, and it is observed if the segmented points move together. If that is not the case, then segmentation result is corrected accordingly.) For contact-reactive methods, it is important to be able to measure the stability of the grasp for a given set of tactile measurements and hand kinematics. The method proposed in [36], which trains a classifier for this purpose, can be used in these applications. Another reactive approach that uses proximity sensors is presented in [77]. In that work, by using the information coming from the proximity sensors placed at the fingertips of the gripper, the finger poses are tuned in order to achieve higher stability grasps. Using the robot actively in the grasping process with such reactive strategies has a clear advantage: it gives the system an opportunity to correct the grasp on the fly and improve its stability. This kind of reactive strategy can be integrated to all the abovementioned passive algorithms to improve their success rate.

It is important to note that, even though both reactive strategies and viewpoint optimization framework presented in this thesis use the robot actively, their role in the grasping process are totally different. The reactive methods help the grasping process after the grasp synthesis stage; they aid the process during the execution of the already synthesized grasp by altering the gripper pose. On the other hand, the viewpoint optimization framework addresses the utilization of robot motion during grasp synthesis. It aims to provide good viewpoint to the passive grasp synthesis algorithm in order to improve its success.

In the grasp synthesis literature for unknown objects, the only method that uses robot actively in grasp synthesis is presented in [159]. In that work, an active vision based grasp synthesis strategy for two-finger antipodal grasps is presented for the task of potatoes picking from a pile. First, the topmost potato is segmented and its contour is extracted. Following that grasping points are

detected on the contour, and the viewpoint of the sensor is changed in such a way that the relative depth of the grasping points converges to zero. Despite presenting a very interesting method based on active vision, this approach is designed for a specific grasp type and task, and it is not applicable to other grasp types, grasp criteria or tasks. Moreover, the active vision algorithm proposed in that work is only valid if the grasping points are at the convex surfaces of the object.

During grasp synthesis, if multiple sensors are used for gathering information about the object shape, a methodology for combining the information from different sources is needed. For that purpose, the framework proposed in [47] can be used. In that work, the object shape is modeled using Gaussians, and such a representation enables fusing information obtained from multiple sensors.

## 3.4 Conclusions

The methods of grasp synthesis are examined in this chapter for known, familiar and unknown objects. In this section, the solved aspects and open problems of the grasp synthesis literature are summarized and discussed. The need for viewpoint optimization is also justified for the grasp synthesis problem for unknown objects.

### 3.4.1 Remarks on the Algorithms for Known Objects:

For known objects, assuming that

1. a precise model of the object exists,
2. pose of the object can be determined precisely,
3. no occlusion exists,
4. no pre-grasp manipulation is needed,

reliable solutions exist in literature for various grasping criteria. When there is pose and model uncertainty, the abovementioned papers [54, 104] provide useful frameworks to handle these uncertainties. Assuming no pre-grasp manipulation is needed, occlusion problems can be solved in the path planning stage using off-the-shelf path planning algorithms. However, pre-grasp manipulation is a topic that is not addressed thoroughly in the grasping literature. Although the push grasp algorithm proposed in [45] and the pre-grasp manipulation strategy presented in [83] provide novel and interesting solutions, the performance of these algorithms are not studied widely in the literature. Also, other kind of pre-grasp manipulation strategies are yet to be discovered.

### 3.4.2 Remarks on the Algorithms for Familiar Objects:

The performance of the grasp synthesis algorithms for familiar objects mainly relies on the amount of the training data and the descriptiveness of the models

that are used in the training process. As sufficient data is supplied to the training algorithm together with a descriptive model, the success rate of these algorithm increases and converges to the ability to grasp arbitrary shaped (unknown) objects. Even though the performance of these strategies has been demonstrated in the literature, which modelling technique is the best for the learning process and which training method gives the best results are questions that are still needed to be addressed and can be answered with a comparative study.

### **3.4.3 Remarks on Algorithms for Unknown Objects:**

The main problem in the grasp synthesis literature for unknown objects is that none of the algorithms can provide a solution that is generalizable for all the objects and for all the viewpoints of the sensor. This thesis provides solutions to this problem by introducing viewpoint optimization to the grasp synthesis process. Besides this problem, there are two other major problems in the field: Difficulty in measuring/comparing the performance of the algorithms and lack of studies on the integration methods. All these issues are discussed below.

#### **Lack of a generalizable solution for all objects**

As we look closer to the methods for grasping unknown objects, it can be seen that they rely on various (implicit or explicit) assumptions: The algorithms that simplify object models assume that simple models can represent object shape good enough. The algorithm in [12] assumes that the object is symmetric. Most feature based algorithms assume that the object is observed from a specific viewpoint. All these assumptions are for handling the missing information of target object's shape. However, in none of the papers in the literature, 100% success rate is reported. This shows that, these assumptions do not always hold even for a limited set of objects. Nevertheless, most of the algorithms do not have 0% successs rate for a given object. Even if the shape of the object is challenging for a grasp synthesis algorithm, the algorithm can still be successful for some viewpoints of the sensor. This shows that, if good viewpoints are supplied to these algorithms, they become more reliable. Unfortunately, all the abovementioned algorithms are passive grasp synthesis algorithms which do not have the ability to choose the viewpoint; they only passively analyze the information that is supplied to them. In the literature, there is no study that addresses the viewpoint optimization problem for the grasp synthesis process. This thesis aims to fill this gap by providing methods for altering the viewpoint of the sensor in order to provide better information to the grasp synthesis algorithms and boost their abilities.

The presented viewpoint optimization framework is applicable to any of the abovementioned passive grasp synthesis algorithm that can provide a grasp quality measure. The methods presented in this framework utilize this quality measure, and maximize it by changing the viewpoint. Nevertheless, in order to conduct efficient viewpoint optimization, the passive grasp synthesis stage is needed to

be fast, since this stage is used as a building block of the continuous loop. The execution times of the local and global viewpoint optimization methods presented in this thesis are negligible compared to the passive grasp synthesis stage.

### **Difficulty in Measuring/Comparing the Performance of the Algorithms**

For grasping unknown objects, various strategies in the literature are presented, and, it is important to note that, it is very hard to evaluate or compare the performance of these algorithms: In most of the publications, the proposed strategies are evaluated with experiments by using a set of objects, and success percentages are presented either per object or for the overall set. Even though these percentages give an insight about the success of the algorithms, they do not mean much for comparison purposes since

1. object sets that are used in the experiments are different,
2. relative pose of the object with respect to robot's sensor, in other words, the viewpoint of the sensor is not presented.

These two reasons prevent to maintain a common ground for the experiments; hence the comparison of the algorithms by the presented results becomes impossible. Moreover, lacking a common ground also makes it more difficult to analyze the grasping field and to detect the weaknesses of the proposed solutions.

All these reasons bring the necessity of a benchmark for the problem of grasping unknown objects. Large object model databases are already available (e.g. Columbia Grasp Database [67]), and these databases can be used as a benchmark for the problem of grasping known objects. However, a model database is not sufficient to built a benchmark for the problem of grasping unknown objects. The sensor viewpoint should also be included in the benchmark since it directly determines the missing data. A benchmark with an object set and viewpoints would contribute to the progress of this field significantly.

With such a benchmark, it will be possible to show the contribution of the viewpoint optimization framework to the failed cases of the algorithms. It will also make it easier to compare the viewpoint optimization methods that will be developed in the future.

### **Lack of studies for the integration of methods**

Although there are algorithms in the literature for various stages of the grasping process, the integration of these stages is not addressed thoroughly. Grasp synthesis is one of those stages and should work in harmony with the rest of the stages in order to achieve successful grasps for the majority of the objects and scenarios: The viewpoint optimization methods presented in this thesis aim to improve the success of the passive grasping algorithms. The reactive algorithms presented in [36, 76, 77, 144] propose methods for the use of tactile and proximity

information during the execution of the grasp in order to increase the probability of a successful grasp. The pre-grasp strategies in [45, 83] ease the grasping problem by manipulating objects. Nevertheless, the best combination of passive grasp synthesis algorithms and active/reactive methods is still unknown.

In this thesis, it is shown that, the viewpoint optimization is a crucial stage for the success of the grasping process. However, the best combination of a passive grasp synthesis algorithm and a viewpoint optimization method is still an open question.

# Chapter 4

## Grasping of Unknown Objects via Curvature Maximization using Active Vision

In this chapter, the viewpoint optimization framework is applied with a simple local viewpoint optimization method. Compared to the global viewpoint optimization method presented in Chapter 7, local methods are easier to implement as they do not require any training process, but they have a higher risk of getting stuck at local minima. The application in this chapter will help to explain the working mechanisms and basic principles of the viewpoint optimization framework. After this simple example, more elaborate local search solutions will be investigated in Chapters 5 and 6.

This work was published in the proceedings of the IEEE/RJS International Conference on Intelligent Robots and Systems (IROS) which was held in 2011 [23].

### Abstract

Grasping unknown objects is a crucial necessity for robots that operate in unstructured environment. In this paper, we propose a novel grasping algorithm that uses active vision as basis. The algorithm uses the curvature information obtained from the silhouette of the object. By maximizing the curvature value, the pose of the robot is updated and a suitable grasping configuration is achieved. The algorithm has certain advantages over the existing methods: It does not require a 3D model of the object to be extracted, and it does not rely on any knowledge base obtained offline. This leads to a faster and still reliable grasping of the target

object in 3D. The performance of the algorithm is examined by simulations and experiments, and successful results are obtained.

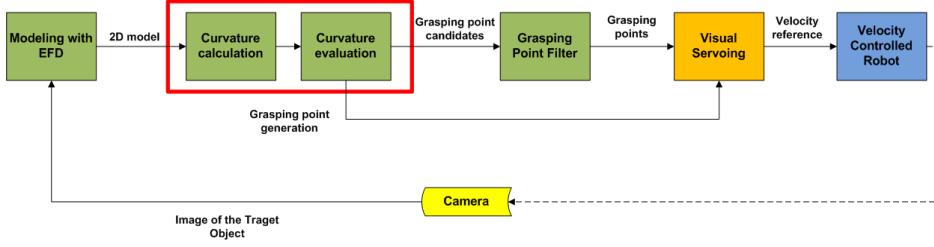
## 4.1 Introduction

Unlike their industrial counterparts, robots that operate in unstructured environments (i.e. homes, offices, etc.) may encounter with vast variety of previously unknown objects. Manipulation of these objects starts with grasping them. However, obtaining a stable grasp for unknown objects is not a trivial task.

To achieve a successful grasp, two problems should be solved: determining the way to approach the target object (determining the approach vector) and positioning the fingers of the robot hand on the target object (detecting grasping points). To solve these problems, some methods in literature analyze the 3D model of the object. In [105], a 3D model is used to calculate the center of mass and axes of inertia of the target object, and grasping points are generated such that the gripper covers the object around the center of mass. [100] uses a genetic algorithm to search for grasping points on the 3D model of the object. In [173] and [175], a cost function is employed to analyze the 3D model for grasping points. [107] and [32] simplify the 3D model into some shape primitives (boxes, cones cylinders etc.). Then, grasping points which are assigned offline to these shape primitives are selected for the corresponding shape. [129] and [130] present a method for grasping the target object from the top. This method obtains a 2 1/2 D model of the object by using a camera and a range scanner. Then, top planes are extracted from this model and a heuristic rule is applied to find grasping points.

Naturally, these algorithms have a preparation step that is extracting the 3D model of the object. However, this may be time consuming and impractical for many robotic applications. Furthermore, it may not be possible for some cases to acquire images that are necessary to build up a 3D model of the object (i.e. objects on a shelf). Even if the 3D model can be extracted, the performance of these algorithms rely on the quality of the 3D model; modeling errors that are caused by segmentation errors, noise etc. may cause bad grasping points. In addition to that, since obtaining the 3D model of the object and processing this data are computationally costly processes, these algorithms are run once before the execution of the grasp, and 3D coordinates of the grasping points are calculated to be used as position references to the manipulator. This may lead to some robustness problems since position references may have errors which are caused by the reconstruction algorithm.

Other approaches use a 2D model of the target object. [121] uses pressure snakes to model the contours of the object. Curvature information obtained from this model is used to generate grasping points. This algorithm is specifically designed to grasp almost flat objects from the top and flat surfaces which may be suitable for grasping are not analyzed in this method. In [13], a 2D model of the object is extracted using a shape context approach. This model is converted



**Figure 4.1:** *Control Loop*

into log-polar form to obtain histogram data. Grasping points are then identified by comparing the histogram data to the training data that is obtained offline. A feature based approach is followed in [142] and [141] to obtain grasping points. In this work, a supervised learning algorithm is applied to synthetic data to identify image features which correspond to object segments that are suitable for grasping. These features are then searched on the target object. Since offline training data makes up the main structure of the last two algorithms, the quality of the grasping point generation is heavily dependent on how much of these data span the possible cases.

In this section, we propose a novel grasping algorithm that is fast, robust and does not necessitate any offline training data or 3D model of the object. The algorithm is designed for eye in hand systems and uses the curvature information of the silhouette of the object to update the pose of the robot using active vision to obtain a good grasping configuration. Again using the curvature information, grasping points are generated for the gripper contacts. Due to the low computational cost, the algorithm can be used within the control loop.

Being able to generate grasping points online brings robustness to the system comparing to the methods that use 3D models. This is very similar to the advantage of dynamic look and move algorithms over static look and move algorithms in visual servoing. Moreover, having online grasping point generation enables the usage of the algorithm for moving objects too.

To the best of our best knowledge, the only work in literature that uses active vision for grasping purposes is presented in [159]. This algorithm is formulated for the antipodal grasps of convex objects. However, it does not address grasping of concave objects in 3D.

Our algorithm is made up of 4 stages: 2D modeling, grasping point generation, grasping point filtering and visual servoing. The control loop can be seen in Figure 4.1 together with these stages.

The first step of the algorithm is to generate a 2D model of the target object. To acquire this model in a fast and robust way, we model the silhouette of the object using Elliptic Fourier Descriptors (EFD) [96]. EFD is a widely used method in object classification and recognition [139, 44]. Recently, some applications are presented regarding the use of EFD in the robotics field [56]. This method is

robust to noisy, erroneous data and occlusions. Thus, it does not necessitate a good segmentation or a clear view of the object. We acquire the model of the object silhouette using low harmonics. In that way we have a rough model of the target object which can be obtained fast, but still has enough information for grasping point generation and pose update of the robot. The EFD modeling stage is presented in Section 4.2.

In the second step of our method, we generate grasping points by interpreting the EFD model. Here, we assume that concavities and flat surfaces of the object are suitable for grasping. Therefore, we analyze the curvature of the EFD model. It is important to note that since we model the object using low harmonics, we still obtain curvature on flat surfaces of the silhouette. Thus, by evaluating curvatures of the model, we do not only evaluate the curvature of the object but also flat surfaces of the object which are also suitable for grasping. Grasping point generation is given in Section 4.3 of the paper.

In the third step, we filter grasping points by using some quality measures. A force closure concept [60] is the first building block of this stage. Also, kinematics of the robot and gripper should be taken into account in this step. The output of this step is grasping points on the target object. The filtering stage is covered by Section 4.4

Visual servoing rule forms the last step. This rule controls the robot in such a way that the curvature measured at grasping points are maximized. Maximizing the curvature means leading the robot to more concave regions of the object. This specifies our approach vector for a better grasping posture. The visual servoing rule is presented in Section 4.5.

The performance of the proposed algorithm is examined by simulations and experiments in Sections 4.6 and 4.7 respectively.

## 4.2 Modeling with EFD

Any closed and bounded curve can be represented by using Elliptic Fourier Descriptors (EFD). We use EFD to model the silhouette of the target object. EFD is an representation composed of sum of sine and cosine functions in different harmonics:

$$x(t) = A_0 + \sum_{n=1}^k (a_n \cos \frac{2n\pi t}{T} + b_n \sin \frac{2n\pi t}{T}) \quad (4.1)$$

$$y(t) = C_0 + \sum_{n=1}^k (c_n \cos \frac{2n\pi t}{T} + d_n \sin \frac{2n\pi t}{T}) \quad (4.2)$$

where  $a_k$ ,  $b_k$ ,  $c_k$  and  $d_k$  are fourier coefficients and  $k$  is the number of harmonics. These coefficients can be obtained using edge data as explained in [96]. The EFD

model provides the coordinates of the object edge in  $x$  and  $y$  direction for a given  $t$ . Also note that these functions are periodic between  $t = [0 \ 2\pi]$ .

The precision of the representation is proportional to the number of harmonic that is used to model the object. Using high harmonics also increases the computational time. We prefer to model the object using lower harmonics since:

1. We are not interested in a detailed model of the object. A rough model of the object silhouette is enough to run the grasping algorithm.
2. By using lower harmonics we obtain curvatures at the model segments that correspond to the flat surfaces of the object. This makes the grasping point generation process easier since we only evaluate curvature of the model rather than considering flat surfaces separately. This phenomenon is presented in Section 4.6 and 4.7.
3. We want to use this algorithm in control loop in an online manner. Therefore, low computational cost is desirable.

In all simulations and experimental results presented in this paper, we obtained EFD model using four harmonics.

### 4.3 Obtaining Grasping Points

By using EFD, we have a parametric representation of the curve. The first derivative of this function gives the tangent vectors and its second derivative gives the normals in  $x$  and  $y$  directions. Therefore for  $x$  direction:

$$T_x(t) = \dot{x}(t) = \sum_{n=1}^k \left( -a_n \frac{2n\pi}{T} \sin \frac{2n\pi t}{T} + b_n \frac{2n\pi}{T} \cos \frac{2n\pi t}{T} \right) \quad (4.3)$$

$$N_x(t) = \dot{T}_x(t) = \sum_{n=1}^k \left( -a_n \left( \frac{2n\pi t}{T} \right)^2 \cos \frac{2n\pi t}{T} - b_n \left( \frac{2n\pi t}{T} \right)^2 \sin \frac{2n\pi t}{T} \right) \quad (4.4)$$

where  $T_x$  and  $N_x$  are  $x$  component of the tangent and normal vectors respectively.

Here, we assume that concave areas and flat surfaces of the object can be good grasping regions, and since we model the object with low harmonics we obtain curvature at both of these segments. From the equations of second derivative of the EFD model, the curvature of the model can be calculated as norm of the normal vector  $N = \begin{bmatrix} N_x \\ N_y \end{bmatrix}$ :

$$C(t) = |N(t)| \quad (4.5)$$

Here Euclidean norm can be used. The local maximum of the curvature data gives us the points where curvature is maximum. However, these points correspond to both most concave and most convex regions of the model. To differentiate

between convex and concave regions, the cross product of two consecutive tangent vectors that are selected in the region can be used. The sign of the  $z$  component of the cross product gives us whether the region is convex or concave. If the edge data is read in a clock-wise direction, then positive values correspond to concave regions and vice versa.

Points with maximum curvature correspond to where the first derivative of the curvature is zero and the second derivative is negative. Assuming the data is read in a clockwise direction, a set of grasping points  $\mathbf{G}$  can be obtained as:

$$\mathbf{G} = \{C(t) | (C'(t) = 0) \wedge (C''(t) < 0) \wedge ([0 \ 0 \ 1] (N(t) \times N(t+1)) > 0)\}$$

It is important to note that these grasping does not correspond to 3D points on the objects. However, they are being used as guidelines for the robot fingers. The usage of grasping points can be seen in the simulation results section. The next stage is filtering these points.

## 4.4 Filtering Grasping Points

The grasping point candidates should be filtered to come up with actual grasping points. The number of grasping points in this final set should be equal to the number of the contact points of the gripper. Therefore, grasping point candidates are grouped to form grasping point groups. Then, these groups are ordered by the sum of curvature value of the member grasping points. The group that has the biggest curvature value corresponds to grasping the object from the most concave parts.

Beginning from the group with highest curvature value, a force closure test is applied to see if the grasp is stable or not. According to this concept, a stable grasp should balance all external forces and moments on the object. If the test is successful, then these points are chosen as final grasping points. If not, a force closure test is applied to the next group.

The method above is valid if the gripper has enough degrees of freedom to reach any grasping points of the object. However, if this is not the case then the kinematics of the gripper should be taken into account while forming the grasping point groups.

There can be situations that none of the grasping point sets satisfy the conditions above. Then, convex regions should also be taken into consideration. Among the convex regions the least convex parts (the ones with low curvature value and the sign is negative) are suitable to be selected as grasping points. The pseudo code of the grasping filter is given below.

After the grasping points are selected, they can be easily tracked since the positions of the points on the model between frames do not change rapidly. Thus, it is sufficient to run the filtering only once in the beginning of the algorithm.

---

**Algorithm 4.1** Grasping point filtering

---

**Require:** grasping point candidates

- 1: Form sets of all possible concave grasping points combinations (Size of the set = number of contact points of the gripper).
- 2: Order the sets with respect to sum of curvature values
- 3: **repeat**
- 4:   For all the sets starting from the set with highest curvature sum.
- 5:   Apply force closure test.
- 6:   Apply kinematics test.
- 7: **until** all test are successful for a set or all sets fail
- 8: **if** all the sets fail **then**
- 9:   **repeat**
- 10:     Form new sets with convex grasping points starting from the least concave one.
- 11:     Apply force closure test.
- 12:     Apply kinematics test.
- 13:     **until** all test are successful for a set or all sets fail
- 14:     **if** all sets fail **then**
- 15:       return failure.
- 16:     **else**
- 17:       Select the set.
- 18:     **end if**
- 19: **else**
- 20:     Select the set.
- 21: **end if**

---



Figure 4.2: *World frame and task frame shown on a Puma robot*

## 4.5 Visual Servoing Rule

In this section, a visual servoing rule is designed to lead a 6 DOF robot arm to the position where the curvature of the model is maximized. This is a local maximum position and we trust the fact that there is not one but many good grasps of an object.

A control rule is employed in task space. A task frame is attached at the tool tip of the robot. (World frame and task frame are shown in Figure 4.2 on a Puma robot.) Three translations and three rotations of task space are grouped for some subtasks. These subtasks are

1. bringing the object at the center of the image,
2. orienting the gripper for a suitable grasp,
3. maximizing the curvature,
4. approaching to the object.

First three subtasks are executed simultaneously, whereas approaching the object is the done when the curvature maximization is finished as explained below.

Rotations in  $x$  and  $y$  directions are controlled (like a pan tilt system) to bring the target object at the center of the image. Here, the center of the first harmonic of the model, the points  $A_0$  and  $C_0$ , are used as feature points. The columns of image jacobian that corresponds to rotations in  $x$  and  $y$  direction are used to obtain the velocity skew as follows:

$$\begin{bmatrix} V_{rx} \\ V_{ry} \end{bmatrix} = -\lambda_{rxy} \begin{bmatrix} xy & -(1+x^2) \\ 1+y^2 & -xy \end{bmatrix}^{-1} \begin{bmatrix} e_{rx} \\ e_{ry} \end{bmatrix} \quad (4.6)$$

Here,  $V_{rx}$  and  $V_{ry}$  are the velocity skew for rotation in  $x$  and  $y$  directions,  $\lambda_{rxy}$  is the control gain and

$$e_{rx} = o_x - a_0 \quad (4.7)$$

$$e_{ry} = o_y - c_0 \quad (4.8)$$

where  $o_x$  and  $o_y$  are the image center pixels.

The rotation in the  $z$  direction is controlled to give orientation to the gripper. This rule should be defined depending on the type of the gripper. For example, if a multi-fingered hand with multiple degrees of freedom (like the Barrett hand [72]) is mounted on the robot arm, then orientation of the gripper is not very important, since the hand has enough degrees of freedom to lead the fingers to the grasping points. However, if a two jaw parallel gripper is used, then we have two grasping points and the gripper should be oriented towards these points. If we take a two jaw parallel gripper into account, then the slope of the line that pass through the two grasping points,  $\phi$ , can be used as follows:

$$V_{rz} = -\lambda_{rz}(\phi_r - \phi) \quad (4.9)$$

where  $\phi_r$  is the reference slope defined by the orientation of the gripper.

Translations in  $x$  and  $y$  directions are assigned to maximize the curvature while approaching the target object. However, since we only have a 2D model of the object and some curvature values extracted from this model, which direction causes an increase in curvature is unknown. This prevents us to control translations in  $x$  and  $y$  directions simultaneously. Even while controlling  $x$  and  $y$  directions separately, a direction check is needed; we need to choose between  $-x$  and  $+x$ ,  $-y$  and  $+y$ .

To detect this direction, the robot goes to a random direction for a very brief moment and checks the curvature measure. According to that observation, the direction of the control is determined. Using this direction, the control law can be written as follows:

$$V_x = \lambda_x D_x \quad (4.10)$$

$$V_y = \lambda_y D_y \quad (4.11)$$

Here,  $\lambda_x$  and  $\lambda_y$  are control gains.  $D_x$  and  $D_y$  take 1 or  $-1$  depending on the direction test. These references are applied until a curvature drop is measured.

When the curvature is maximized, the approaching phase begins. Since we are using the task frame, the  $z$  direction is along the optical axis of the camera. The object is always in the center of the image by rotations in  $x$  and  $y$  direction. Thus, moving along  $z$  direction corresponds to approaching the object. This can be done by introducing a constant feedforward velocity reference ( $V_{ff}$ ):

$$V_z = V_{ff} \quad (4.12)$$

The value of  $V_{ff}$  directly defines how fast the robot approaches to the object. Here, the stop condition is important. The robot should stop as the gripper is close enough to the object. This can be detected by a distance sensor, or a depth estimation algorithm can be used. As appropriate distance is achieved, the gripper should execute the grasping task by using the grasping points to position the fingers around the object.

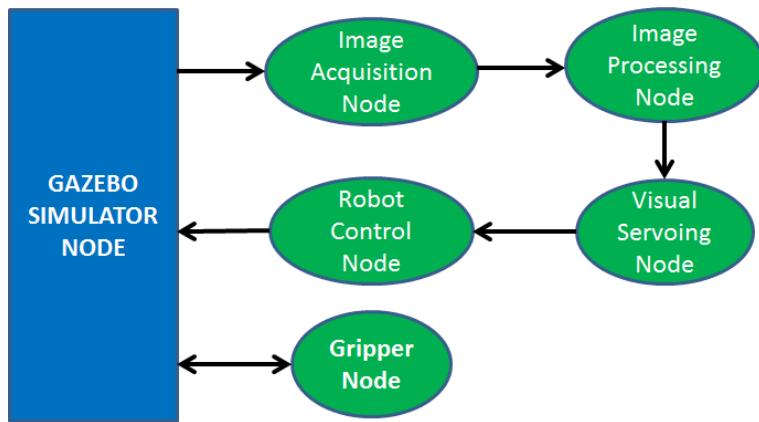
## 4.6 Simulation Results:

Simulations are conducted to determine the performance of the grasping algorithm. In these simulations, the Robot Operating System (ROS) is used together with an Open Dynamics Engine (ODE) based simulator Gazebo.

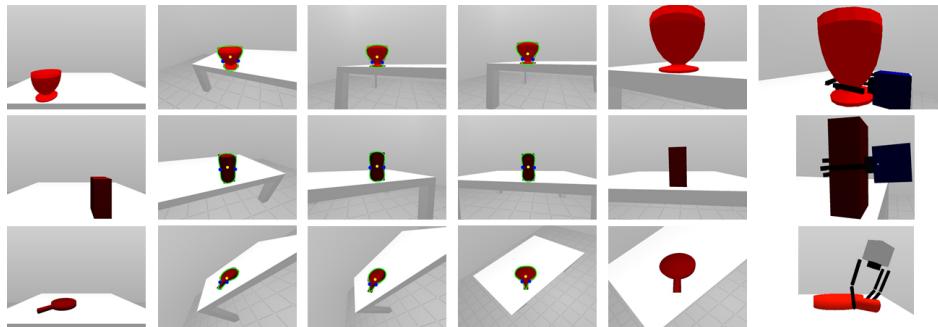
A virtual camera is placed in the end effector of a six DOF robot in Gazebo. A three finger gripper is attached to the end effector together with a distance sensor placed at the palm. Six ROS nodes are used for the simulation: Gazebo simulator, image acquisition, image processing, visual servoing, gripper control and robot control. The image acquisition node is used to acquire frames from the virtual camera. The image processing node takes images as input and run the EFD algorithm. This node sends  $A_o$ ,  $C_o$ , pixel values of the grasping points and curvature information to the visual servoing node. The visual servoing node processes this data as explained in Section 4.5 and sends velocity reference to the robot control node. The robot control node generates control inputs to the robot joints and sends them to Gazebo simulator. Finally, the gripper node reads the output of the distance sensor and closes the gripper when a suitable distance is achieved. This structure is visualized in Figure 4.3.

The algorithm is tested on a cup, a rectangular box and a table tennis racket. These objects are chosen to demonstrate the performance of the algorithm on different types of objects. The cup is used to represent 3D objects that have concave parts. The rectangular prism is an example of objects with flat surfaces. Finally, the table tennis racket is used to demonstrate the performance on almost planar objects. Snapshots of these three objects are given for different phases of the algorithm in Figure 4.4.

In this paper, the cup example is examined in detail. For this object, the curvature value and velocity reference in  $x$  and  $y$  directions are given in Figure 4.5. Here, the trial motion begins by giving a velocity reference in negative  $y$  direction. This corresponds to a motion upwards and a curvature drop is measured. Since the aim of the algorithm is to maximize the curvature value of the selected grasping point, the direction is reversed. Thus, the motion in positive  $y$  direction begins and it lasts until a curvature drop is measured. Then the trial motion in  $x$  direction begins. Since this object has an axial symmetry, motion in that direction does not cause a significant curvature change. Therefore, pose correction motion stops



**Figure 4.3:** ROS structure



**Figure 4.4:** Snapshots from the different phases of the algorithm. Green line: EFD model, Blue points: Grasping points, Yellow point: Center of the EFD model ( $(A_o, C_o)$  point). From left to right: Camera view in initial position, after the algorithm is enabled and object is centered by using the center of the model, after the pose correction is applied on the y direction, after correction is applied on the x direction, while the robot approaches in the z direction of the task space, snapshot of the gripper after the grasp is executed.

after a brief motion. Alternatively, the algorithm can be stopped if the curvature change is under a certain threshold in the trial phase.

As it is explained in Section 4.2, we were modelling the silhouette of the object using low harmonics. One reason was that, by doing so, we were able to obtain curvature values in long flat surfaces of the object that are suitable for grasping. The curvature value increases as the flat surfaces elongates. As it can be seen in the results with the rectangular prism in Figure 4.4, the algorithm is able to update the pose of the robot in  $x$  and  $y$  direction of the task space using this property.

With an almost flat object, the table tennis racket, the result is also successful. The algorithm successfully leads the robot to grasp the object from the top. However, the distance threshold to trigger the grasp is increased manually to enable a fingertip grasp.

The performance of the algorithm is analyzed further with experimental results in the next section.

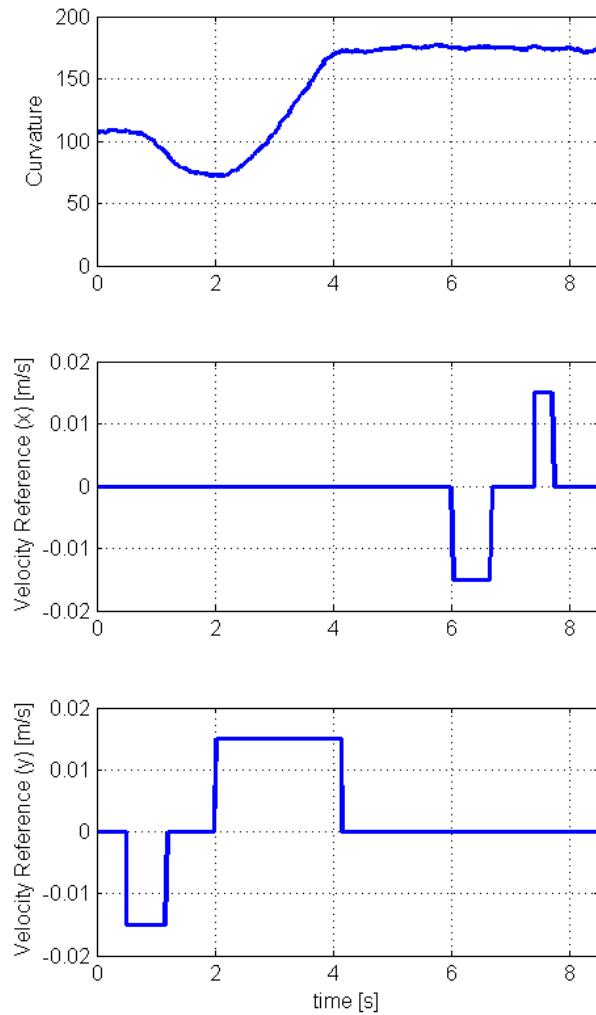
## 4.7 Experimental Results

Experimental results of the algorithm are presented in this section. We used the Delft Personal Robot Prototype 1 (DPR1) as our test bed. The DPR1 is a nonholonomic mobile robot with an underactuated gripper attached to it. Similar to simulations, a distance sensor is attached on the gripper. The robot has an embedded velocity controller that is running at 100 Hz and the EFD algorithm supplies velocity references at 10 Hz.

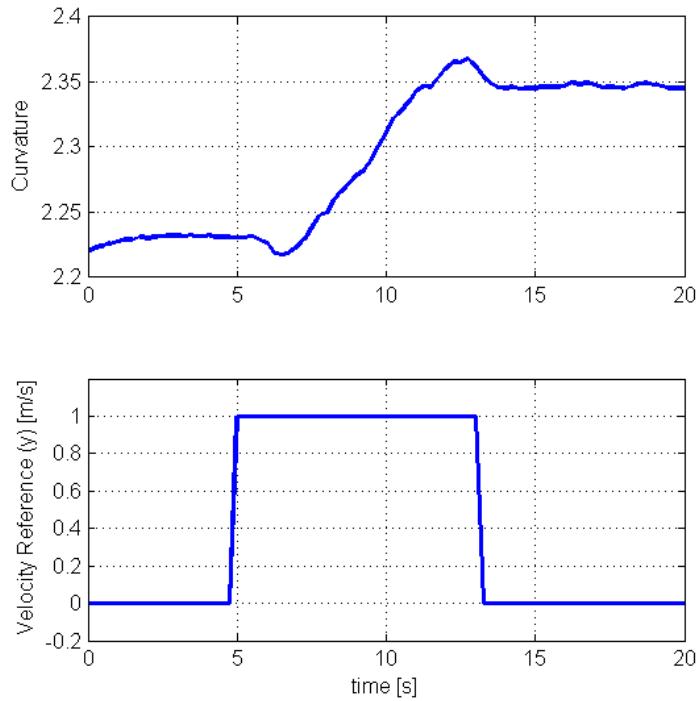
Due to the nonholonomic nature of the robot, we were able to test the algorithm in  $y$  direction of the task space. Our test objects are a cup and a rectangular box. The velocity reference in  $y$  direction and the curvature value for the cup are given in Figure 4.6. Here, we preferred not to interrupt the velocity reference if the trial direction does not need to be reversed. Since, this was the case in the experiment, the robot continues to move to positive  $y$  direction until it measures a curvature drop. Frames captured in the initial position and after the curvature is maximized are presented in Figure 4.7. Considering the offset of the camera and the position of the arm in maximum curvature, the position of the arm is updated. Then, the robot approaches to the object and grasps it. Figure 4.8 presents some snapshots of the executed grasping task. For the rectangular box the initial and final images are given in Figure 4.9.

## 4.8 Conclusion

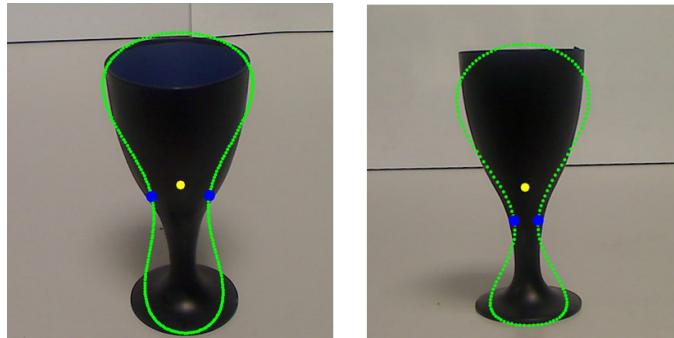
In this chapter, a novel grasping algorithm is presented. The target object is modeled by Elliptic Fourier Descriptors and a visual servoing rule is designed to achieve a good grasping configuration using curvature measure. The algorithm is



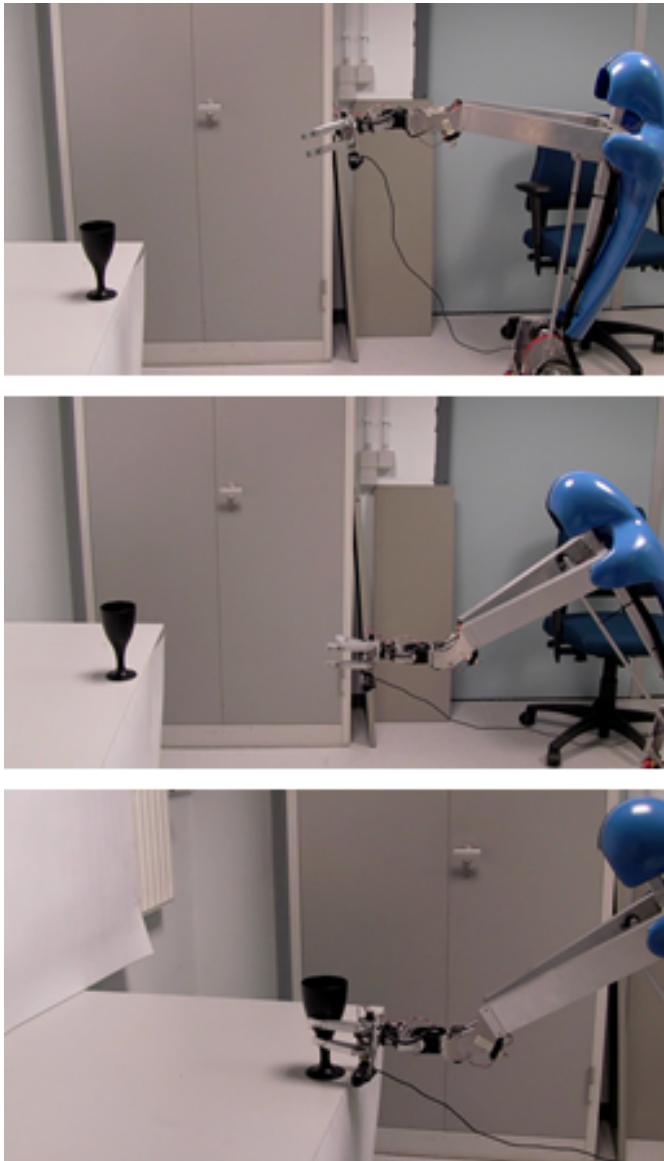
**Figure 4.5:** Curvature value of the selected grasping point and velocity references in  $x$  and  $y$  direction for the cup.



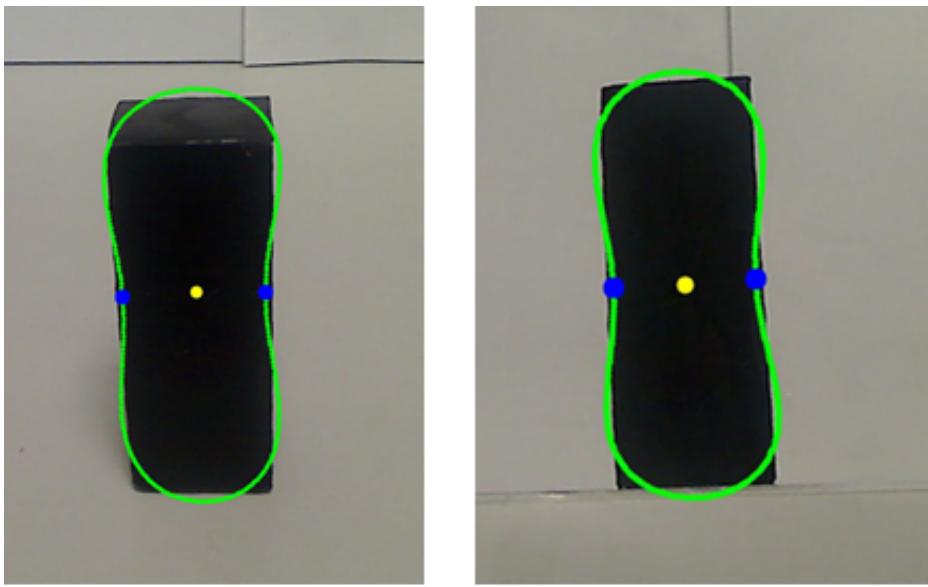
**Figure 4.6:** Plots of the experiment.



**Figure 4.7:** Snapshots taken from the eye in hand camera before and after the execution of the pose correction algorithm for the cup.



**Figure 4.8:** DPR1 while executing the grasping task. Initial view, after pose correction, after the object is grasped.



**Figure 4.9:** Snapshots taken from the eye in hand camera before and after the execution of the pose correction algorithm for the rectangular box.

fast enough to be used in the control rule online. Promising results are obtained in simulations and experiments.

To our opinion, the algorithm can be extended by adopting an online learning mechanism that makes grasping better at every execution. Also different properties of the EFD model can be examined for grasping point generation.

### Final Notes:

The algorithm that is presented in this chapter is a simple application of the proposed viewpoint optimization framework. In the following chapter, the optimizers in literature that are suitable as a solution to the viewpoint optimization problem are investigated in detail.

# Comparison of Extremum Seeking Control Algorithms for Robotic Applications

In order to conduct efficient viewpoint optimization, a powerful optimizer that is suitable for our problem should be chosen. The viewpoint optimization problem for grasp synthesis algorithms for unknown objects has two distinctive characteristics: 1) The objective function of the optimization is unknown, 2) The gradient of the objective function is unknown. These imply that the direction to follow in order to increase the objective value (grasp quality) is not directly calculable. Extremum seeking control algorithms present efficient solutions to such optimization problems. In this chapter, the extremum seeking control methods have been investigated with a comparative study, and the most suitable method to our problem is aimed to be chosen.

This work was published in the proceedings of the IEEE/RJS International Conference on Intelligent Robots and Systems (IROS) that was held in 2012 [22].

## Abstract

The purpose of this chapter is to help engineers and researchers to select extremum seeking control (ESC) techniques for robotic applications such as object grasping, active object recognition and viewpoint optimization. These techniques are categorized into five main groups: Sliding mode ESC, neural network ESC, approximation based ESC, perturbation based ESC and adaptive ESC. These groups are explained briefly by stressing their working principles and the effect of the parameters. Then, the techniques are compared with respect to their ro-

bustness to noise and system dynamics by simulations. In conclusion, we propose the usage of the approximation based methods when the noise level is negligible. When noise is present, the neural network based optimizers are a better choice thanks to their hysteresis functions. However, if the system has both high noise and dynamic effects, then the perturbation based method is preferable since large motions provide robustness to noise and smooth references generated by the algorithm are less likely to cause instability. An application example was also given on texture density maximization.

## 5.1 Introduction

Extremum seeking control (ESC) is a well developed field which addresses the problem of objective value optimization when the objective function, its gradient and optimum value are unknown. Some widely known applications of ESC are ignition time selection for combustion engines [84], bioreactor optimization [169] and anti-lock braking system control [48]. A large list of applications can be found in [179].

Besides these applications, ESC algorithms have great potential in robotics. Especially for robots that operate in unknown/dynamic environments (i.e. homes, offices, elderly care centers), the optimum values of the objective variables of a task are unknown because of the unknown state/effect of the environment. In these cases, online optimization tools like ESC can be utilized in order to optimize these task variables.

A very recent example of ESC in the robotics field is proposed in [180]. In this work, an eye in hand system is used and the saliency (a measure of interest) of the view of the camera is maximized by providing references to the robot arm. Here, since the saliency map of the environment is unknown, the objective function cannot be used for the optimization purposes. However, the objective value can be calculated for each view. The ESC framework enables online optimization for such a task. Another example is presented in [29] and [177] where ESC is implemented for non-holonomic systems in order to seek the source of a signal.

Our interest in ESC algorithms stems from our vision based grasping research [23]. One challenge in this field is to develop algorithms that can calculate the positions of stable grasping points on an object. This can be achieved by constructing an objective function based on the object model and running an optimization algorithm which searches for the optimum grasping points. However, we are specifically working on grasping unknown objects, which automatically implies that the model of the object is unknown. On the other hand, using an eye in hand system, an objective value can be calculated for each image of the object. In this case, an ESC algorithm can be utilized in order to run an online optimization for grasping purposes, so that the object modeling process can be skipped and a faster grasping can be achieved. With a very similar formulation, these algorithms can also be used to increase the performance of the object re-

cognition systems by altering the viewpoint of the object in order to see surfaces with high texture density.

In this paper, we analyze the ESC algorithms for robotics applications. The robotic systems can be characterized by their nonlinear dynamics and sensor noise. The algorithm should be able to perform well under these circumstances. The performance of the algorithms are analyzed by their transient response using the rise time and settling time measures. The smoothness of the velocity references generated by the algorithms is also another criterion. Moreover, since most robots can operate in multiple dimensions, the performance of the algorithm in the multivariate case is also crucial. For this case, the trajectories generated by the algorithms are presented in order to analyze the motion in the search space, and a numerical comparison is given by presenting the total distance traveled until 95% of the optimum value is reached. In most robotic applications, the objective value can be measured continuously. For example, in the implementation of [180], the saliency value can be computed for each image acquired from the camera. Therefore, in this paper we specifically concentrate on *analog* ESC algorithms, which utilize the objective value continuously, and exclude algorithms like the simplex method [111].

Two other comparison papers on ESC can be found in [118] and [116]. These papers present a detailed comparisons of two ESC methods whereas we focus on robotic applications and present a larger set. Also, a good history study of the ESC can be found in [157].

The analog extremum seeking control algorithms can be categorized into five groups: Sliding mode ESC, neural network ESC, approximation based ESC, perturbation based ESC, adaptive ESC. The next section summarizes these ESC methods. The third section gives comparative simulation results. The fourth section presents an analysis based on the simulation results. In the fifth section, experimental results on texture density maximization is presented. Finally, in section six the paper is concluded with a summary.

## 5.2 Analog Extremum Seeking Control Methods

The problem of ESC is formalized as follows: For a given system

$$y = f(x) \quad (5.1)$$

where  $f(x)$  is the unknown objective function,  $x \in R^n$  is the state vector with dimension  $n$ , and  $y$  is the objective value that can be measured continuously, an analog extremum seeking control algorithm searches the state space for the state vector  $x$  that corresponds to the optimum (most of the time local optimum) of the objective value. To achieve this goal, the following algorithms are presented in the literature (all the derivations below are for the minimization of the objective value):

### 5.2.1 Sliding mode ESC

In early 70's, Korovin and Utkin proposed sliding mode ESC [89, 90]. This method is based on a driving signal. Unlike conventional control problems, the reference value is unknown in the ESC framework since the optimum value of the objective function is unknown. The driving signal in sliding mode ESC functions as a reference generator for the system, and it is designed to be monotonically decreasing when searching for the local minimum. The ESC rule is then formulated such that the system tracks this driving signal.

As in the most sliding mode systems, the tracking of the signal relies on the high frequency chattering around the signal's value. This behavior makes this ESC algorithm unsuitable for robotic applications. Preserving the main idea, Yu and Ozguner [176] altered this method in order to prevent the high frequency chattering. This type of sliding mode ESC is analyzed in this paper.

Define a monotonically decreasing driving signal  $g(t)$  whose derivative is given as

$$\dot{g}(t) = -\rho \quad (5.2)$$

where  $\rho$  is a positive constant. The error between objective value and the driving signal is

$$e = y - g(t) \quad (5.3)$$

The output of the ESC algorithm is designed as

$$u = \text{sign}(\sin(\pi e/\alpha)) \quad (5.4)$$

where  $\alpha$  is a positive constant. This output is fed to the system as velocity reference with a gain  $k$  which determines the convergence rate:

$$\dot{x}_r = ku \quad (5.5)$$

The reason of using a sine inside the signum function is the following: since the objective function is unknown, the ESC algorithm does not know the direction in which it should lead the system in order to minimize the objective value. If the system goes in a wrong direction, the error will increase. This rise will continue until the value of the sine function changes sign. Then, the velocity reference will also change direction, and the system will start following the driving input. The bound on the error that is allowed by the system can be tuned by the parameter  $\alpha$ . If the system is slower or faster than the driving signal, this will cause chattering and the frequency of this chattering can be remarkably decreased by increasing the  $\alpha$  parameter. When the optimum value is reached, the system oscillates within the allowed error bound. This method is implemented for an anti-lock braking system in [176].

**Multivariate Extension** Although it is quite straightforward to implement sliding mode ESC in the unidimensional case, the multidimensional implementation is

problematic. For the algorithm in [89], there are some propositions on multivariate extensions, but to the best of our knowledge there is no successful implementation of sliding mode ESC in the multivariate case. Thus, in Section 5.3 we will only analyze the performance of the sliding mode algorithm for a single dimension.

### 5.2.2 Neural network ESC

Neural network ESC [160, 161] is based on a minimum peak detector and two switching functions with hysteresis. Just like the driving signal concept in sliding mode ESC, neural network ESC also has a reference generator. One component of this reference is the minimum peak detector, and it is designed to be monotonically decreasing as follows:

$$\dot{y}_p = \begin{cases} 0 & (y_p \leq y) \\ -M & (y_p > y) \end{cases} \quad (5.6)$$

where  $y_p$  is the value of the minimum peak detector, and  $M$  is a parameter that defines the speed of convergence of  $y_p$  to  $y$ , when  $y$  is smaller. This speed should be higher than the rate of change of the objective value in order to maintain convergence:

$$|\dot{y}| < M \quad (5.7)$$

The second component of the reference is provided by the switching function  $W$ . This switch starts functioning after the optimum value is reached, and it will be explained shortly. The reference is formed by the addition of the two components:

$$y_r = y_p + y_w \quad (5.8)$$

The error function is defined as the difference between the objective value and the reference value:

$$e = y_r - y \quad (5.9)$$

The control law  $u$  is another switching function with hysteresis and is designed as:

$$u = \begin{cases} -A & e < -\delta \\ A & e > \delta \\ [previous\_state] & \text{otherwise} \end{cases} \quad (5.10)$$

Here,  $A$  is a constant that specifies the magnitude of the velocity reference, and  $\delta$  is the hysteresis width. If the direction of the system makes the objective value converge to the optimum value, then this direction is kept. Otherwise, the error will increase, and when it is greater than the hysteresis, the direction will be reversed. However, this function can not stop the system when the optimum value is reached. The role of the  $y_w$  signal begins here and it alters the reference value. The derivative of the  $y_w$  signal is defined as

$$\dot{y}_w = \begin{cases} 0 & e > \Delta \\ B & e < -\Delta \\ [previous\_state] & \text{otherwise} \end{cases} \quad (5.11)$$

where  $B$  is a positive constant. The value of the second hysteresis width  $\Delta$  should be greater than  $\delta$ , since  $y_w$  should start functioning after the optimum value is reached:

$$\Delta > \delta \quad (5.12)$$

The control signal  $u$  is fed to the systems as velocity input:

$$\dot{x}_r = u \quad (5.13)$$

When the optimum value is reached, the system oscillates with an error under  $\Delta$ .

**Multivariate Extension** In order to extend the neural network based algorithm to the multivariate case, the usage of a coordinate descent algorithm is proposed [161]. The main logic of this algorithm is to descritize the search directions and to minimize in one direction at a time. To achieve this, the control law  $u$  is implemented with  $2n - 1$  switching functions with different hysteresis values where  $n$  is the dimensionality of the state. The direction of the search is changed whenever a hysteresis threshold is reached in that direction and this brings the following constraint to the hysteresis widths:

$$\delta_1 < \delta_2 < \dots < \delta_{2n-1} < \Delta \quad (5.14)$$

### 5.2.3 Approximation based ESC

When the objective function to be optimized is unknown, one natural choice is to derive a local representation of this function based on the past data. This representation can then be used with either a gradient or a non-gradient based approach. For feedback linearizable systems, a trust region based algorithm is designed in [178]. The main idea of the trust region algorithm is the following: First a well-poised approximation set is formed, and a local approximation of the objective function  $\hat{f}(x)$  is obtained. Since this is only a local approximation, it can only be trusted in a local region. An initial value  $\Delta_0$  is set for this trust region, and the approximated objective function is optimized within this region. Let's say the system is at  $x_k$ , and the optimum value in this local region is found at  $x_k + p_k$ . The system is led to this new set point and whenever the set point is reached the following ratio is calculated in order to measure how good the approximation of the objective function was:

$$\rho_k = \frac{f(x_k) - f(x_k + p)}{\hat{f}(x_k) - \hat{f}(x_k + p)} \quad (5.15)$$

If  $\rho_k$  is close to 1, this signifies a good approximation, and trust region radius can be increased, otherwise it should be decreased. This update rule is as follows:

$$\Delta_{k+1} = \begin{cases} [\Delta_k, \infty], & \rho_k \geq \eta_2 \\ [\gamma_2 \Delta_k, \Delta_k] & \rho_k \in [\eta_1, \eta_2] \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k], & \rho_k < \eta_1 \end{cases} \quad (5.16)$$

where  $0 < \eta_1 \leq \eta_2 < 1$  and  $0 < \gamma_1 \leq \gamma_2 \leq 1$ . For more detailed information about the trust region methods, see [3].

Considering [178], the advantage of using feedback linearizable systems is to be able to ensure the convergence of the system to the set points and be able to estimate the regulation time, so that the time when  $x_k + p_k$  is reached is known approximately. Since we don't want to impose any constraints on the system in this paper, we used the error between  $x_k + p_k$  and  $x$  in order to detect when the reference is reached.

Alternatively, since the gradient of the objective function can be estimated locally by the derivative of the approximated objective function, a line search algorithm such as gradient descent can also be utilized as in [179]. In this case, the velocity reference can be generated as follows:

$$\dot{x}_r = -k \frac{d\hat{f}(x(k))}{dx} \quad (5.17)$$

where  $k$  is a positive gain. The simulation results are presented for both methods in Section 5.3.

**Multivariate Extension** For the multivariate case, multivariate approximation techniques can be used to approximate the objective function locally. We used bilinear approximation in this paper for two dimensions. The rest of the derivation is the same as the unidimensional case.

#### 5.2.4 Perturbation based ESC

It would be safe to say that perturbation based ESC framework is the most popular method in the literature. These types of methods [95], [6], [5], [117], [134] use an external perturbation signal and modulation theory to find the optimum value. The most commonly used perturbation signal is the sine wave. The ESC algorithm sends the sine wave to the system as position reference together with an adaptation input:

$$x = a \cdot \sin(\omega t) + \hat{x} \quad (5.18)$$

with some amplitude  $a$  and modulation frequency  $\omega$ . The adaptation signal  $\hat{x}$  shifts the sine wave towards the gradient direction. A way of calculating this signal is the following procedure: Combining the adaptation signal with the sine

signal as in (5.18) is the modulation phase of the algorithm. The response of the system to this signal is measured in the objective value  $y$ . This output is filtered by a high pass filter to eliminate the DC component and demodulated by the same sine signal to extract the gradient direction.

$$\xi = y \left( \frac{s}{s + h} \right) (a \cdot \sin(\omega t)) \quad (5.19)$$

This information is utilized in order to calculate the shift in the sine signal towards the gradient. This part of the algorithm is called the adaptation law.

$$\dot{x} = -\xi \frac{k}{s} \quad (5.20)$$

Here,  $k$  is a positive constant that specifies the adaptation speed. The algorithm has several variations. Some of them include applying a low pass filter to  $\xi$  before using it in the update rule since only the DC component of the demodulated signal is needed for gradient calculation. In some other variations, the adaptation law is combined with a compensator which relaxes the design constraints on the parameter  $k$  [94].

The amplitude and the frequency of the sine wave signal and the cutoff frequencies of the filters are important design parameters. A detailed study of how the design of the perturbation signal affects the performance of the ESC system is presented in [112]. Further analysis with variable gains and with the presence of noise in the measurement channel is presented in [152]. The applications of perturbation based ESC on a bioprocess, fluid flow control, magnetically suspended flywheel and online parameter tuning for combustion timing can be found in [46], [85], [167] and [84] respectively. Also, an application for non-holonomic systems is given in [28] where the algorithm is enhanced for being able to alter the trajectory of the robot.

**Multidimensional Extension** Extending this method to the multidimensional case is done by assigning a sine wave to every input channel with some phase shifts. For example, in two dimensional case, if the phase shift between the sine waves of the first channel and the second channel is  $\pi/2$ , then the system moves with circles in the task space, and the center of the circle is shifted by the adaptation input which is calculated by the same modulation and demodulation steps.

### 5.2.5 Adaptive ESC

Adaptive ESC [153], [172] uses adaptive control schemes for the online optimization. Unlike the approximation based methods, the adaptive ESC approximates the objective function globally. This algorithm necessitates the type of the objective function to be known. The adaptive ESC implementations for Hammerstein and Wiener type nonlinear objective functions are given in [172] and [171]. In this paper, we do not want to impose any constraints on the objective function,

so we will not analyze this method. However, we find it important to note that, if the form of the objective function is known, then this method provides efficient results by quickly identifying the extremum, and driving the system towards it.

## 5.3 Simulations

The performance of the above mentioned ESC algorithms are analyzed by simulations in both unidimensional and multidimensional case. All Matlab files of these simulations can be found at <sup>1</sup>.

### 5.3.1 Unidimensional Simulations:

In this subsection, the robustness and general characteristics of the ESC algorithms are examined for the unidimensional case. The simulations are conducted for the following cases:

1. System without dynamics and noise
2. System with noise without dynamics
3. System with dynamics without noise

For the third case, the algorithm is analyzed by using a model with the dynamics of a six DOF robot manipulator provided by a Matlab robotics toolbox [30]. The optimization problem is defined in the task space and the velocity references that are supplied by the ESC algorithms are sent to an inner velocity control loop.

The objective function is specified as a Gaussian based function where  $\mu$  and  $\sigma$  signify the mean and the variance respectively:

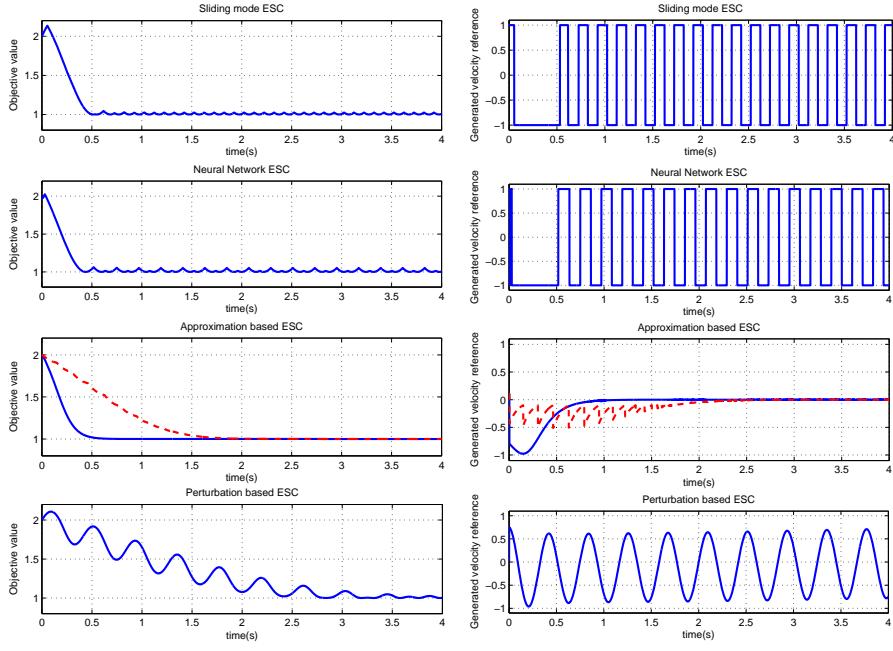
$$f(x) = 2 - e^{\frac{-1}{2} \frac{(x-\mu)^2}{\sigma^2}} \quad (5.21)$$

Here, the parameters  $\mu$  and  $\sigma$  are set to -0.4 and 0.27. The system starts at  $x = 0$ , and the objective value measured at that state is  $y = 2$ . The ESC algorithms search for the minimum value of this function which is  $y = 1$ .

The maximum velocity of the systems are limited to  $1m/s$ . The algorithms are tuned for each separate case considering fast convergence, low chattering, and the steady state error is aimed to be kept below %5. For all the algorithms, the objective value is plotted with respect to time. These plots are evaluated using rise time and settling time measures. Also, the references generated by these algorithms are analyzed considering smoothness.

---

<sup>1</sup><http://www dbl tudelft nl/over-de-faculteit/afdelingen/biomechanical-engineering/onderzoek/dbl-delft-biorobotics-lab/software/>



**Figure 5.1:** Simulation results of the analog ESC methods without noise and dynamics in unidimensional case.

**Simulations without noise and dynamics** Results for the simulations without noise and dynamics are presented in Figure 5.1. It can be seen that the *sliding mode ESC* and the *neural network ESC* performs very similarly. They both lead the system to the wrong direction for a brief amount of time. Then, due to the increasing error between the driving signal and the measurement, both algorithms change the sign of the velocity reference by the mechanisms explained in Section 5.2, and the system starts to approach to the objective value with a constant velocity. The rise time and the settling time of the sliding mode ESC are  $\approx 0.31$  and  $\approx 0.44$  seconds. These values are  $\approx 0.31$  and  $\approx 0.37$  seconds for the neural network controller. When the optimum value is reached, the systems oscillate around the optimum value.

Among the *approximation based ESC* methods, the results of the trust region and gradient based methods are given with the dashed red and blue lines in Figure 5.1 respectively. For the trust region algorithm, each peak at the generated velocity corresponds to a new set point. It can be seen that the thrust region width first increases at the almost linear region of the Gaussian function since the local approximations hold for larger regions. However, near the optimum value, the shape of the Gaussian function changes more rapidly which causes a decrease of the width. The objective value is reached with a rise time of  $\approx 1.39$  seconds, a

settling time of  $\approx 1.46$  seconds.

On the other hand, the gradient based method generates a very smooth velocity reference. This feature is clearly better than the previous algorithms. The rise time is measured as  $\approx 0.4$  seconds, and the settling time is  $\approx 0.42$  seconds. It is clear that the performance of the gradient based method is better than the trust region algorithm considering both the speed of convergence and reference smoothness. Thus, for the rest of this section, only the results with gradient based method will be presented.

The *perturbation based ESC* algorithm provides sinusoidal velocity references to the system. The system converges to the optimum value while oscillating at the same time with smooth references. When the optimum value is reached, the amplitude of the oscillations decreases gradually. This algorithm necessitates a larger motion and this makes rise time and settling time increase drastically. These values are measured as 2.15 and 2.35 seconds respectively.

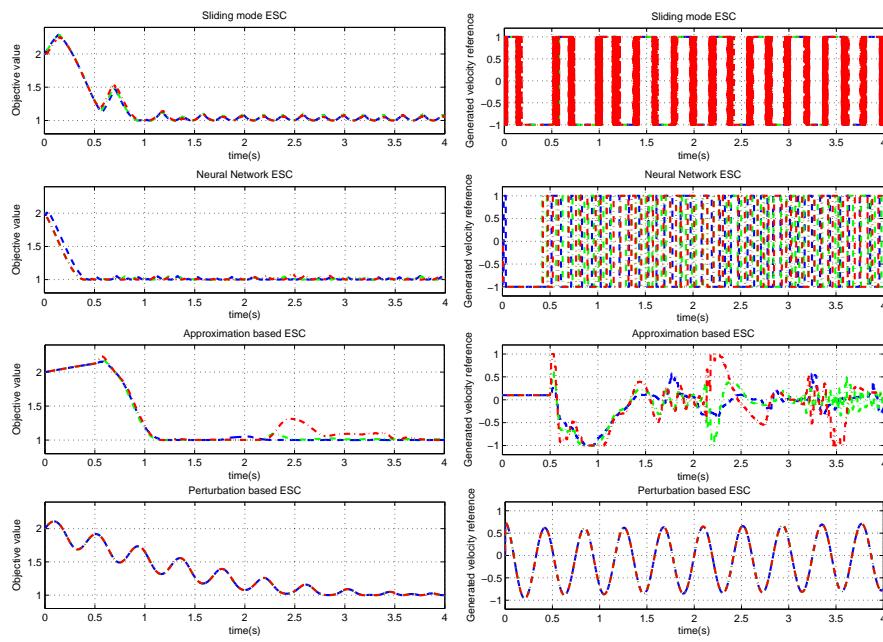
**Simulations with noise without dynamics** The simulations are run for three different measurement noise amplitudes: 0.05, 0.1 and 0.2. The noise is uniformly distributed. These results are presented in Figure 5.2. For the *sliding mode ESC*, an oscillation is observed in the transient response. The rise time and settling time are increased to  $\approx 0.58$  and  $\approx 0.88$  respectively. However, more importantly, high frequency oscillations are observed at the direction change points. The amount of oscillations is increased as the noise amplitude gets higher. These oscillations are prevented by the hysteresis mechanisms of the *neural network ESC*. For this method only small irregularities are observed at the steady state oscillations. Other than that, its performance is very close to the case without noise.

For the *approximation based ESC*, more data points are used for interpolation than in the noise-free case in order to cope with noise. The motion until  $t = 0.5$  is for collecting these necessary data. This motion caused an increase in the settling time which is measured as  $\approx 1.08$ . Additionally, although the objective value converges smoothly to the optimum value, the smoothness of the velocity reference is effected negatively. This causes the system to diverge from the optimum value, and it gets worse when the noise is increased.

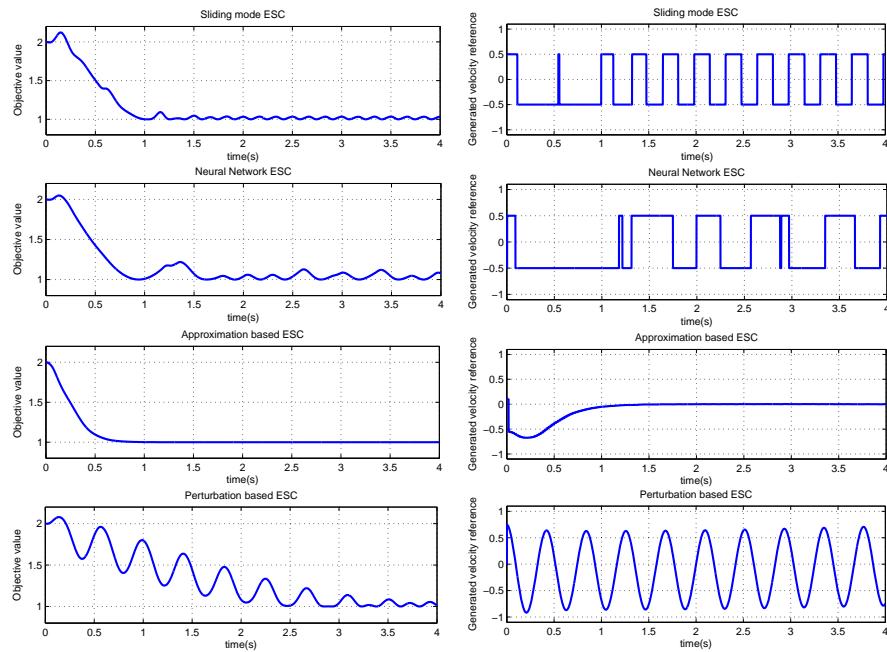
The *perturbation based ESC* stays unaffected for all three noise levels. It provides the same performance as the noise-free case.

**Simulations with dynamics without noise** The results with dynamics are given in Figure 5.3. For the *sliding mode ESC*, the convergence to the optimum value is maintained with an oscillation in the transient. The rise time and settling time values are measured as  $\approx 0.65$  and  $\approx 0.88$  seconds respectively. There is also a risk of drift in the steady state which is discussed in Section 5.4.

For the *neural network ESC* method, higher amplitude oscillations are observed in the steady state. The reason is that, because of the dynamics of the system, the system does not stop immediately after the first threshold is exceeded



**Figure 5.2:** The simulation results with three different noise levels. The blue, green and red dashed lines are the results with noise amplitudes 0.05, 0.1, and 0.2 respectively.



**Figure 5.3:** The simulation results with a dynamic model of a six DOF robot.

which causes the system to exceed the second threshold as well. The rise time and settling time values are again very close to the sliding mode ESC; the values are  $\approx 0.57$  and  $\approx 0.8$  seconds respectively

*Approximation based ESC* is quite robust to the robot dynamics. Since the approximation phase uses the current state values of the robot, it is not affected by the dynamics. The rise time and settling time values are smaller than the previous methods with values  $\approx 0.52$  and  $\approx 0.58$  seconds. The *perturbation based method* is again very robust to the robot dynamics and performs very close to the dynamics free cases.

### 5.3.2 Two dimensional Simulations:

In this section, the motion generated by the algorithms in the two dimensional case is analyzed. No noise or system dynamics are used (like in Subsection 5.3.1) in order to examine the motion characteristics separately. The total distance traveled until 95% of the optimum value is reached is also given for numerical comparison. Since there is no extension of sliding mode ESC to the multivariate case, only the other algorithms are considered. The objective function is defined as a multivariate Gaussian based function:

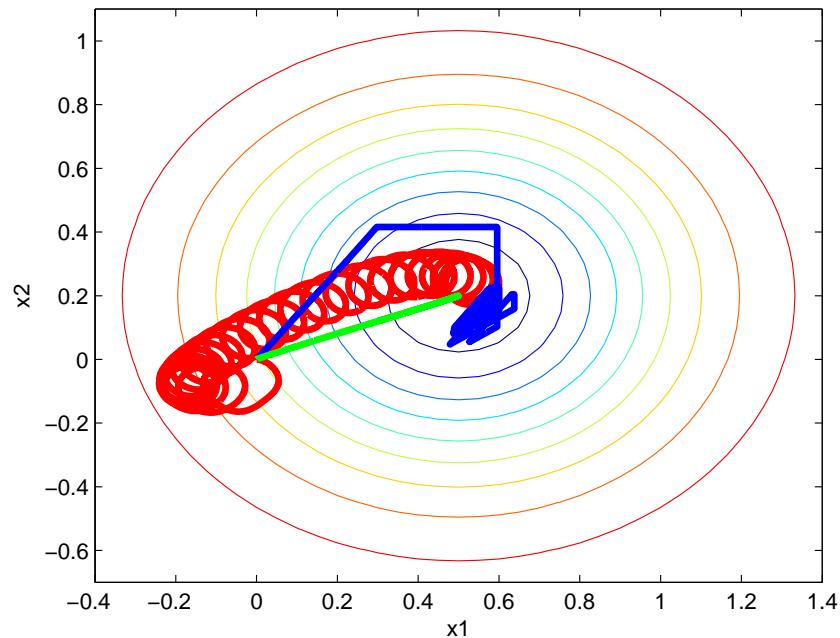
$$y = 2 - e^{-0.5(x-\mu)^T \Sigma^{-1}(x-\mu)} \quad (5.22)$$

Here, the mean vector  $\mu$  and the covariance matrix  $\Sigma$  are selected as  $\begin{pmatrix} 0.5 \\ 0.2 \end{pmatrix}$  and  $\begin{pmatrix} 0.15 & 0 \\ 0 & 0.15 \end{pmatrix}$  respectively.

The results are presented in Figure 5.4. For *neural network based ESC*, the discretization of the search space can be seen clearly. The total distance traveled is  $\approx 0.81$  meters. For *approximation based ESC*, it can be seen that the objective value is reached following an almost optimal route with a total distance of  $\approx 0.41$ . The result for *perturbation based ESC* clearly shows the circular motion in task space with a traveled distance of  $\approx 6.73$  meters.

## 5.4 Analysis

Effectively, sliding mode ESC and neural network ESC functions almost the same when there is no noise and dynamics effects. However, for sliding mode ESC, there should be a balance between  $\dot{g}(t)$  and  $\frac{dy}{dx}$  in order to avoid chattering. In neural network ESC, the driving input moves together with the objective value. This makes the parameter tuning of the neural network method easier. The approximation based method generates very smooth references for the system, and looks like the best choice for the systems without noise. It does not oscillate around the optimum value and has a very small steady state error. The perturbation based method looks like overkill for systems with negligible noise and dynamics. The



**Figure 5.4:** The simulation results of neural network ESC (blue) the approximation based ESC (green), and the perturbation based ESC (red) for the two dimensional case: The algorithms lead the system to the minimum value of the Gaussian like function.

rise time, settling time and the total distance results are all significantly larger than the other methods.

It is seen that the noise in the measurement channel affects all the methods except the perturbation based one, however the settling time of this method is still larger. The neural network ESC also shows robustness by its hysteresis mechanism except for the small steady state irregularity, and it shows the best performance in settling time. In the sliding mode based approach, the transient response is also affected, but high frequency oscillations when the velocity reference is changing are a greater problem. The performance of the approximation based method also drops due to the degradation of smoothness of the generated velocity reference, and higher settling time is observed.

It is seen that sliding mode ESC and neural network ESC show a performance decline due to the dynamics effects. However, for sliding mode ESC there is a potential risk of drift of the system in steady state. The reason of this is the growth of the driving signal. Actually, due to the design of the driving signal in (5.2), it increases all the time, even after the optimum value is reached. Therefore, the error between the driving signal and the objective value grows, and the error signal loses its meaning for the control law in (5.4); it only causes oscillations as the sine function changes sign with the growing error. In order to solve this problem, the driving input can be designed similar to the one in [90] so that if the objective value is unable to track the driving signal, the driving signal takes some steps back. Another option may be to use some heuristics. If the dynamic effects are not too large, the neural network ESC method is still usable. However, as a result of the smooth velocity reference generation, the approximation based method shows the best performance. Perturbation based ESC again presents a very robust result under the dynamic effects too. Considering its performance in for all of the cases, this method is preferable due to the smooth references and when consistency is an issue.

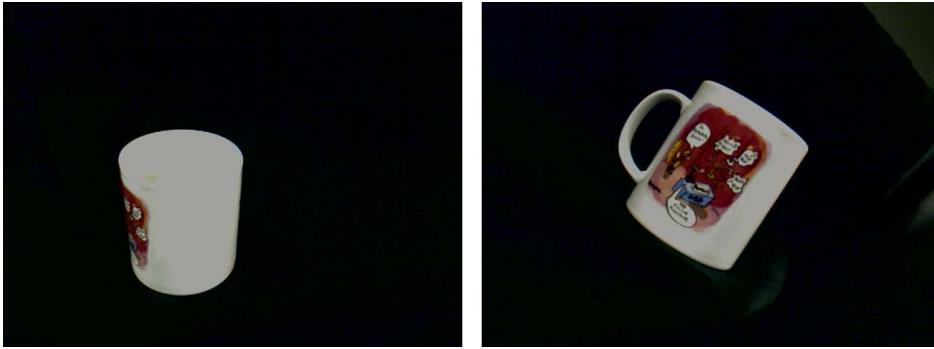
Among the algorithms that have been extended to the multivariate case in the literature, the approximation based method looks the best. However, it should be noted that, in the presence of noise, the performance of the algorithm will be degraded as in Section 5.3.1. The neural network ESC presents a reasonable result with a good convergence rate. On the other hand, the perturbation based algorithm makes the system travel long distances. All these results are presented in Table 5.1.

## 5.5 Experimental Results

In this section, we show that ESC works for our application area by giving an indicative example. A six DOF UR5 type Universal Robots arm is used with a camera mounted on the tooltip. In this experiment, the goal is to change the viewpoint of an object in order to maximize textured surface. This kind of system can be used to increase the success rate of an object recognition system.

	Effect of noise	Effect of system dynamics	Smooth References	Multivariate Extension / Traveled distance
Sliding mode ESC	Distortion in transient, high frequency oscillation in steady state	Minor distortion at transient, risk of drift	No	No
Neural Network ESC	Minor irregularities in steady state	Minor distortion in the steady state	No	Yes / medium
Approximation based ESC	Distortion at the generated velocity, performance fall at the steady state	Robust	Yes, if noise is negligible	Yes / low, very close to optimum
Perturbation based ESC	Robust	Robust	Yes	Yes / large

Table 5.1: Summary of the analysis of the analog ESC algorithms.



**Figure 5.5:** Snapshots from the experiment. The robot starts with the initial image on the left. The neural network ESC algorithm provides references in order to maximize the texture density and the snapshot on the right is achieved at the final position.

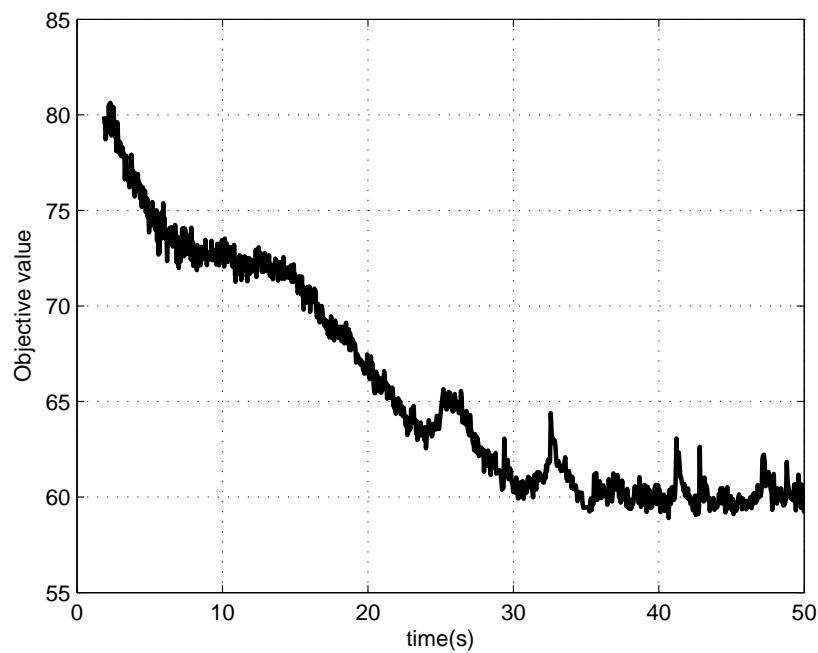
The texture density value is calculated by counting the edge density over a region, and it is subtracted from the maximum possible value.

The velocity references of the ESC algorithm are fed to the translation motions in the  $x$  and  $y$  directions of the task space. The orientations around the  $x$  and  $y$  axes of the tooltip are controlled in order to keep the object always at the center of the image. The translation in the  $z$  direction and the orientation around the  $z$  axis are not controlled. Based on our analysis in Section 5.3, we chose to use the neural network ESC algorithm since we have noise on the texture density calculation and low dynamics effects.

The initial and final views of the object are given in Figure 5.5. The objective value with respect to time is given in Figure 5.6. The ESC algorithm successfully minimizes the objective value and leads the robot to a much better view point for recognition purposes, indicating that the algorithm is indeed a good fit to our experimental conditions. In these experiments, since the workspace of the robot is limited, we needed to introduce the workspace constraints to the ESC problem. We have noticed that this can be easily achieved in the neural network ESC by forcing a direction change (just like it happens when the threshold of that direction is exceeded) whenever the workspace borders are reached.

## 5.6 Conclusion

In this paper, we categorized and summarized a number of analog extremum seeking controllers. They were analyzed for their usability in robotic applications. For the systems with negligible noise and dynamic effects, approximation based methods look like the best option. When noise is present, neural network ESC gives robust and efficient results compared to the other algorithms. Under dynamic effects, the approximation based method presents efficient and robust



**Figure 5.6:** Objective value plot for the experiment.

results. On the other hand, it is observed that, the perturbation based method gives very consistent results with smooth references under both high noise and dynamic effects.

## Final Notes

Among the extremum seeking control methods that are analyzed in this chapter, neural network extremum seeking control appears to be the most suitable one for solving the viewpoint optimization problem for grasp synthesis since it is robust to objective value noise and robot dynamics. In the following chapter, this method is applied to our problem.

Chapter 6

# Grasping Unknown Objects with Active Exploration via Online Viewpoint Optimization

In the previous chapter, extremum seeking control algorithms are analyzed, and neural network extremum seeking control method is chosen to be the most suitable method for our viewpoint optimization problem. In this chapter, this method is applied to our problem.

## Abstract

In this paper, the problem of robotic grasping of unknown objects is formulated as a viewpoint optimization problem. Using this formulation, we present a framework which is made up of two main steps: evaluating the quality of the viewpoint for grasp synthesis, and changing the viewpoint in order to increase the quality. The optimization problem is solved by utilizing a continuous extremum seeking controller, which optimizes the viewpoint within the control loop. Experiments are conducted using an eye-in-hand system and six different objects with various initial viewpoints. 94% success is achieved for the power grasps. However, for precision grasp, the success rate fell to 33%, and the reasons are discussed. The presented framework is the first with a systematic exploration phase. Although analyzed here in conjunction with our highly efficient silhouette based grasping algorithm, this framework is also expected to increase the success rate of most other existing grasping algorithms.

## 6.1 Introduction

For robots which are designed to assist people in daily life, grasping objects is a key feature as it enables many manipulation based tasks. If the object to be grasped is known a priori, then the grasping problem can be formulated as an optimization problem over the 3D model of the object surface ([60, 26, 108]) for a set of grasping criteria ([155]). However, daily environments generally include a huge variety of objects. This brings the necessity of developing grasping algorithms for previously unknown objects for assistive robots.

In order to solve this problem, autonomous systems are developed which construct a full 3D model of the object prior to the grasping operation ([16, 175, 173]). This preparation step requires data to be acquired from various viewpoints of the object and run a reconstruction algorithm. This process is reported to be time consuming by many researchers, and although [104] proposed a parallel architecture for simultaneous grasp planning and model refinement in order to speed up the reconstruction phase, acquiring all the necessary data for the reconstruction is still time consuming and not always possible due to occlusions and workspace constraints of the robot.

Less time-consuming solutions include learning based methods and heuristics based methods that do not necessitate a 3D model for grasp synthesis. [141, 142] proposed a feature based method in which image features that correspond to good grasping points are trained by using synthetic data via supervised learning. [82] presented a two step learning algorithm for finding the full pose of the gripper from RGBD images. [14] used the shape context descriptor (a histogram like descriptor for a global representation of the shape) together with supervised learning. [76] used heuristics for various types of grasps such as top grasps, side grasps and high point grasps. [47] models the visible side of the object using Gaussian process implicit surfaces and utilizes task variables based on some heuristics.

All the heuristic and learning based methods mentioned above are faster than the previously mentioned 3D model based solutions, but in a sense they seem to have thrown the baby out with the bath water; from full 3D information they go to single-image information without utilizing the robot's ability to move while proceeding to grasp. The fact that they synthesize a grasp using the data acquired from one single viewpoint makes the pose of the object with respect to the sensor a crucial factor: When the object is neatly positioned in front of the sensor, these algorithms provide a very high success rate for a huge variety of objects. However, for many other viewpoints, the success rate of these algorithms drops significantly. This problem can be solved by changing the viewpoint of the robot's vision sensors actively. As it is indicated by [128], this kind of visual exploration is a crucial element in order to fill the gap between neuroscience models and robotic implementations of grasping.

In this paper, we propose an online viewpoint optimization framework for grasping purposes for an eye-in-hand system. In other words, we change the viewpoint of the object online in order to achieve higher quality grasps with re-

spect to the given grasp measures. The proposed framework is the first in the literature that addresses systematic exploration for grasping unknown objects. This kind of framework does not only bring new algorithmic possibilities to efficiently solve the unknown object grasping problem, but it can also increase the success rate of the existing algorithms. The next chapter gives an overview of this framework.

## 6.2 Overview

This section begins with the formulation of the viewpoint optimization problem for an eye-in-hand system. Following that, we present our new framework as a solution to that problem.

### 6.2.1 Viewpoint optimization problem

The purpose of the optimization is to alter the viewpoint of the robot's vision sensor by maximizing the viewpoint quality in order to achieve a better grasp synthesis. This problem is formulated as follows:

Let's denote the optical center of the camera as  $o_c$ , and attach a coordinate frame  $\{x_c, y_c, z_c\}$  to  $o_c$  where  $z_c$  is perpendicular to the image plane. Suppose that a coordinate frame is attached to the target object with an origin denoted by  $o_o$ . A viewpoint  $v \in SE(3)$  can be defined as the pose of the camera frame with respect to the object frame:

$$v = \{T_c^o, R_c^o\} \quad (6.1)$$

where  $T_c^o$  and  $R_c^o$  are the translation and orientation of the camera frame with respect to the object frame respectively. These elements can be written more explicitly as follows:

$$T_c^o = \begin{bmatrix} T_{c_x}^o \\ T_{c_y}^o \\ T_{c_z}^o \end{bmatrix} \quad (6.2)$$

$$R_c^o = [R_{cx}^o | R_{cy}^o | R_{cz}^o] \quad (6.3)$$

where  $T_{c_x}^o, T_{c_y}^o, T_{c_z}^o$  correspond to  $x, y$  and  $z$  components of the translation vector, and  $R_{cx}^o, R_{cy}^o, R_{cz}^o$  are the column vectors which are the representations of the unit vectors in  $x, y$  and  $z$  direction of the camera frame expressed in object frame.

Our aim is to search within the viewpoints for the best grasp synthesis, and since both translation vector and rotation matrix have three degrees of freedom, it is a six dimensional optimization problem. However, since we are only interested in the viewpoints for which the object is visible to the camera, we can limit our

search space to a subspace  $V$ , where  $z_{cam}$  passes through  $o_o$ . This can be achieved by the following constraint:

$$V = \{V \subset SE(3) : T_c^{o'} R_c^o = [ \begin{array}{ccc} 0 & 0 & |T_c^o| \end{array} ]\} \quad (6.4)$$

where  $T_c^{o'}$  is the transpose of the  $T_c^o$  vector. This constraint brings the object to the center of the image by imposing constraints on the orientation around  $x$  and  $y$  axes of the camera for a given camera position  $T_c^o$ , since:

$$T_c^o \cdot R_{cx}^o = 0 \quad (6.5)$$

$$T_c^o \cdot R_{cy}^o = 0 \quad (6.6)$$

where  $(\cdot)$  is the dot product. Thus, by selecting the search space as  $V$ , the dimension of the optimization problem is reduced from six to four. This constraint can be satisfied by keeping the center of the object at the image center via visual servoing.

The dimension of the optimization problem can further be reduced from four to two, if the viewpoint quality is designed to be scale and rotation invariant. In this case, the distance of the camera to the object and the remaining degree of freedom of rotation matrix, which is the rotation of the camera around the optical axis, will not affect the viewpoint quality. This corresponds to moving on a virtual sphere around the object while keeping the object always at the image center, and our new search space  $\bar{V} \subset V$  becomes

$$\bar{V} = \{(T_{cx}^o, T_{cy}^o) : T_{cx}^{o2} + T_{cy}^{o2} \leq \phi^2\} \quad (6.7)$$

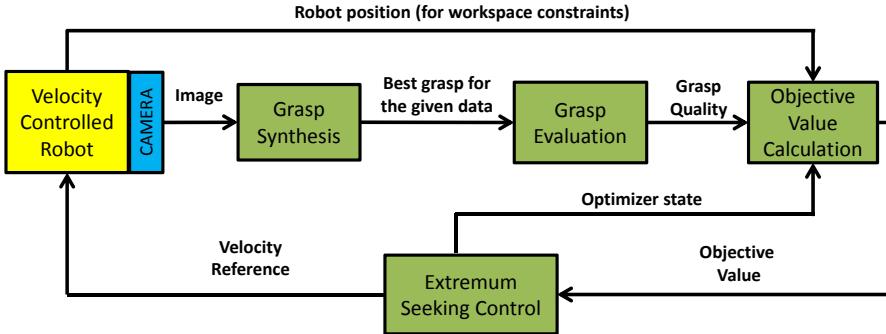
Here  $\phi$  is the radius of the sphere and can be kept constant.  $T_{cz}^o$  can be calculated in term of  $T_{cx}^o$  and  $T_{cy}^o$  from the equation of the sphere. In this paper, scale and rotation invariant quality measures are used. However, this dimension reduction is not a must, and the optimization problem can also be solved in three or four dimensions by the framework given in this paper.

The purpose of the optimization algorithm is to find a  $v \in \bar{V}$ , that has the maximum score for a given grasp quality criteria:

$$v^* = \arg \max_v f(v) \quad (6.8)$$

where  $f$  is the objective function which should be designed considering the grasp quality and  $v^*$  is the optimum viewpoint. Unfortunately, eq. (6.8) requires two additional inputs due to the nature of our system. First, robots typically have workspace constraints, and in order to impose these constraints, the state of the robot is needed. Second, depending on the type of the optimization algorithm, the objective function can also include algorithm specific elements to impose time constraints. Therefore, the objective function of the optimization does not only depend on the quality of the synthesized grasp, but also on the state of the robot  $x_s$  and on the state of the optimizer  $x_o$ :

$$v^* = \arg \max_v f(v, x_s, x_o) \quad (6.9)$$



**Figure 6.1:** The viewpoint optimization loop for an eye-in-hand system: Using the image captured from the eye-in-hand system, a grasp is synthesized and its quality is evaluated. This grasp quality is used while calculating the objective value together with the robot’s workspace constraints. The objective value is then send to the extremum seeking control algorithm which generates velocity references for the velocity controlled robot in order to maximize the objective value, therefore the grasp quality.

In the next subsection, we present our new framework that solves this optimization problem.

### 6.2.2 Proposed framework

The viewpoint optimization framework for an eye-in-hand system is given in Fig. 6.1. In this framework, the first step is the grasp synthesis using the image acquired from the current viewpoint. Then, the quality of this grasp is calculated and used as the objective value for the viewpoint optimization. Note that the objective value is also influenced (to a lesser degree) by workspace constraints and the optimizer state, cf. eq. (6.9). At the last stage, an extremum seeking control algorithm maximizes this objective value (thus the quality of the grasp) by changing the viewpoint of the sensor via supplying velocity references to the eye-in-hand system.

Within the above-mentioned framework, for grasp synthesis, we used a silhouette based algorithm in which the grasping point locations are optimized on a parametric model of the silhouette considering the force closure and curvature of the contact surface based on the idea presented by [23]. Using silhouette data in the viewpoint optimization framework is a natural choice, since it is clearly a function of the viewpoint. This grasp synthesis is given in Section 6.3.

Since the model of the object is unknown, the objective function based on the grasp quality cannot be calculated for an unvisited viewpoint. From the optimization perspective, this means the objective function that we want to optimize is

unknown. Moreover, the gradient of this function is not calculable and it cannot be measured directly. We can only calculate the value of the objective function for our current state, or in other words, our objective value. For this kind of optimization problems, a suitable solution is the use of continuous extremum seeking control techniques. The objective value calculation is explained in Section 6.4, and an extremum seeking controller design for our viewpoint optimization problem is given in Section 6.5.

The performance of the proposed scheme was evaluated by experiments using an eye-in-hand system. In these experiments, six household objects are used and the robot was initialized with six different initial viewpoints. These experiments are presented in Section 6.6.

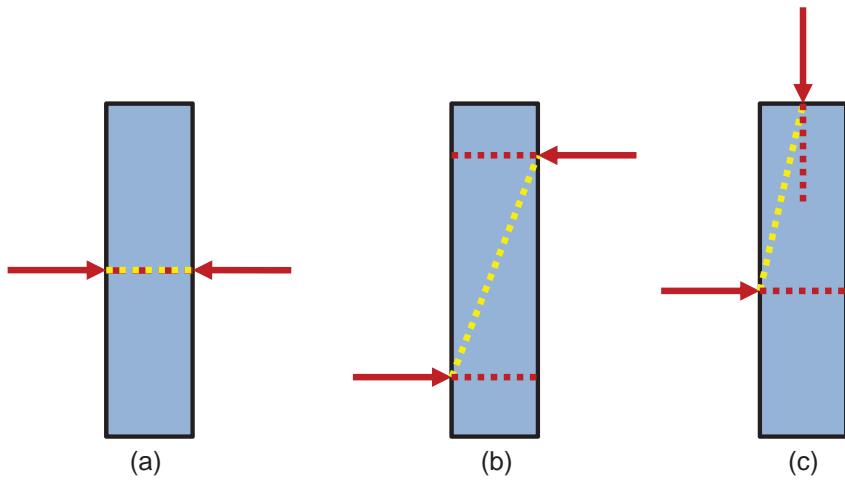
## 6.3 Grasp Synthesis

For the viewpoint optimization, we use the quality of the synthesized grasp as the main optimization criterion. In this section, a silhouette based grasp synthesis algorithm is presented for two-finger grasps. First the grasping strategy is explained in Section 6.3.1. Then, obtaining a parametric model of the silhouette and calculating the optimum grasp according to this strategy using the parametric model are presented in Sections 6.3.2 and 6.3.3 respectively.

### 6.3.1 Grasping strategy

While designing the grasp, we used the first and second order effects of grasping which are force closure and the curvature of the contact surface respectively ([11]). Force closure is a widely used static force and moment balance analysis. Considering also the friction between the surface of the target object and the gripper, the quality of this feature can be measured by calculating the area of the convex hull of the grasp polygon ([108]). The result of this calculation is an approximation since the friction at the contact points are generally unknown. In this paper, we will use another approximation of force closure quality for two-finger grasps; this approximation is presented by [99] as gradient angle features. This quality measure is efficient to compute over the parametrized models, and can be calculated as follows: Suppose that there are two grasping point candidates on the object. The quality of the grasp using these points can be calculated by the angles between force application vectors and the line that passes through the grasping points (Fig. 6.2). As the angles get closer to 0 and  $\pi$  radians, the quality of the grasp increases. Our other grasp quality measure is the curvature of the contact surface which is known as the second order effect of grasping. According to [132], the stability of the grasp increases as the object is grasped from a more concave surface. This effect is especially important while moving the object.

These measures are optimized on the parametric model of the silhouette data. A silhouette of the object is the projection of the object points to the image plane



**Figure 6.2:** An illustration of the gradient angle features that are introduced by [99]. For a given set of two grasping point locations, the gradient angle features can be calculated as angles between force application directions (the red dashed lines) and the line that connects the two grasping points (the yellow dashed line). These two angles are expected to be close to 0 and 180 degrees for a good grasp. These features are simple and efficient to calculate and can distinguish between good grasps (i.e. (a)) and bad grasps (i.e. (b) and (c)).

for a given viewpoint  $v$ . We use a global representation of the silhouette edge using Elliptic Fourier Descriptors (EFD). The next section explains EFD briefly and shows how to calculate the necessary information out of it.

### 6.3.2 Modeling using Elliptic Fourier Descriptors

Any closed and bounded curve can be represented by using Elliptic Fourier Descriptors (EFD). We use EFDs to model the edge of the silhouette of the target object ([23]). EFDs provide a representation of the closed curve that is composed of the sum of sine and cosine functions as follows:

$$P_x(m) = a_0 + \sum_{n=1}^k (a_n \cos(2\pi nm) + b_n \sin(2\pi nm)) \quad (6.10)$$

$$P_y(m) = c_0 + \sum_{n=1}^k (c_n \cos(2\pi nm) + d_n \sin(2\pi nm)) \quad (6.11)$$

where  $(P_x, P_y)$  are the points on the silhouette edge parametrized by  $m \in [0, 1]$ ,  $(a_0, c_0)$  is the model center that is calculated by the mean of the edge data,  $a_n$ ,  $b_n$ ,  $c_n$  and  $d_n$  are the Fourier coefficients, and  $k$  is the number of harmonics. The Fourier coefficients can be obtained by using edge data as explained by [96]. The precision of the model is determined by the number of harmonics  $k$ .

By using EFD, we have a parametric representation of the silhouette edge data. From this representation, valuable information can be extracted that can be used for the grasp synthesis as follows: The first derivative of the  $P_x$  and  $P_y$  functions gives the  $x$  and  $y$  components of the tangent vector for a given  $m$ . If the tangent vector is denoted by  $Z(m)$  and its  $x$  and  $y$  components are denoted by  $Z_x(m)$  and  $Z_y(m)$ , then the tangent vector can be calculated by

$$\begin{aligned} Z_x(m) &= \frac{d(P_x(m))}{dt} = \sum_{n=1}^k (-a_n(2\pi n) \sin(2\pi nm) \\ &\quad + b_n(2\pi n) \cos(2\pi nm)), \end{aligned} \quad (6.12)$$

$$\begin{aligned} Z_y(m) &= \frac{d(P_y(m))}{dt} = \sum_{n=1}^k (-c_n(2\pi n) \sin(2\pi nm) \\ &\quad + d_n(2\pi n) \cos(2\pi nm)). \end{aligned} \quad (6.13)$$

Likewise, the derivative of the normalized tangent vectors gives the normal vectors. Therefore the normal vectors,  $N(m) = \begin{bmatrix} N_x(m) \\ N_y(m) \end{bmatrix}$ , can be calculated by

$$N_x(m) = \frac{d(\frac{Z_x(m)}{|Z|})}{dt} \quad (6.14)$$

$$N_y(m) = \frac{d(\frac{Z_y(m)}{|Z|})}{dt} \quad (6.15)$$

where  $|.|$  is the Euclidean norm. From the equations of the normal vectors, the curvature of the model can be calculated as the norm of the normal vectors as

$$C(m) = |N(m)|. \quad (6.16)$$

where  $C(m)$  is the curvature value for a given point  $m$ . Although the curvature value can be calculated with this expression, it cannot be understood whether the point is in a convex or a concave region. In order to differentiate between convex and concave regions, the cross product of tangent and normal vectors of the corresponding point can be used. The sign of the  $z$  component of the cross product signifies whether the region is convex or concave. If the edge data is read in a clock-wise direction, then positive values correspond to concave regions and vice versa. We will use the signed curvature which is positive if the surface is concave and negative otherwise:

$$\bar{C}(m) = \text{sign}(z_p)|C(m)| \quad (6.17)$$

where  $z_p$  is the  $z$  component of the above-mentioned cross product as the data is read clock-wise.

With the EFD representation, the normal vectors and the signed curvature information can be calculated for a given point on the silhouette. In the next subsection, these equations will be used for the optimization of the grasping point locations.

### 6.3.3 Calculating the best grasp

Using the model of the silhouette obtained by EFD, we find the best grasping point pair via optimization by considering force closure and curvature of the contact surface, as follows: Denote the positions of two grasping points on the EFD model as  $m_{g_1}$  and  $m_{g_2}$ . An objective function that measures the quality of the grasp  $Q(m_{g_1}, m_{g_2})$  by using the first and second order effects of grasping can be formulated as:

$$Q(m_{g_1}, m_{g_2}) = w_c * (\bar{C}(m_{g_1}) + \bar{C}(m_{g_2})) - w_f * ((\alpha_1)^2 + (\pi - \alpha_2)^2) \quad (6.18)$$

where  $w_c$  and  $w_f$  are the weighting scalars, and

$$\alpha_i = \text{acos} \left( N(m_{g_i}) \cdot \left( \begin{bmatrix} P_x(m_{g_i}) \\ P_y(m_{g_i}) \end{bmatrix} - \begin{bmatrix} P_x(m_{g_i}) \\ P_y(m_{g_i}) \end{bmatrix} \right) \right). \quad (6.19)$$

Then, the optimization in order to find the optimum grasp for a given silhouette can be written as:

$$G^* = \arg \max_G Q(m_{g_1}, m_{g_2}) \quad (6.20)$$

where  $G = \begin{bmatrix} m_{g_1} \\ m_{g_2} \end{bmatrix}$ , and  $G^*$  is the optimum grasping point pair. This optimization problem can be solved using gradient ascent and the result is the optimum grasping point locations. The quality of this optimum grasp is used as the main component of the objective value calculation as it is explained in the following section.

## 6.4 Objective Value Calculation

The core of the viewpoint optimization for grasping is the optimization criterion, i.e. the objective value. The grasp quality value, which is calculated for the optimal set of grasping points in the previous section, forms a key part of this objective value. In this section, an objective function  $f$  that gives us this objective value is proposed for the optimization problem in eq. (6.9). This function is made up of three components:

$$f(v, x_s, x_o) = f_g(v) + f_w(x_s) + f_o(x_o) \quad (6.21)$$

In this equation,  $f_g(v)$  is the component related to the grasp quality,  $f_w(x_s)$  is used to impose workspace constraints and  $f_o(x_o)$  is the optimizer specific component to impose time constraints to the optimizer. These components are explained in the following subsections.

### 6.4.1 Grasp Quality Component

As the optimum grasp is found for the current viewpoint of the object (eq. (6.20)), the quality of this grasp can be used as the main criterion of the viewpoint optimization:

$$f_g(s(v)) = Q(G^*) \quad (6.22)$$

It is important to note that, to reduce the dimension of the optimization problem as it is explained in Section 6.2.1, the EFD model should be normalized so that the curvature related components of the quality measure become scale invariant; all the components of the measure is already rotation invariant.

### 6.4.2 Workspace Constraints Component

The workspace constraints of the robot can be imposed to the viewpoint optimization by forming a barrier function ([73]). By this way, instead of solving a constrained optimization problem, the constraints are added to the objective function. This can be achieved by using the following  $f_w$  component:

$$f_w(x_s) = \mu_w \sum_{n=1}^k \ln(x_{wn} - x_{sn}) \quad (6.23)$$

Here  $x_{sn}$  is the state of the robot,  $x_{wn}$  is the corresponding workspace constraints and  $\mu_w$  is the barrier parameter. This function drops rapidly as the robot approaches to the close vicinity of the workspace constraints.

### 6.4.3 Optimizer specific component

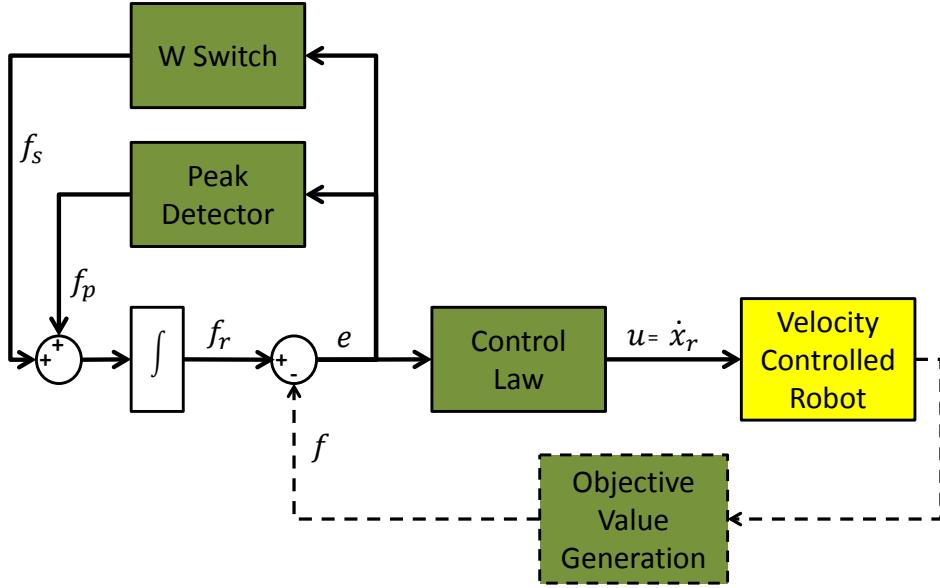
While conducting viewpoint optimization, some objects (i.e. objects with cylindrical symmetry) may have zero gradient in certain directions in terms of grasp quality. This may make some of the optimization algorithms stuck in these states, and the efficiency of the algorithm degrades. If such an optimizer is used (i.e. the extremum seeking control algorithm explained in Section 6.5), then imposing a time constraint by a barrier function helps to increase the efficiency of the viewpoint optimization algorithm. The following  $f_o$  function is used for this purpose:

$$f_o(x_o) = \mu_s \ln(\tau_o - \tau_p(x_o)) \quad (6.24)$$

Here  $\tau_p$  is the elapsed time since the maximum value of  $f_g$  is measured during current state of the optimizer. The value of this component will drop rapidly as the state of the optimizer does not cause the objective value to increase until the allowed time  $\tau_o$ . This will act as a penalty component in the objective function, and the state change for the viewpoint optimizer will be triggered.  $\mu_s$  is the barrier parameter of this component. In the next section, an extremum seeking controller will be designed in order to maximize the objective function  $f(v, x_s, x_o)$ .

## 6.5 Extremum Seeking Controller

As the optimum grasp is synthesized for the current viewpoint (Section 6.3), and the objective function is formed by using the quality of this grasp (Section 6.4), the next step of the presented framework is the viewpoint optimization algorithm. The purpose of the viewpoint optimization is to find the viewpoint that has the maximum objective value, thus the highest quality grasp. In this optimization, since the model of the target object is unknown, the value of the objective function cannot be calculated without visiting the corresponding viewpoint. From the optimization perspective, this means the objective function and its gradient is unknown. In order to solve this kind of optimization problems, extremum seeking control (ESC) techniques can be used. These algorithms does not rely on the knowledge of the objective function and only use the current objective value for optimization purposes. Moreover, if calculating the objective value continuously is possible (which is the case for our problem, thanks to the efficiently computable objective function), the optimization can be conducted by using continuous ESC methods. The continuous ESC methods provide continuous motion signals (i.e. Cartesian velocity signals for the robot) using continuously evolving objective value due to this motion.



**Figure 6.3:** A scheme for neural network extremum seeking control algorithm.

There are various types of continuous ESC techniques ([22]). Among the techniques that can be extended to the multivariate case, we chose to use neural network ESC, since this algorithm can give efficient results even under objective value noise. We didn't prefer to use approximation based ESC and perturbation based ESC since the former one is not robust enough to the noise on objective value and although the latter one is robust to the noise, it makes the system travel long distances ([22]). Now we will design a neural network ESC for our viewpoint optimization problem.

### Neural network ESC:

Neural network ESC is proposed by [160, 161]. This method provides velocity references to the system by analyzing the objective value using a set of switching functions. We will now design the neural network ESC for our viewpoint optimization problem in eq. (6.9). As explained in Section 6.2.1, the dimension of the optimization problem is reduced to two. Therefore we need to design a two dimensional neural network ESC.

A scheme for neural network ESC is presented in Figure 6.3. As it can be seen from this figure, the neural network ESC generates its own reference,  $f_r$ , by using the peak detector and the W switch:

$$f_r = f_p + f_s \quad (6.25)$$

In this equation,  $f_p$  is the output of the peak detector and  $f_s$  is the output of the W switch. These switches use the error that is obtained by the difference between the current objective value and the reference value:

$$e = f_r - f \quad (6.26)$$

As the algorithm is initialized with the initial objective value ( $f_r = f$ ), the W switch is not active; the only contribution to the  $f_r$  value comes from the peak detector. The switching mechanism of the peak detector is as follows:

$$\dot{f}_p = \begin{cases} M & (e < 0) \\ 0 & (e \geq 0) \end{cases} \quad (6.27)$$

where  $M$  is a positive constant. By this mechanism, the peak detector converges to the value of the maximum objective value that is measured during the process. In order to maintain this convergence, the rate of change of the objective value should be smaller than  $M$ :

$$|\dot{f}| < M \quad (6.28)$$

The control law  $u$ , which is fed to the robot as velocity reference, tries to minimize the error by the following switching functions with hysteresis:

$$u_1 = \begin{cases} -U & e < -\delta_1 \\ U & e > \delta_1 \\ [previous\_state] & \text{otherwise} \end{cases} \quad (6.29)$$

$$u_2 = \begin{cases} 0 & e < -\delta_2 \\ -U & e > \delta_2 \\ [previous\_state] & \text{otherwise} \end{cases} \quad (6.30)$$

$$u_3 = \begin{cases} 0 & e < -\delta_3 \\ 2U & e > \delta_3 \\ [previous\_state] & \text{otherwise} \end{cases} \quad (6.31)$$

$$u = \begin{bmatrix} u_1 + u_2 \\ u_2 + u_3 \end{bmatrix} \quad (6.32)$$

Here,  $U$  is a constant that specifies the magnitude of the velocity reference, and  $\delta_i$  are the hysteresis widths. The control law should be initialized with the following values:

$$(u_1, u_2, u_3) = (-U, 0, 0) \quad (6.33)$$

The switching mechanism of the control law works as follows: If the direction of the system makes the objective value converge to the optimum value, then this direction is kept. Otherwise, the error will increase, and when it is greater

than the hysteresis values, the direction will be changed. This nested structure of hysteresis functions brings the following constraint to the hysteresis widths:

$$\delta_1 < \delta_2 < \delta_3 \quad (6.34)$$

When  $e$  becomes greater than  $\delta_3$ , the control law enters its last state. This state ends as the error is larger than the threshold  $\Delta$  where

$$\Delta > \delta_3. \quad (6.35)$$

As the  $\Delta$  threshold is exceeded, the reference value  $f_r$  should be reset, so that a new cycle can start. The  $f_s$  function performs this by the following switching function:

$$\dot{f}_s = \begin{cases} -W & e > \Delta \\ 0 & e < -\Delta \\ [previous\_state] & \text{otherwise} \end{cases} \quad (6.36)$$

Here,  $W$  is a positive constant. By this function, when the error is greater than the  $\Delta$  threshold, the  $f_r$  will decrease until the error is less than  $-\Delta$ . This makes the control law to go back to its first state.

The control signal  $u$  is fed to the systems as velocity input:

$$\dot{x}_r = u \quad (6.37)$$

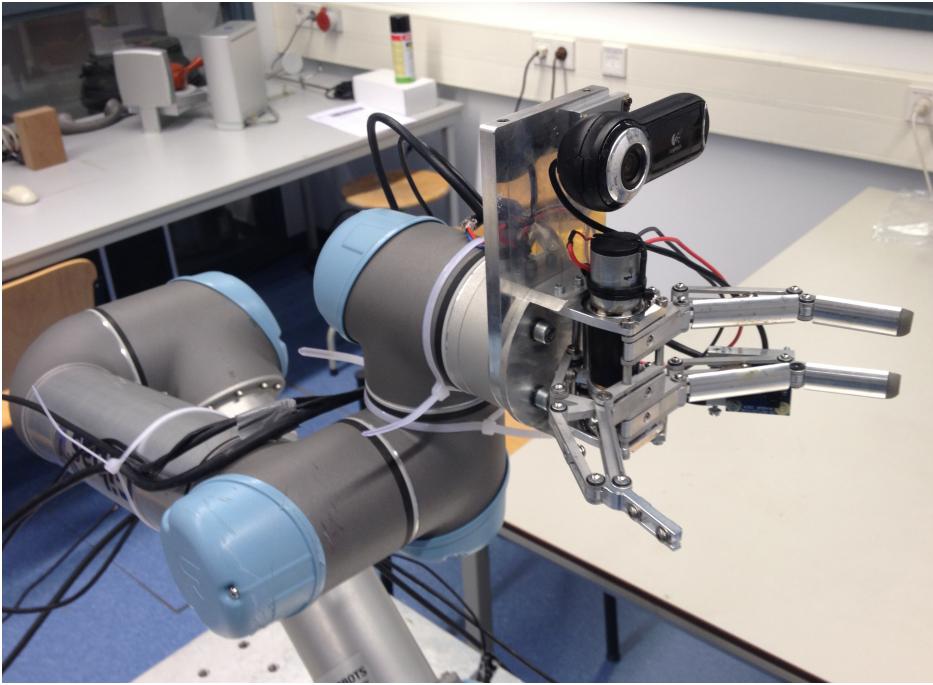
As it is mentioned in Section 6.4.3, the neural network ESC algorithm can get stuck in a state that does not cause a significant change in the objective value, since none of the hysteresis values will be exceeded. In order to make the algorithm more time efficient,  $\tau_p$  of the equation (6.24) can be selected as the time elapsed since the last change of  $f_p$  for a given state. By this way, the state change will be triggered if  $\tau_p$  approaches to  $\tau_o$ .

The performance of the overall system is examined by experiments with an eye-in-hand system. These experiments are given in the next section.

## 6.6 Experimental Results

The experiments are conducted using a six degrees of freedom UR5 type Universal Robots arm. The gripper that we used is Delft Hand 3 ([92]) which is an under-actuated gripper. This gripper can apply a power grasp as well as a precision grasp. A distance sensor, which is used to close the gripper when the object is close enough, is placed at the palm of the gripper. A webcam is attached to the tool tip of the robot to form an eye-in-hand system which can be seen in Fig. 6.4. The robot is controlled in task space as follows:

- Translation in  $x$  and  $y$  direction are controlled via the neural network ESC algorithm.



**Figure 6.4:** The eye-in-hand system formed with a UR5 type universal robot arm, a Delft Hand 3 gripper and a Logitech webcam.

- Translation in  $z$  direction is not controlled while the neural network ESC is active. Whenever the viewpoint optimization is done, this degree of freedom is used for approaching the object.
- Orientation in  $x$  and  $y$  directions are used for keeping the object always at the center of the image by image based visual servoing.
- Orientation in  $z$  direction is used to align the fingers of the gripper with the optimal grasping points for that location.

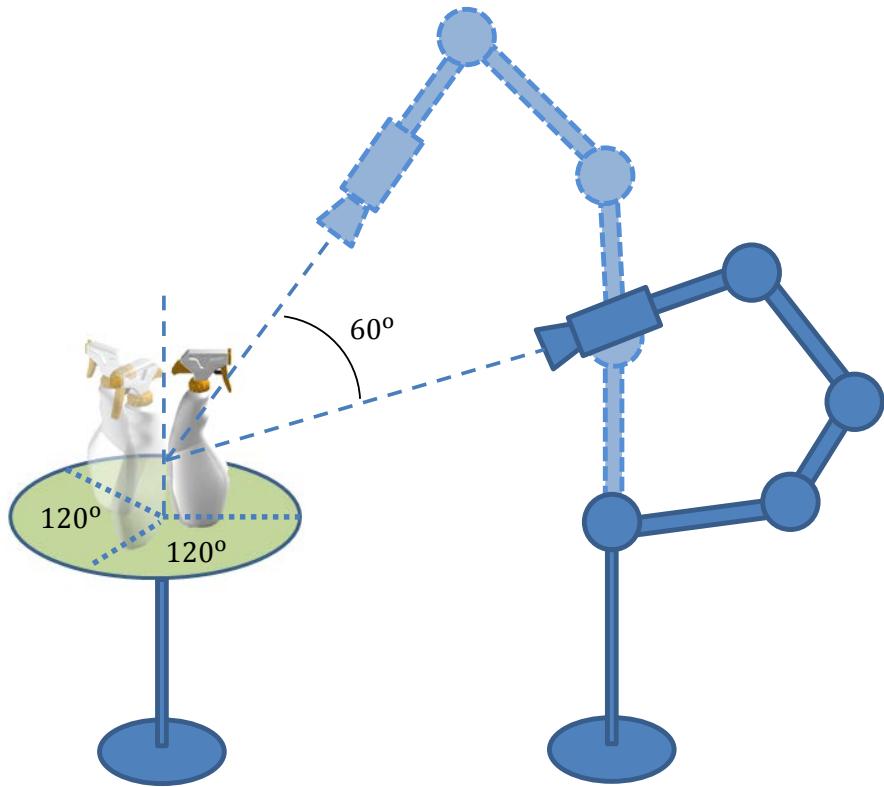
Six household objects with different shapes are selected as seen in Fig. 6.5. The objects are a spray bottle, a wine glass, a box, a water bottle, a beer bottle and a table tennis racket (Objects like large bowls are omitted in the test set, since it is not possible to detect grasping points on such objects using silhouette information). The algorithm is run for various initial viewpoints of the objects: For the objects without cylindrical symmetry, the objects are placed with 120 degrees rotation difference around the table plane normal, and for each pose, the robot has an up and down configuration which has 60 degrees rotation difference around the  $x$  axis of the task space (in total six different initial viewpoints for each



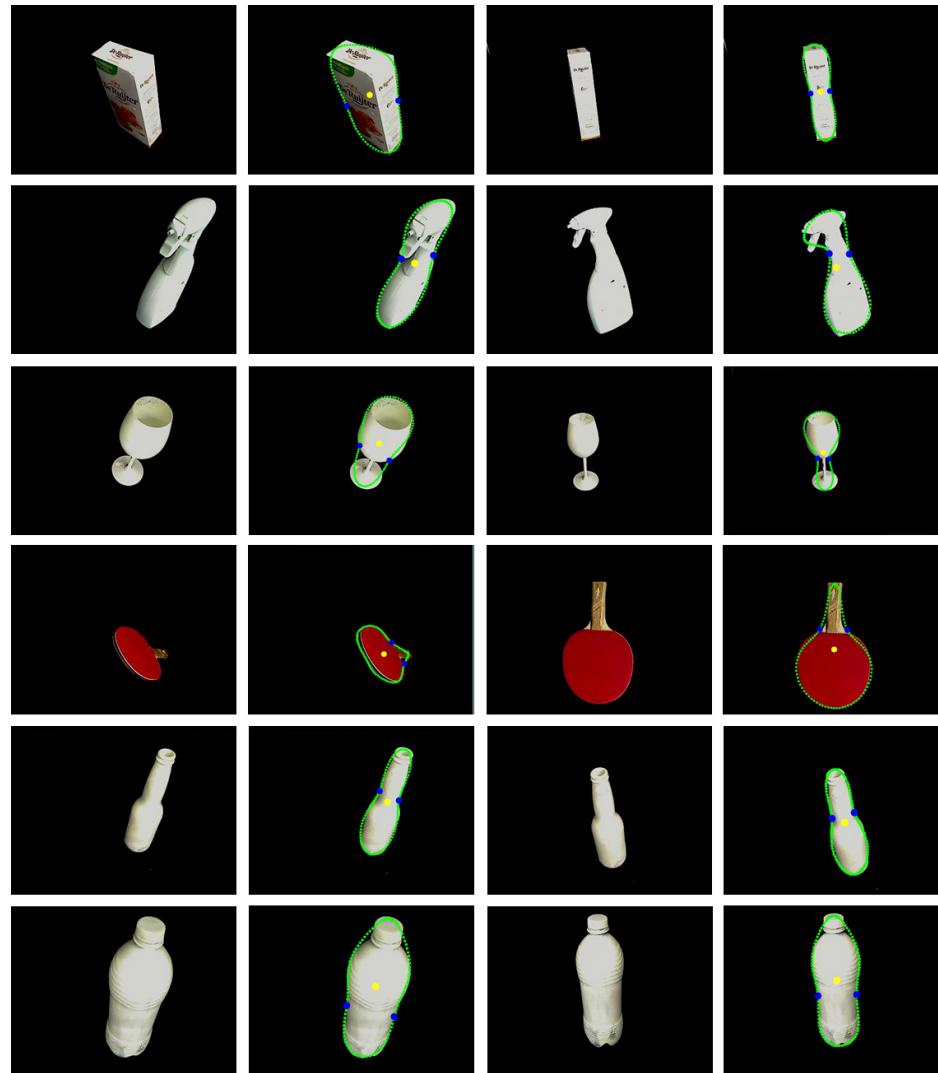
**Figure 6.5:** The objects used in the experiments. From left to right: a spray bottle, a water bottle, a box, a wine glass, a beer bottle and a table tennis racket.

object). Fig. 6.6 illustrates these different poses of the object and the robot. For the objects with cylindrical symmetry only up and down positions are considered. The distance threshold to grasp the object is set to power grasp for all the objects except for the table tennis racket. The experiments are conducted in a black environment in order to simplify the segmentation process, since segmentation is a only technical difficulty regarding the viewpoint optimization problem.

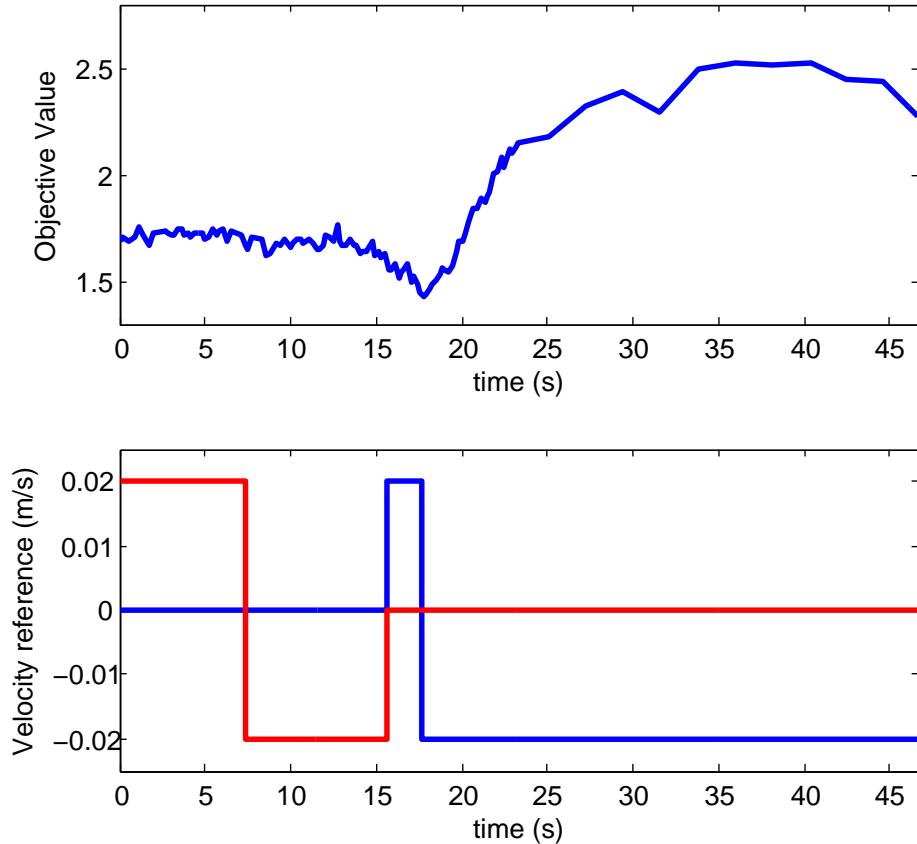
The results of the viewpoint optimization for an initial viewpoint for each object is given in Fig. 6.7. Comparing initial views with the result of the viewpoint optimization, it can be seen that, the viewpoints are optimized successfully for the combination of force closure and the curvature of the contact points. As a result, a viewpoint that is much more suitable for the grasp synthesis algorithm is achieved. Fig. 6.8 presents the objective value evolving in time and the generated velocity references for the spray bottle experiment whose snapshots are given in Fig. 6.7. Here, the velocity references in  $x$  and  $y$  direction are changed by the algorithm as the thresholds  $\delta_1$ ,  $\delta_2$  and  $\delta_3$  are exceeded. The algorithm is stopped and grasping is initiated as the  $\Delta$  threshold is exceeded. In all the experiments, it is seen that, the viewpoint optimization was successful in increasing the viewpoint quality with respect to the given measures. This aids the grasp synthesis algorithm, and the grasping points are placed to much better locations on the object. Apart from that, the resultant viewpoints provide better approach vectors for grasping. These advantages bring successful grasps for the box, wine glass, beer bottle and water bottle for all different initial viewpoints. For the spray bottle, only one initial view is failed. This makes up 94% success for the objects that are grasped with



**Figure 6.6:** The illustration of the experiments: The three different orientations for the object with  $120^\circ$  rotation difference and two different initial positions of the robot which has  $60^\circ$  rotation difference around the x axes of the camera frame.



**Figure 6.7:** Snapshots from the experiments: First and third columns are the initial views and the views obtained after the viewpoint optimization respectively. Second and forth columns presents the EFD model and synthesized grasping points for these views; the green points are the points on the EFD model, the yellow points are the centers of the models ( $(a_0, c_0)$  points) and the blue points are the grasping points. Comparing the initial viewpoints with the final one, it can be seen that, the viewpoint optimization does not only help the grasping point generation algorithm to generate better grasping points, but it also provides a good approach vector for the grasp execution.



**Figure 6.8:** The plots of the objective value and velocity reference generated ESC algorithm for the spray bottle experiment that is presented in the second row of Fig. 6.7. In the velocity reference plot, the red and blue lines show the velocity references in  $x$  and  $y$  directions respectively.

Object Name	Successful Exp. / Total Exp.
Box	6/6
Spray Bottle	5/6
Glass	2/2
Tennis Racket	2/6
Beer Bottle	2/2
Water Bottle	2/2
Total	19/24

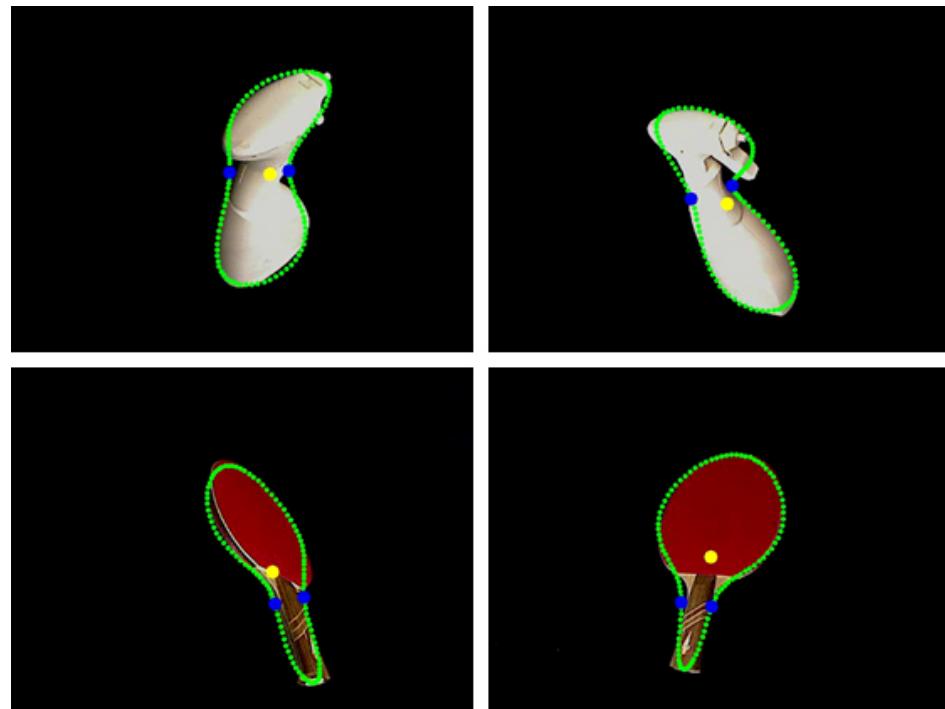
**Table 6.1:** *The summary of the experimental results.*

power grasps. However, for the table tennis racket, four failed out of six initial viewpoints. These results are summarized in Table 6.1.

The failed spray bottle experiment and one example of the failed table tennis racket experiments are presented in Fig. 6.9. The reason of the failure for the spray bottle was that the optimizer overshot a better viewpoint because of the hysteresis mechanism. As a result the viewpoint became harder for grasping, and that led to failure. The solution of this would be to increase the signal-to-noise ratio. This can either be done by trying to reduce the noise on the objective value, or adding more features while calculating the quality of the viewpoint, so that better viewpoints will provide much prominent quality measurements. For the table tennis racket, although the viewpoint optimizer led the robot to a better viewpoint for grasping synthesis, the grasp still fails because the precision grasp brings the necessity for the fingers of the gripper to be aligned precisely on the object. The algorithm fails to supply this precision, and since we do not use any depth information, this precision is hard to supply. Similar configuration imperfections appear also for the beer bottle and water bottle, however, since these objects were grasped by power grasps, these imperfections are negligible.

## 6.7 Discussion

In this chapter a novel viewpoint optimization framework is proposed for grasping purposes. The framework is unique as it uses the robot's vision sensors actively and systematically in order to increase the success rate of the grasp synthesis algorithm. We believe that such a framework can be used in order to increase the success rate of the other non-3D-model-based algorithms in literature (i.e. [142], [14]). The only necessity to enable such a viewpoint optimization is an objective value calculation that can measure the quality of the viewpoint for the grasping algorithm. It is also important that the objective value should be calculated efficiently so that it can be used continuously in the control loop for more efficient results.



**Figure 6.9:** Snapshots from the failed experiments: First column the initial views, second column the final views, both with EFD models. Green points are the points on the EFD model, yellow points are the centers of the models ( $(a_0, c_0)$  points) and blue points are the grasping points.

In the experiments, it is seen that the proposed combination of silhouette based grasping synthesis algorithm and neural network ESC provides successful results for power grasps. However, one problem is the noise on the objective value. Although the noise is compensated with the hysteresis mechanisms of the neural network ESC algorithm, keeping the hysteresis values high causes overshoot of the optimum viewpoint. Lowering the noise on the objective value can decrease the amount of the overshoot and that can increase the success rate of the algorithm significantly.

The reasons of the failures of the precision grasps are mainly due to the lack of the depth information. By using RGBD images, depth based quality criteria can also be included in the objective function. This will not only create much prominent attractive regions for the viewpoint optimizer, but it will also enable the use of the algorithm for the bowl like objects that had been excluded from the experiment set. Moreover, by using the depth information, obstacles on the approach vector can also be avoided by the minimization of the occluded regions by the viewpoint optimization.

## 6.8 Conclusion

In this chapter, an online viewpoint optimization framework is proposed for grasping purposes. This scheme enables systematic object exploration for a given grasp quality measure. Within this framework, a neural network extremum seeking controller is coupled with a silhouette based grasp synthesis algorithm. By using fairly simple viewpoint optimization criteria, 94% success is obtained in the experiments for power grasps. However it is seen that, improvements are necessary in order to use the algorithm for the precision type grasps. The utilization of depth information is a key factor for such an improvement. We believe that the proposed framework can provide novel possibilities for increasing the success rate of the grasping algorithms in literature.

## Final Notes:

This chapter presents a local viewpoint optimization with neural network extremum seeking control. This method is easy to implement as it does not require any training procedure, but failiures are reported due getting stuck at local optima. In the following chapter, a global viewpoint optimization method is presented that aims to escape from these local optima and achieve global optimum solutions.

Chapter 7

# Viewpoint Optimization for Grasp Synthesis via Supervised Learning

## Abstract

Grasp synthesis for unknown objects is a challenging field as the algorithms are expected to cope with missing information of the target object's shape. This missing information is a function of the viewpoint of the vision sensor. The majority of the grasp synthesis algorithms in literature synthesize a grasp by using one single image of the target object and making (implicit or explicit) assumptions on the missing shape information. Contrary to the present algorithms, this chapter proposes the use of a robot's vision sensor, actively and in a systematic manner. With such an approach it is aimed to relax the assumptions on the viewpoint of the sensor (thus the missing object shape information) and boost the abilities of the grasp synthesis algorithms. For this purpose, a methodology is proposed to integrate viewpoint optimization within the grasp synthesis process. The viewpoint optimization aims to increase the quality of the synthesized grasp over time by changing the viewpoint of the sensor. For achieving this, a supervised learning technique is employed to obtain a global viewpoint optimization policy. The training process and automated training data generation procedure are explained in detail. The methodology is applied to a force closure based grasp synthesis strategy. A thousand simulations with five objects with various shapes and initial poses are conducted in which the grasp synthesis algorithm was not able to obtain a good grasp with the initial viewpoint. The results demonstrate that in 94% of the cases, the viewpoint optimization policy enables a good grasp. It is also shown with the simulations that the proposed method is superior to a random and a heuristic based search techniques in terms of both success rate and efficiency.

## 7.1 Introduction

Grasp synthesis for unknown objects aims to enable robotic grasping even when the shape of the target object is not known by the robot a priori. This ability is crucial for robots that operate in unstructured environments e.g. homes, offices and disaster areas. To solve this problem, an option is to generate a full model of the target object prior to the grasp synthesis process and apply the grasp synthesis algorithms for known objects [7, 16, 174]. This approach is far from being efficient since the robot should acquire images from all the viewpoints that are necessary for shape reconstruction. Moreover, such an approach is not applicable when some surfaces of the object are occluded or the robot’s workspace is not sufficient. Therefore, grasp synthesis algorithms that do not rely on a full object shape information are needed.

In literature, various techniques are employed to solve the grasp synthesis problem with partial shape information, and almost all of these methods (the exceptions are given in section 7.2) use a single image of the target object and make implicit or explicit assumptions on the missing shape information of the target object. The feature based methods in [141, 142, 143, 99, 86, 82] aim to find features that correspond to good grasping locations. These approaches assume that the sensor observes the object from a specific viewpoint and train their system accordingly. The method in [12] assumes that the object is symmetric, it completes the shape model by calculating the symmetry axis and then synthesizes a grasp. The method in [64] fits shape primitives to the partial shape information and uses predefined grasps for the assigned shape. This approach assumes that the correct shape primitive can be fit from the given viewpoint and that the assigned primitive represents the shape well enough for grasp synthesis. The knowledge base method in [32] uses a database of objects represented by oriented bounding boxes together with corresponding grasps. As the partial shape of the target object is acquired, the oriented bounding box model is generated, the best match is found from the database and the associated grasp is applied. This method assumes that the oriented bounding box model that is fit from the current viewpoint is sufficient enough to find a good match from the database.

In the context of this paper it is important to note that the missing shape information is a function of the viewpoint of the sensor. Therefore, the assumptions on the missing object shape information is a function of the viewpoint of the sensor. Almost all algorithms in literature use one single image to synthesize a grasp. In these papers, experimental results are presented with several objects and various object poses. As we analyze these results, we see that in none of these papers, 100% success rate is reported. This shows that the assumptions on the missing information (or the viewpoint) do not always hold, even for a limited set of objects. Moreover, for none of the objects, a 0% success rate is reported. This implies that these algorithms can have successful results if good viewpoints are supplied to them.

This chapter presents a methodology for using a robot actively in the grasp

synthesis process by changing the viewpoint of the vision sensor in order to aid the grasp synthesis algorithms. For this purpose a viewpoint optimization methodology is proposed for eye-in-hand systems. The viewpoint optimization algorithm leads the sensor to a global optimum viewpoint so that a sufficiently good grasp can be obtained by the grasp synthesis algorithm. For each visited viewpoint, the collected data about the object shape is fused with the previous ones in order to benefit from all the data that is acquired during the process.

We applied this methodology to an eye-in-hand system made up of a manipulator and a depth sensor attached to the end effector which supplies a point cloud of the scene. A force closure based grasp synthesis strategy is utilized, and in the data fusion stage, the point clouds collected from the visited viewpoints are stitched.

For obtaining a viewpoint optimization strategy, a supervised learning scheme is used. A viewpoint optimization policy is obtained by training the system with partial point cloud data that are labeled with the direction to follow in order to lead the vision sensor to the optimum viewpoint. The output of the training process is one single policy that provides an exploration direction for given partial point cloud data. It is important the note that the resulting policy provides a generalized viewpoint optimization strategy that is applicable to the objects outside the training data set. Since obtaining the training data with manual labelling is a cumbersome process, an automated training data generation procedure is also presented.

The proposed viewpoint optimization method can be applied to various grasp synthesis techniques in literature for increasing their success rate. Naturally, it can only be applied to systems with movable sensors. The methodology also requires a quality value both for training the system and terminating the viewpoint optimization process. This quality value can be the quality of the synthesized grasp as well as the quality of the data. This value can be supplied by the majority of the algorithms in literature as most algorithms already use a quality measure for their internal grasp optimization process.

Simulations are conducted with the proposed method to evaluate its performance and to compare it with random and heuristic based exploration techniques: Five objects are chosen with various shapes and a thousand simulations are conducted with each method for random poses of the objects. The results demonstrate that the proposed viewpoint optimization enables a good grasp for the 94% of the cases in which synthesizing a good grasp was not possible for the grasp synthesis algorithm from the initial viewpoint. It is also seen that the proposed method is superior to the random and heuristic based exploration techniques techniques in terms of both success rate and efficiency.

This paper is organized as follows: First, the global viewpoint optimization methodology is presented in Section 7.2. Following that the details of our implementation is given in Section 7.3. The simulation results are presented and discussed in 7.4. In the last section, the paper is concluded with possible future advancement.

## 7.2 Global Viewpoint Optimization Methodology

The purpose of viewpoint optimization is to change the viewpoint of the sensor to increase the quality of a synthesized grasp. Only other approaches that study viewpoint optimization for grasp synthesis are proposed in [23, 24]. They propose a local viewpoint optimization scheme that is easy to implement as they do not need any training procedure, but at the risk of getting stuck in local optima. On the contrary, the method presented in this paper has a lower risk of getting stuck at local minima, but requires a training process.

Let's start by explaining the proposed exploration scheme.

### 7.2.1 The Viewpoint Optimization Scheme

In this section, an overview of the viewpoint optimization scheme is presented. For this scheme to be applicable, the robot needs to have a movable sensor. The sensor is constrained to move on a view sphere around the object as in [24, 119]. Therefore, the sensor is always pointed to the target object and its position on the virtual sphere which is defined around the object is altered by a viewpoint optimization strategy.

The proposed viewpoint optimization scheme can be seen in Figure 7.1. First, data are acquired from the sensor and fused with the data acquired from the other visited viewpoints during the process (the fusion step is skipped in the first cycle since the system is still at the initial viewpoint). Then, a grasp is synthesized using the fused data. If the quality of the grasp is good enough (above a predefined grasp quality threshold), then the procedure is terminated and the grasp is executed. If the grasp quality is not sufficient, then the fused data are modeled using a modeling technique and sent to the viewpoint optimization policy obtained by the training procedure. The policy returns a direction to follow in order to lead the robot to a successful grasp. This direction is with respect to the sensor's coordinate frame. The sensor is moved to this direction on the view sphere for a certain distance, called a step in this paper. From this new position (viewpoint) new data are acquired, and the procedure is run again until a successful grasp is obtained.

This scheme can work with many grasp synthesis algorithms in literature; the only prerequisites to apply the scheme is the provision of a quality value and a quality threshold that signifies good grasps. As can be seen in Figure 7.1, the quality value and its threshold are used to terminate the viewpoint optimization process. Moreover, these values are necessary while training the system for obtaining the viewpoint optimization policy. Supplying a grasp quality is a common feature of many grasp synthesis algorithms as they already use a quality measure for optimizing grasping points and/or a grasp pose. Therefore, the quality of the best synthesized grasp can be used as the quality of the viewpoint. With this interpretation, the whole procedure can be seen as two cascaded optimizations: The first optimization stage is the grasp synthesis algorithm which aims to find the

best grasp for given data. The second optimization is the viewpoint optimization that aims to supply better data to the grasp synthesis algorithm.

Even though the details of our implementation will be presented in Section 7.3, we would like to give a brief overview of our implementation here to provide a concrete example for the components of the scheme in Figure 7.1. In our implementation, we used an eye-in-hand system: a manipulator with a depth sensor attached next to its gripper. The sensor supplies point cloud data of the scene. In the data fusion stage, we stitch the point clouds that are acquired during the process. As grasp synthesis algorithm, we used optimization of force closure in 3D. In the modeling stage, we utilized height accumulated features (HAF) to model the stitched point cloud. We add to this model the position of the sensor relative to its initial position and a model of the unexplored regions and form a state of the system. The state is the input of the viewpoint optimization policy that is obtained by a supervised learning procedure. The policy generates a direction that is relative to the depth sensor. We discretize the direction as north, northwest, west, southwest, south, southeast, east and northeast, therefore the policy returns one of these values. The robot moves the depth sensor for a step, so that the direction is followed while staying on the view sphere. From the new viewpoint of the sensor the procedure is run again until a successful grasp is obtained.

The viewpoint optimization policy is at the heart of this process. It is aimed to obtain a policy that can lead to a successful grasp synthesis with the lowest number of iterations (steps) possible. The procedure of obtaining this policy via supervised learning is explained next.

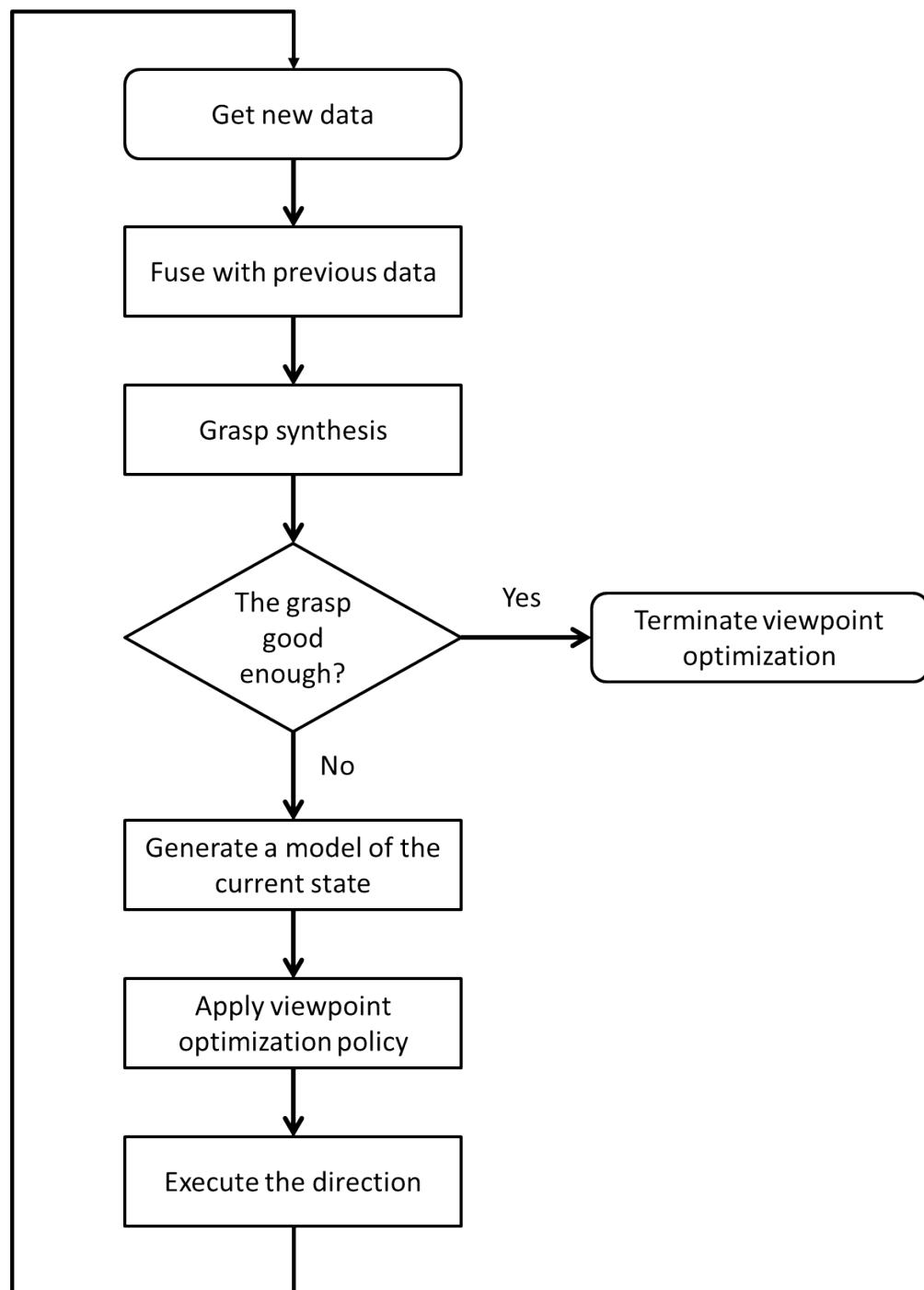
### 7.2.2 The Training Process

The aim of the training is to obtain an exploration policy that supplies an exploration direction to the sensor for the current state (a model of the partial shape data of the target object combined with other state information like the position of the robot, if necessary). For this purpose, we propose the use of supervised learning techniques. This method requires labeled training data which is composed of various states and an exploration direction assigned per se. This exploration direction should make the sensor collect enough information for synthesizing a sufficiently good grasp in the shortest amount of steps. As a result a policy is obtained which returns an exploration direction for a given state.

Formally speaking, the exploration policy is obtained by supervised learning as follows: Define a policy  $\pi$  as a mapping from states to actions (directions). Its state-action value function  $Q^\pi(s, a)$  encodes, for every state  $s$ , the expected value (grasp quality) of taking action  $a$  and following  $\pi$  afterwards. Let  $\pi_0$  be the random policy, taking uniformly distributed random actions independent of the state. Then

$$\pi_1(s) = \arg \max_a Q^{\pi_0}(s, a), \quad (7.1)$$

which is greedy with respect to  $Q^{\pi_0}$ , can be used as a first approximation of



**Figure 7.1:** The global exploration scheme

the optimal policy  $\pi^*$ , which always takes the action that leads to the highest expected value [156]. Because  $s$  is continuous, we need to approximate  $\pi_1$ . We use supervised learning to achieve this. To create a sample set, we drive the system to a series of random states  $s$ , execute every action and record the value of following  $\pi_0$ . Through monte-carlo policy evaluation, we then approximate the mapping

$$s \rightarrow \pi_1(s) \quad (7.2)$$

using a linear classifier. To reduce the dimensionality, principal component analysis and linear discriminant analysis are first applied to the state features.

It is important to note that the exploration policy that is obtained via supervised learning is algorithm dependent; since the best viewpoint can be different for different algorithms, the system should be trained per algorithm. This means training data that is labeled according to the characteristics of the algorithm is necessary. Generating labeled training data manually is a cumbersome and difficult process, since determining the best exploration direction for a given state considering the characteristics of the algorithm may not be straightforward. For this purpose, an automated training data generation procedure that can be implemented in a simulation environment is also proposed in this paper. This procedure is explained next.

### 7.2.3 Automated Training Data Generation Procedure

The automated training data generation procedure aims to generate data for the supervised learning. This data is made up of various states and corresponding exploration directions. Here, we decouple direction determination and data modeling stages; we assign the exploration directions to raw data obtained by the vision sensor, and then generate models using the raw data. By this way, the learning performance with various modelling techniques can be examined without rerunning the data generation procedure.

The automated training data generation procedure can be seen in Algorithm 7.1. The procedure is made up of the following main steps: Finding a random initial sensor position that cannot be grasped immediately, finding the direction that leads to a good grasp with fewest number of steps, assigning this direction to the raw data. The procedure is run for all the objects in the training dataset for the requested data number. The main steps are explained below:

#### Finding a random initial sensor position

In the beginning of the process, first the target object is spawned. Then the robot is moved to a random initial position. Following that, an image is taken in that position and the best grasp for the given data is synthesized. If the quality of this grasp is already above a certain threshold, then the procedure is restarted since viewpoint optimization is not necessary. If the grasp quality is not

sufficient enough, then the image is recorded to the *image\_arr* array. Next, the best exploration direction for that position is found with the following procedure.

### Finding the best exploration direction

The procedure for determining the exploration direction that leads to a grasp with sufficient quality in the fewest number of steps starts at line 13. The first step of this sub-procedure is to lead the robot in a certain exploration direction *k* for a step. If a sufficiently good grasp can already be achieved by this motion, then the grasp quality is recorded, and the step number necessary to achieve this grasp is set to 1. In this case, the rest of the loop (random search for a good grasp) is not necessary for this specific direction (*k*), since a good grasp is already obtained with the fewest number of steps, and the sub-procedure continues for the other directions. If a good grasp cannot be achieved by following this certain direction *k* for a step, a random search is started (line 22) for finding the step number necessary for obtaining a good grasp from this position. The random search is conducted *no\_random\_search* times and each search continues until a good grasp is found or the step number exceeds *max\_search\_step\_no*. Within this random search, the minimum step number that leads to a good grasp and the quality of that grasp is recorded to the corresponding entries of arrays *rec\_step\_no* and *rec\_grasp\_quality* respectively. If a good grasp cannot be achieved by this random search, then the quality of the highest quality grasp is recorded, and as step number, *rec\_step\_no[k]* is assigned to *max\_search\_step\_no*. These steps are applied to all directions.

### Assigning direction to raw data

By the recorded values obtained from the previous sub-procedure, the direction that achieves a good grasp with fewest number of steps is assigned to the recorded image (line 39). If there are directions with the same number of steps, then the one with higher grasp quality is chosen. The outputs of the procedure are the arrays *image\_arr* and *dir\_arr* which are the raw data and assigned exploration directions respectively. For using these data for training, the raw data should be modelled.

Here the variety of the object shapes used in the training is important: By the training, we aim to obtain a policy that can work for objects with different shapes. As the shape variety in the training data set increases, the possibility of getting an efficient viewpoint optimization also increases. The other important point is the modeling technique that is used to model the states. This model should be descriptive enough to represent the state, so that the policy can differentiate between different states and make correct decisions.

An application of the proposed methodology is given in the following section.

---

**Algorithm 7.1** Automated training data generation procedure

---

**Require:**

*no\_of\_objects*: number of object models for training,  
*no\_of\_data\_per\_object*: number of data per object,  
*no\_search\_dir*: number of discrete search directions,  
*no\_random\_search*: number of random search for a direction,  
*max\_search\_step\_no*: maximum number of steps within a random search,  
*quality\_threshold*: threshold for the quality of a sufficiently good grasp.

```

1: initialize dir_arr and image_arr.
2: for i = 1 : no_of_objects do
3:   for j = 1 : no_of_data_per_object do
4:     % Find a random initial sensor position without good grasp
5:     Spawn object model i with random pose.
6:     Move the sensor to a random initial position, continuously fuse data.
7:     Synthesize a grasp and get grasp_quality.
8:     if grasp_quality  $\geq$  quality_threshold then
9:       Decrease j by one, skip the rest and continue the loop.
10:    end if
11:    Record the image to image_arr[j].
12:    % Find the best direction
13:    for k = 1 : no_search_dir do
14:      Move to direction k for a step. Take an image from the current position,
15:      synthesize a grasp and get grasp_quality.
16:      if grasp_quality  $\geq$  quality_threshold then
17:        rec_grasp_quality[k] = grasp_quality
18:        rec_step_no[k] = 1
19:        Skip the rest and continue the loop.
20:      end if
21:      % Find shortest path to grasp from this location
22:      for m = 1 : no_random_search do
23:        for n = 1 : max_search_step_no do
24:          if rec_step_no[k] < n then
25:            Break the loop.
26:          end if
27:          Move to a random direction for a step, take an image from the
28:          current position, synthesize a grasp and get grasp_quality.
29:          if grasp_quality  $\geq$  quality_threshold OR u ==  

max_search_step_no then
30:            rec_grasp_quality[k] = grasp_quality
31:            rec_step_no[k] = u
32:            Break the loop.
33:          end if
34:        end for
35:      end for
36:    end for

```

---

---

```

37: % Update best direction and step limit
38: Initialize min_step_no to max_search_step_no and max_grasp_quality to
0.
39: for k = 1 : no_search_dir do
40:   if (rec_step_no[k] < min_step_no) (rec_step_no[k] == min_step_no
AND rec_grasp_quality[k] > max_grasp_quality) then
41:     min_step_no = rec_step_no[k]
42:     max_grasp_quality = rec_grasp_quality[k]
43:     exp_dir = k
44:   end if
45: end for
46: dir_arr[j] = exp_dir
47: end for
48: end for

```

---

### 7.3 An Implementation of the Viewpoint Optimization Methodology

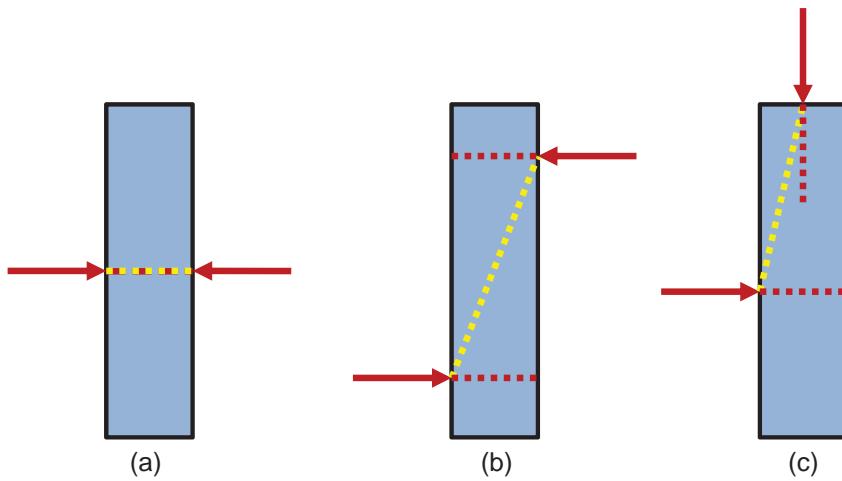
In this section an application of the abovementioned methodology is presented. We designed the method for an eye-in-hand system with a depth sensor that supplies point cloud data. As a grasp synthesis algorithm we used a force closure based method. We chose to use this algorithm since it is simple and its working principles are straightforward. By this, we aim to keep the emphasis on the viewpoint optimization strategy, and make the strategy and its results easier to analyze.

For a representation of a state we used a model of the object point cloud, the position of the robot relative to its initial position and a model of the unexplored regions. For modeling the point clouds we used Height Accumulated Features (HAF).

In this section the grasp synthesis algorithm and the formation of the state are explained in detail.

#### 7.3.1 Grasp Synthesis Algorithm

The stages of the force closure based grasp synthesis strategy are the following: The input to the algorithm is the latest stitched point cloud of the scene. First the points that belong to the object are extracted from the point cloud by removing the major plane (points that belong to the table), clustering the remaining points above the plane and choosing the cluster that belongs to the target object. The point cloud that belongs to the major plane is saved and will be used shortly. Following that, downsampling is applied in order to remove the sensor noise and restrict the amount of data. Surface normals are calculated next from



**Figure 7.2:** An illustration of the gradient angle features that are introduced by [99]. For a given set of two grasping point locations, the gradient angle features can be calculated as angles between force application directions (the red dashed lines) and the line that connects the two grasping points (the yellow dashed line). These two angles are expected to be close to 0 and 180 degrees for a good grasp. These features are simple and efficient to calculate and can distinguish between good grasps (i.e. (a)) and bad grasps (i.e. (b) and (c)).

this downsampled data, and using these normals a force closure based grasp synthesis algorithm is applied. This algorithm is a 3D version of the algorithm used in [24, 99]. This algorithm is efficient to compute and can be calculated as follows: Suppose that there are two grasping point candidates on the object. The quality of the grasp using these points can be calculated by the angles between force application vectors (the normals at those points) and the line that passes through the grasping points (for a 2D visualization see Fig. 7.2). As the angles get closer to 0 and  $\pi$  radians, the quality of the grasp increases. In our application, the highest quality grasp that can be achieved on the object point cloud is found by a brute force search: For each point pair, the force closure based grasp quality measure is calculated. If no point pair can achieve a grasp above a predefined quality, the algorithm fails to synthesize a grasp. If there are point pairs above the threshold, then starting from the pair with the highest quality grasp, collision is checked between the possible poses of the gripper and 1) the object point cloud, 2) the major plane point cloud and 3) a point cloud that represents the unexplored regions (forming the explored regions point cloud is explained in detail in the following subsection). If a pose of the gripper can be found for grasping the object from this point pair without colliding those point clouds, then a successful grasp is found. If not the algorithm fails.

An important feature of this algorithm is that it does not make any assumptions on the missing shape: it only uses the data that is acquired during the process and avoids interacting with the unexplored regions of the scene. If the algorithm was to be used with a single point cloud, it would fail for the majority of the cases. However, thanks to using the vision sensor actively by the proposed methodology, we achieve high success rates.

### 7.3.2 State Modelling

In this application, our state model is composed of three components: a model of the partial object shape data, a model of the unexplored regions and the position of the robot relative to its initial position in the viewpoint optimization process. For detecting the unexplored regions, first we cast regularly distributed virtual points around the object in the beginning of the viewpoint optimization process. We call this set of points “unexplored regions point cloud”. For each point in this point cloud, we check if there are object points in between this virtual point and the sensor’s optical center. If there are no object points in between, then this means that the point is visible to the camera and erased from the virtual point cloud. If there are object points in between, then this means the view of this point is blocked by the object, and therefore the point is not explored yet, thus kept in the point cloud. The point removal procedure is continuously run as the viewpoint of the sensor changes.

For modeling the partial object shape point cloud and unexplored region point cloud, we adopt height accumulated features (HAF) explained in [61] as follows: We defined a five by five height grid at the level of the major plane. The position and the scale of the grid are calculated so that it covers the whole object point cloud with the smallest scale possible. By this way we maintained scale invariance of the model. For modeling the unexplored regions point cloud, a larger scale should be chosen (we used 1.5 times the scale of the object grid) since the backside of the object is also populated by this point cloud.

We conducted simulations to evaluate the performance of the viewpoint optimization strategy with the explained grasp synthesis and state modelling techniques. The results are presented in the next section.

## 7.4 Simulation Results

The application of the viewpoint optimization methodology presented in Section 7.3 is implemented in order to evaluate its success and efficiency. Both for automated training data generation and obtaining simulation results, Robot Operating System (ROS) is used together with the Gazebo simulation environment and the Point Cloud Library PCL.

The system is trained by using two objects, a square prism and a bowl shown in Figure 7.3. For the training, 200 data are collected for the various poses of



**Figure 7.3:** The objects models used in the training: A square prism and a bowl.



**Figure 7.4:** The objects models used in the simulations: A square prism, a rectangular prism, a driller, a glass and a bowl.

the square prism and 50 data are collected for the bowl. For testing the resulting policy, 5 objects are chosen from the household objects database package of ROS which supplies models of real household objects. The shapes and sizes of the square prism-like object and bowl are different from the ones used in the training process. These models can be seen in Figure 7.4. 200 simulations have been conducted for each object with various poses, standing and lying on a table. Only poses that are in principle successful are considered. For example for the rectangular prism, if the object lies on the two long edge, then the gripper opening is not sufficient enough to grasp the object successfully. Therefore, this pose is not used while conducting simulations with this object.

In the simulations, the policy based exploration is compared with a heuristic based search technique and a random search technique. As mentioned previously, the search directions are discretized as north, northwest, west, southwest, south, southeast, east and northeast. The algorithms return one of these values for each

decision and the robot follows this direction with respect to the sensor frame for a step. By the simulations that we led the robot manually by user input, we have observed that even in the most difficult cases, five iterations are sufficient to lead to a successful grasp. Therefore, we set the maximum number of iterations to five steps; if an algorithm can achieve a successful grasp within five steps, then the exploration is considered as successful. Otherwise, it is considered as failed. The heuristic based search always takes the steps west, north, north, east, east in this order. With such a strategy, the object can be explored in west, east and north directions of the initial position of the sensor. In the random search random steps are taken. If any of the algorithms violates the workspace constraints of the robot, then a random step is taken which is forced to be different than the previous decision.

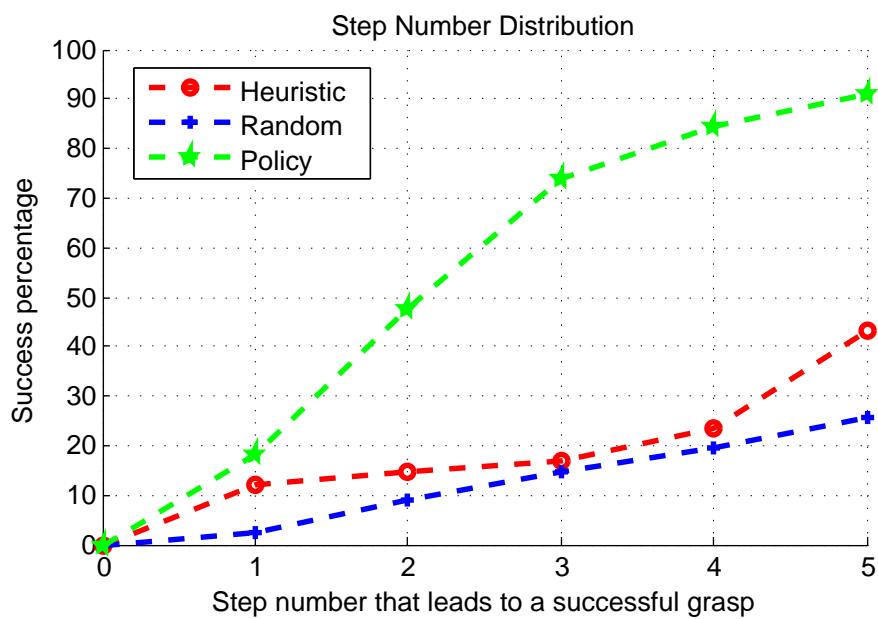
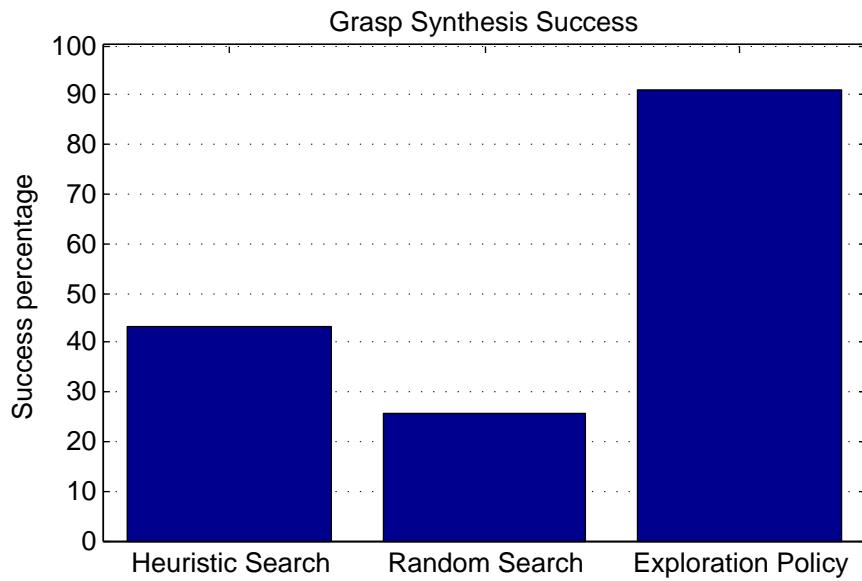
The results can be seen per object in Figures 7.5, 7.6, 7.7, 7.8 and 7.9. These plots show the success percentages of the algorithms and the step number distributions for successful cases. For example, in the bottom plot of Figure 7.5, around 20% success rate is obtained by taking only one step provided by the exploration policy. The success rate increases to around 50% as two steps are taken and so on. As we analyze the results in general we see that the performance of the exploration policy is superior to the heuristic based search and the random search in terms of both success rate and efficiency: For all objects, the policy exploration is more successful and obtains a good grasp within considerably less amount of steps.

Considering each object separately, we see that the policy based exploration brings a huge advantage for the box shaped objects, square prism and rectangular prism. The sharp edges of these objects make the job of the exploration algorithms harder than the smooth surfaces, as the algorithm should make the sensor travel larger distances before a new surface is revealed. The exploration policy successfully does its job for the majority of the cases which brings 92% and 87% success rates for the square prism and the rectangular prism respectively whereas the other algorithms perform below reliable rates. In the cases that the algorithm fails, it is seen that even though the algorithm leads the robot to a good viewpoint in the first three steps, bad decisions are made in the last steps. This can be overcome by adding more training data for the later stages of the process.

For simulations with the driller and the glass, the exploration policy obtain a success rate close to 100%. In this case, the heuristic based search and the random search also bring high success rates. This is due to the relatively smooth surfaces of these objects. This shows that using the vision sensor actively in grasp synthesis process, in some cases by not even using an intelligent algorithm can bring advantage. Nevertheless, the exploration policy achieves sufficiently good grasps in lower number of steps for both objects.

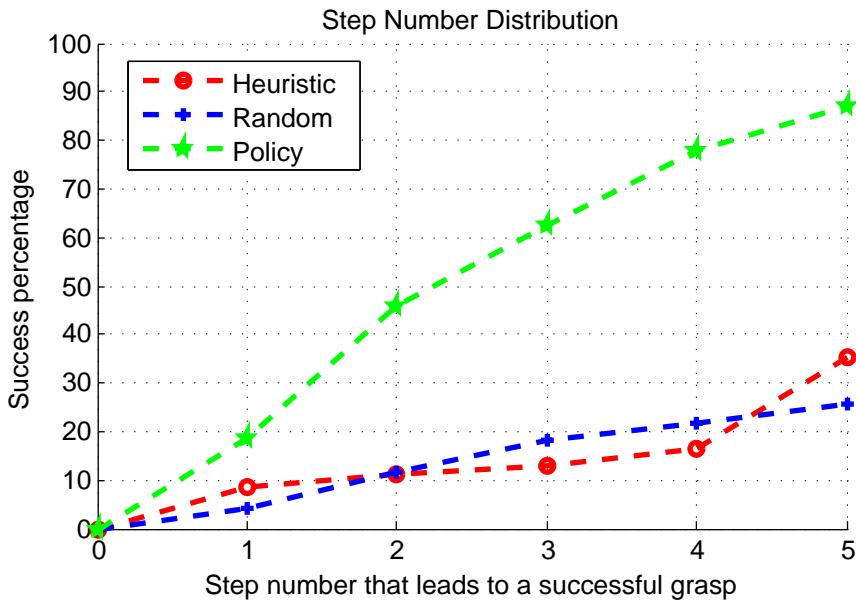
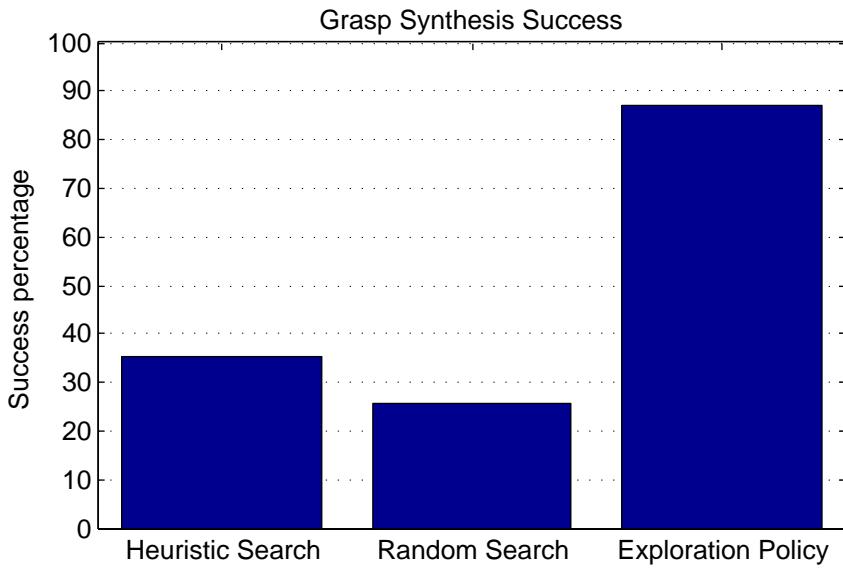
For the bowl object, both the heuristic based search and the exploration policy bring a 100% success rate. However, the exploration policy achieves this rate with one single step whereas the heuristic based exploration takes three. The random search becomes successful in 87% of the cases.

## Experiments with Square Prism



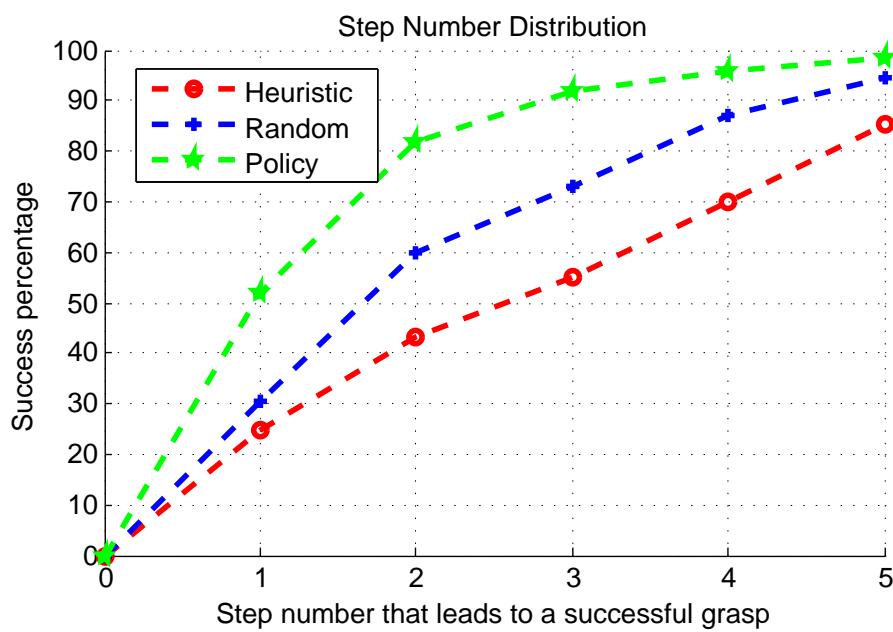
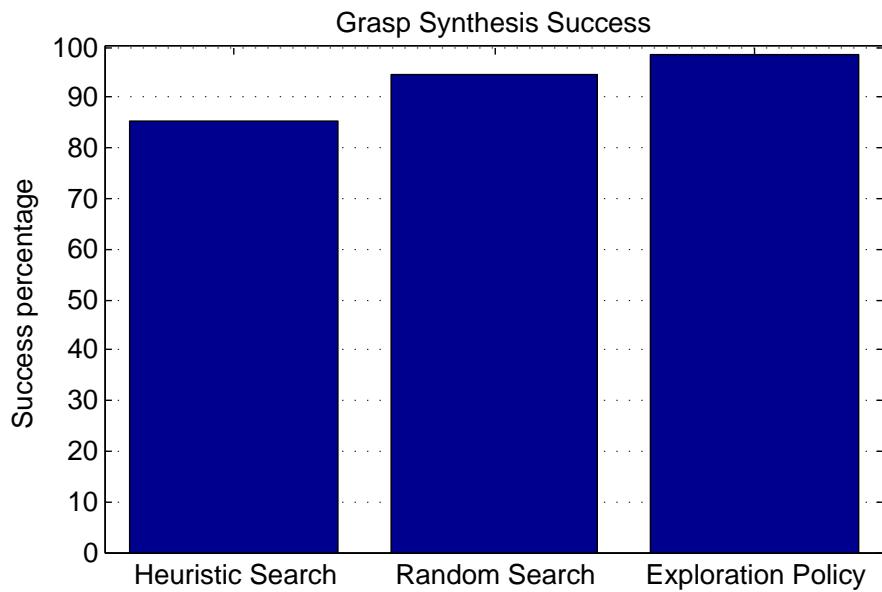
**Figure 7.5:** The results of 200 simulation with each method when using the square prism as a target object.

## Experiments with Rectangular Prism

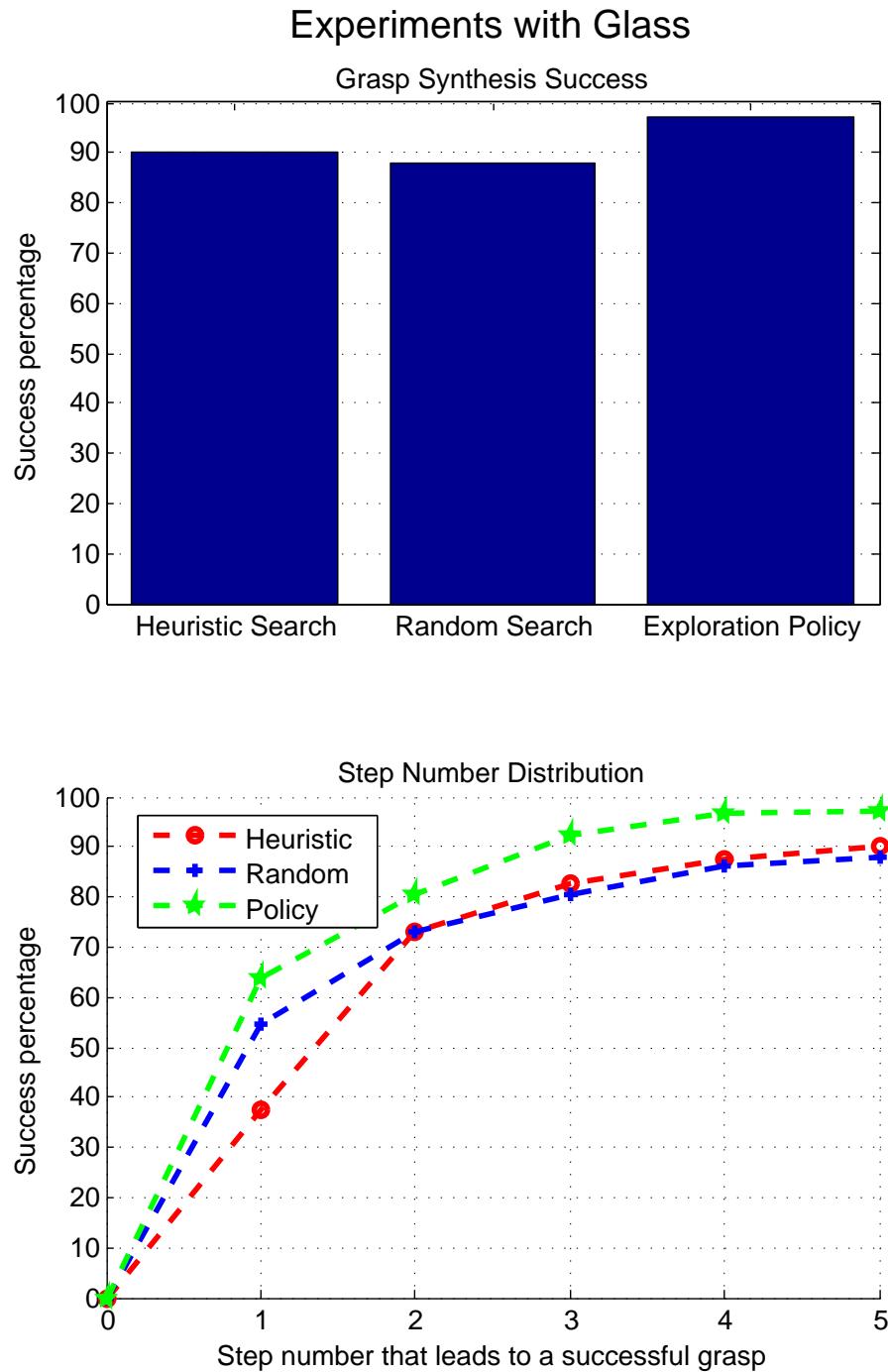


**Figure 7.6:** The results of 200 simulation with each method when using the rectangular prism as a target object.

## Experiments with Driller

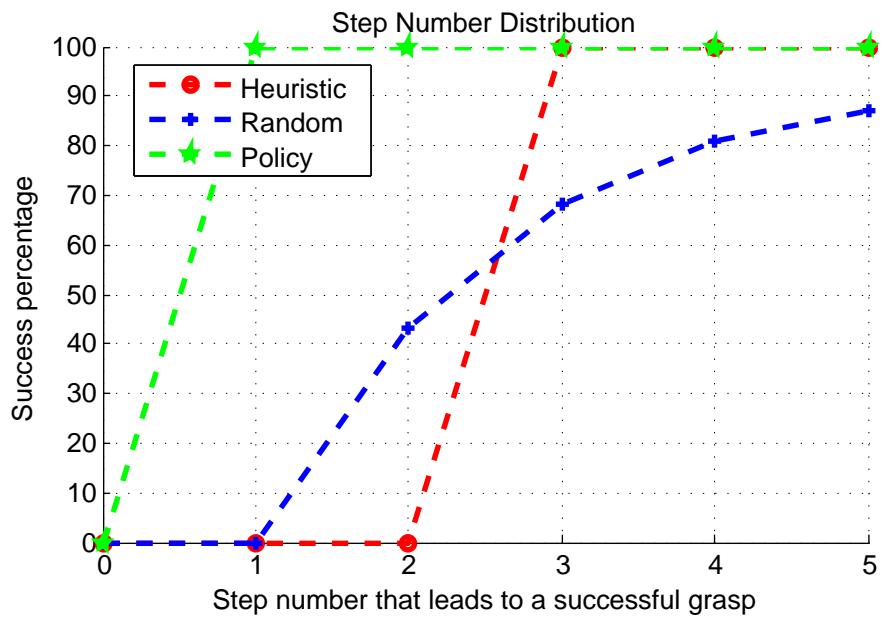
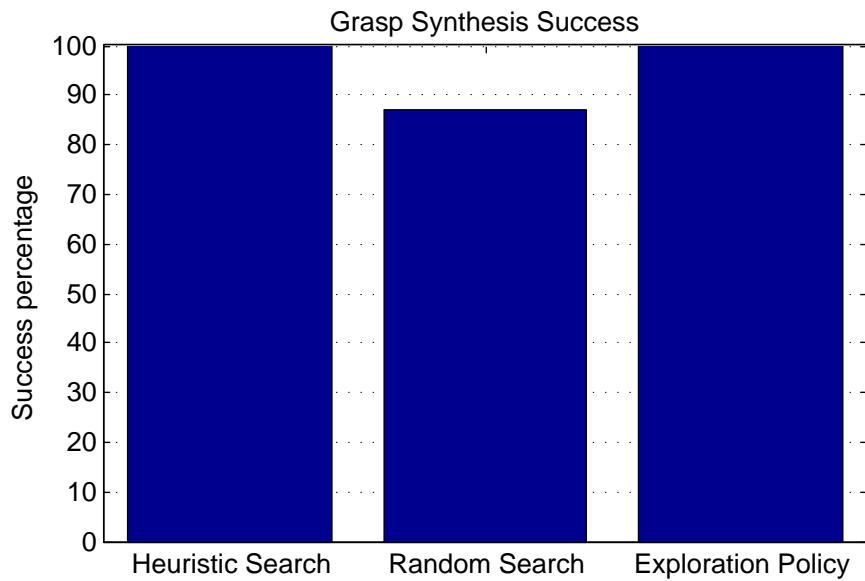


**Figure 7.7:** The results of 200 simulation with each method when using the driller as a target object.



**Figure 7.8:** The results of 200 simulation with each method when using the glass as a target object.

## Experiments with Bowl



**Figure 7.9:** The results of 200 simulation with each method when using the bowl as a target object.

Considering the high performance of the exploration policy, one can conclude that using the proposed state modeling technique and the learning procedure, a good generalization is obtained even by using two objects in the training process. In the final section, the conclusions are summarized and possible improvements are discussed.

## 7.5 Conclusions and Future Work

The novelty of this paper lies in the use of a robot's sensor actively in the grasp synthesis process. By the proposed viewpoint optimization methodology, the performance of grasp synthesis algorithms in literature can be boosted considerably. Since manual training data generation is a cumbersome and difficult process, we also presented an automated training data generation procedure. The application of the methodology to a force closure based grasp synthesis method is given. The simulation results with this method show that the exploration policy obtained via supervised learning is superior to a heuristic based search and a random search in terms of both success rate and efficiency.

In the simulation results, it is shown that even with a limited training set a good generalization can be obtained for objects with various shapes. However, the performance of the algorithm can be improved even further by increasing the number of data and shape variety of the training data set. Also, choosing a more descriptive modelling technique would help to increase the performance of the algorithm, as with HAF models the concavities of the objects cannot be modelled for some poses of the objects.

# Active Object Recognition for Service Robots with Extremum Seeking Control

In this chapter, the proposed active grasp synthesis framework is applied to a similar problem i.e. active object recognition. Similar to the grasp synthesis problem, the performance of a recognition algorithm is dependent to the viewpoint of the sensor, and active object recognition algorithms aim to alter the viewpoint of the sensor in order to increase the recognition rate. The application of the framework proposed in this thesis to the active object recognition problem overcomes disadvantages of the current algorithms in the literature. The performance of the algorithm is evaluated with extensive experiments.

## Abstract

A novel active object recognition strategy is introduced in this chapter. The viewpoint of the camera is continuously optimized in order to increase the success of the recognition system via extremum seeking control. The algorithm does not use any offline data while optimizing the viewpoint. This brings robustness to structured noise (i.e. occlusions, extreme lighting conditions) and imperfections of the training data. Moreover, conducting continuous optimization provides efficiency compared to discrete methods, since all the acquired data is utilized. Experiments with and without occlusions demonstrate the abilities of the proposed method.

## 8.1 Introduction

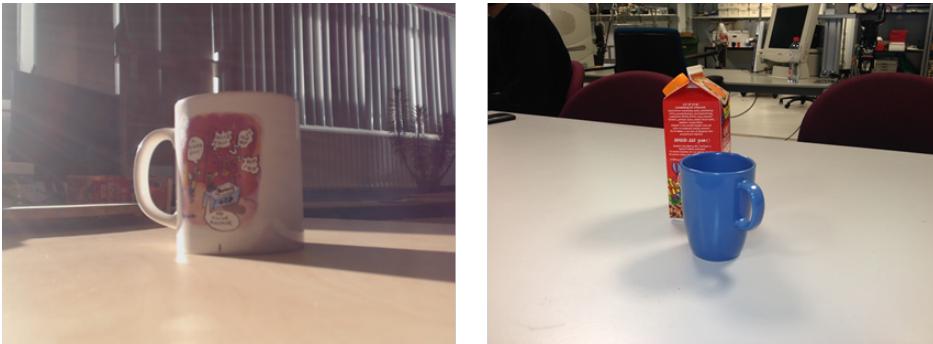
Object recognition is one of the core abilities of service robots. With recognition algorithms, robots are able to locate target objects in a map [53], fetch requested objects for the user [91], use predefined strategies for manipulation [136] or draw some conclusions about the environment of the recognized objects [165]. However, most object recognition algorithms are designed for static systems, and they draw conclusions from a single image of the object. This makes the recognition of a 3D object a difficult problem since different viewpoints generally have different recognition rates; some viewpoint can be much more informative than some other viewpoints about the class of the object [98]. For this reason, the utilization of active methods has been proposed, mostly in the last two decades, and the field of active object recognition has emerged [170]. The algorithms developed in this field enable the system to change the sensor parameters (i.e. viewpoint of the sensor, lighting conditions etc.) in a systematic way in order to increase the success of the recognition algorithm.

The active object recognition problem has different aspects depending on the properties of the system. For viewpoint selection, most systems rely on the change of the camera viewpoint (i.e. [52, 133, 88]), whereas some specialized systems have turntables to change the viewpoint of the target object [57]. Some systems have the ability to manipulate the object, either by pushing [145] or grasping [20]. Some systems are equipped with a light source, so that the lighting conditions can also be controlled [170]. Considering service robots, the most common configuration is having a camera mounted on the head and/or end effector of the robot. Therefore, viewpoint planning of the camera (next best viewpoint problem) is the most important aspect of active object recognition for service robotics.

Unfortunately, current active object recognition algorithms are not yet suitable for service robots simply because they are vulnerable to the effects of structured noise (i.e. occlusions and extreme lighting conditions, see Figure 8.1) which are prevalent in daily environments. This sensitivity is due to the usage of offline data for the viewpoint selection. Moreover, they use discrete search techniques which reduces the efficiency.

In this chapter, we propose a novel active recognition strategy that does not rely on any offline data and enables the system to conduct continuous viewpoint optimization. For the optimization, we propose the usage of extremum seeking control techniques which continuously supply velocity references to the robot in order to optimize the object recognition success. We applied this strategy to a widely used feature based recognition technique, bag of features [35], by coupling it with a neural network extremum seeking control algorithm [161], and show that robust and efficient results can be obtained with such a continuous viewpoint optimization scheme.

The rest of the chapter is organized as follows: We present the related work in the literature in Section 8.2, and discuss the problems of the current methods considering the applications in service robotics. Following that, our extremum



**Figure 8.1:** Images as an example of structured noise which is very common in daily environments. The image at the left is an example of extreme lighting conditions, the image at the right is an example of an occlusion.

seeking control based method is proposed in Section 8.3. Experimental results obtained by using the proposed method are given in Section 8.4. In the last section, the paper is concluded with a discussion on the method and the future work.

## 8.2 Related Work

Various solutions have been proposed for the next best viewpoint selection in the active object recognition literature. A survey on the active recognition field can be found in [50]. Also another survey on active vision systems is presented in [27] which provides recent active object recognition literature. In addition to these surveys, a comparison of active recognition methods is given in [39].

In literature, one of the most common ways of selecting the next best viewpoint is by increasing the mutual information [42, 79, 17]. By this approach, planning the next viewpoint is conducted by searching the whole action space for the maximum object class and observation mutuality, considering the previous actions and observations.

Another common approach is increasing the discriminative information among the class predictions by entropy minimization [97, 181]. In [4], flow images are generated offline in order to provide motion references to the system for minimizing the entropy.

Learning techniques are also used for viewpoint planning in active recognition systems. By these methods, a policy that maps states to actions is learned for increasing the discriminative information. A very common way of learning this policy is via reinforcement learning algorithms [119, 148, 40, 41, 42]. Another method that uses reinforcement learning is presented in [39], in which the policy is learned to optimize the recognition success directly instead of discriminative

information.

In addition to these methods, a method based on feature space trajectory representation is presented in [149]. By this representation, the most discriminative viewpoints are computed in an offline manner. Also, a rule based method is presented in [69], where the recognition system is trained for recognizing Fourier Descriptor based models of the object silhouette. In this method, the next viewpoint is selected based on some heuristics assuming that the object is not occluded.

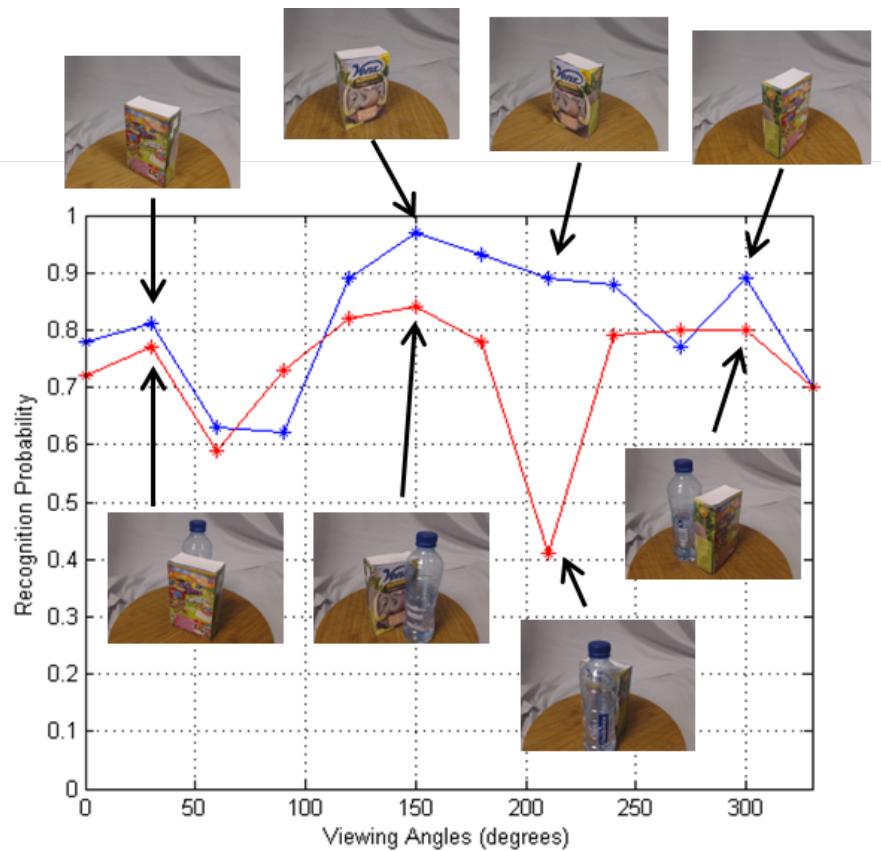
All the algorithms described above suffer from one or more of the following problems:

- The effect of the structured noise.
- Inefficiency caused by discrete search.
- The effect of imperfect training data.

These problems are examined below:

### 8.2.1 The effect of the structured noise

All the above-mentioned active vision methods rely on offline information that is mostly assumed to be obtained in the training phase: In all these cases, it is assumed that the observation probability distribution given the action and current class belief is known. More specifically, if an action, an observation and an object class are denoted by  $a$ ,  $o$  and  $c$ , these algorithms need to know the probability distribution  $P(o|c, a)$  in order to decide on the next action (the way that this decision is made varies depending on the algorithm). The learning based methods and offline methods like [149, 4] do not explicitly need this information in the decision phase, but it is encoded to the policy that is learned during the training process. Knowing  $P(o|c, a)$  implies that, given the class of the object that we are looking for, we need to know what kind of measurement that we are going to have, if we take action  $a$ . This information can be obtained from the position labeled offline training images. However, counting on this observation probability distribution for selecting an action relies on a very strong assumption: The data that is acquired offline should be valid for the current environmental conditions. In other words, if the system does not see what it expects to see in the destination viewpoint, this will affect the belief distribution on object class, and the system will tend to believe that the initial reasoning about the object class and pose was wrong. However, unexpected observations at the destination viewpoint may occur due to the occlusions, extreme lighting conditions or any kind of structured noise. This structured noise is very common in daily environments where the robot is expected to operate, and the presence of these conditions makes the offline data unreliable. As an example, the effect of the occlusion on the object recognition success can be seen in Figure 8.2.



**Figure 8.2:** The effect of occlusion to the object recognition success: Recognition probability of an object is given for various viewing angles as it is the only object in the scene (blue line), and while it was occluded by an object outside the training data set. It can be seen that occlusion affects the recognition rate drastically.

One of the rare active object recognition system that addresses the occlusion problem is presented in [57], where a solution is provided by adding synthetically occluded images of the objects to the training data set. However, such a system can only be robust to the trained occlusions. Moreover, their system includes a turntable and two movable cameras with 90 degrees viewing angle difference, which makes the job of the algorithm much easier compared to our case.

### 8.2.2 Inefficiency caused by discrete search

Another disadvantage of the presented active object recognition methods is that, while moving to the next best viewpoint, the data that can be acquired on the way to the destination is not used. However, utilization of this data can be very useful in reducing the execution time of the algorithms.

Very few algorithms uses the data continuously in active recognition literature. [79, 52] use continuous viewpoint space, but the resultant decisions are discrete, so they do not utilize the data on the way to the destination. [133] uses Fourier descriptors to model the silhouette of the object and recognize the object by the continuous change of the modelled shape. Although successful and efficient results are obtained by this method, the algorithm is vulnerable to acquired shape changes due to occlusions.

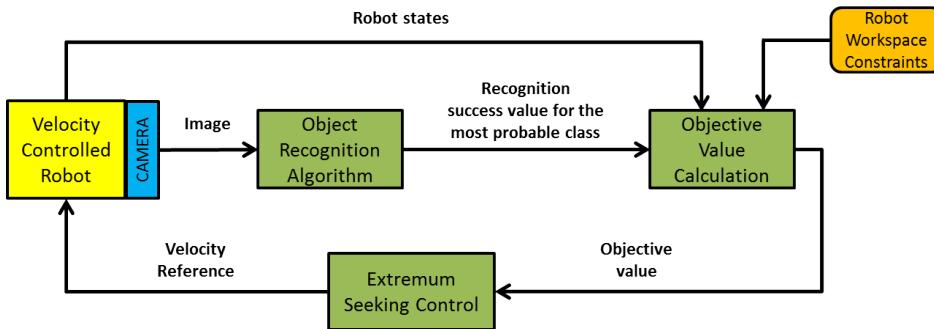
### 8.2.3 The effect of imperfect training data

Using training images for selecting the next best viewpoint may be problematic when the training images are not taken in laboratory conditions. Two common cases are providing object images from the internet search [59, 75] and a robot learning a new object [71]. In both cases, the training images are not taken in ideal conditions. Actually, this problem is analogous to the one explained in 8.2.1, but this time it is caused by the training data. Moreover, in the case of obtaining images from the internet, the relative pose of the camera with respect to the object is unknown. This makes the effect of the action on the current state impossible to calculate, since the current state and states of the other online images cannot be known. As a result, the reasoning process of the above-mentioned object recognition algorithms become inapplicable.

In the next section our method is presented which is not affected by the problems stated in this section.

## 8.3 Active Object Recognition with Extremum Seeking Control

The three problems that are stated in the previous section are mainly due to two reasons: the use of offline training data and inefficiency caused by the discrete search techniques. In this paper, we propose a novel method that does not rely on



**Figure 8.3:** Scheme of the overall system.

any offline data and the system conducts the viewpoint optimization continuously. We achieve this by using an extremum seeking control algorithm that continuously supplies velocity references to the robot by evaluating an objective value that is related to the recognition success. Continuous viewpoint optimization provides efficiency to the system, and not using offline data brings robustness to the structured noise and imperfect training data. In the following subsections we explain the architecture and its modules.

### 8.3.1 System Architecture

The scheme of the overall system is presented in Figure 8.3. As it can be seen from this figure, the optimization loop is made up of three main stages: object recognition, objective value calculation and velocity generation by extremum seeking control. From the camera that is attached to the robot, the object recognition algorithm receives images, runs the classifier and returns a recognition success value for the most probable class. This value is used to calculate an objective value, together with workspace constraints of the robot. The objective value is then utilized by the extremum seeking control algorithm to provide velocity references to the robot. This loop is being run continuously as new recognition success value is returned from the object recognition stage (i.e. 7 Hz in our experiments in Section 8.4). We will now discuss our implementation of each stage.

### 8.3.2 Object Recognition

For object recognition, we use the bag of features method [35]. The features are detected using SURF descriptors. Each object in the training set is trained with a one-vs-all scheme via a support vector machine (SVM) using a radial basis function as kernel. In the classification phase, SVM provides signed distances to the separating plane for each object. These distances are converted to probabilities by fitting sigmoid functions using Platt Scaling [122]. The parameters of

the scaling are obtained by using a separate training image set of each object. The probabilities of each class are then compared to find the most probable class. The next section explains how the objective value of the optimization is formed by using the information provided by the recognition stage.

### 8.3.3 Objective Value Calculation

In this stage, the objective value is calculated which will be optimized by the extremum seeking control in order to increase the success of the recognition algorithm. Our objective value design is made up of recognition success value of the most probable class  $f_o$  and workspace constraints component  $f_w$ :

$$f(x) = f_o + f_w(x, x_c) \quad (8.1)$$

where,  $x$  is the position of the robot and  $x_c$  is the workspace constraints.

For calculating the  $f_o$  component, we directly use the distance to the separating hyperplane of the most probable class instead of using the classification probability values that are obtained by Platt Scaling. There are mainly two reasons for this choice: First, the sigmoids fit by Platt Scaling saturates at the both ends of the curve which affects the optimizers performance; very small changes are observed at the saturated regions which does not provide information that is rich enough for the optimization. Also, in the non-saturated regions, the increase in the recognition value may be too sudden and steep due to the scaling which also affects the performance of the optimizer negatively. The other reason is that, the scaling parameters that are obtained by the fitting process are dependent on the quality of the training image set. On the other hand, by minimizing the distance to the separating hyperplane, we are able to optimize the recognition performance regardless to the scaling quality. This choice is validated by our preliminary experiments.

The workspace constraints component of the robot is imposed to the optimization by the  $f_w$  term in Eq. (8.1). This term is designed as an exponential barrier function [109]:

$$f_w(x, x_c) = \sigma_1 \sum_i^m e^{\sigma_2(x_i - x_{c_i})} \quad (8.2)$$

Here the  $\sigma_1$  and  $\sigma_2$  values are the parameters that are used to weight and shape the barrier function respectively, and  $m$  is the total number of states. This component does not have a significant effect on the objective value if the robot operates within its workspace. As the robot approaches to the boundaries of its workspace, the value of the barrier function increases exponentially which repels the optimizer from the workspace boundaries.

The next section explains the extremum seeking control algorithm used in this paper which utilizes the objective value designed in this section.

### 8.3.4 Extremum Seeking Control

Extremum seeking control (ESC) algorithms are optimizers that do not need the objective function or its gradient to be known; they only utilize the objective value for the optimization. Specifically in robotics, saliency maximization [180] and source seeking [65]. A categorization and comparison of various ESC algorithms can be found in [22].

The advantage of using ESC algorithms in active object recognition problem is not relying on the knowledge of the objective function. This means that we do not need any offline data during the optimization process; we can use only the current recognition success value. This brings robustness to structured noise and imperfect training data.

During optimization, the robot is constrained to move on a view sphere around the object as in [119]. In other words, assuming a camera is attached on the end effector of the robot, the extremum seeking control supplies velocity references in  $x$  and  $y$  directions of the camera frame (as  $z$  axis is orthogonal to the image plane), while the object is always kept at the center of the image by a separate control loop. Therefore, our problem is a two dimensional optimization problem.

By examining the comparison of the ESC methods presented in [22], we chose to use neural network extremum seeking control, since it provides efficient results while being robust to the fluctuations of the objective value. In a nutshell, the neural network extremum seeking control generates its own reference by the sum of two switching functions  $f_p$  and  $f_s$ :

$$f_r = f_p + f_s \quad (8.3)$$

Then, the error is defined as the difference between our objective value  $f(x)$  (Eq. (8.1)) and this reference value.

$$e = f_r - f \quad (8.4)$$

For two dimensional case, this error is minimized by using the following extremum

seeking control rule:

$$u_1 = \begin{cases} -U & e < -\delta_1 \\ U & e > \delta_1 \\ [previous\_state] & \text{otherwise} \end{cases} \quad (8.5)$$

$$u_2 = \begin{cases} 0 & e < -\delta_2 \\ -U & e > \delta_2 \\ [previous\_state] & \text{otherwise} \end{cases} \quad (8.6)$$

$$u_3 = \begin{cases} 0 & e < -\delta_3 \\ 2U & e > \delta_3 \\ [previous\_state] & \text{otherwise} \end{cases} \quad (8.7)$$

$$u = \begin{bmatrix} u_1 + u_2 \\ u_2 + u_3 \end{bmatrix} \quad (8.8)$$

Here,  $\delta_1$ ,  $\delta_2$  and  $\delta_3$  are the hysteresis values, and  $U$  is a constant parameter. This hysteresis mechanism provides robustness to objective value fluctuations, and helps the optimizer escape from the local optimum. The calculated  $u$  vector is directly fed to the robot as velocity reference:

$$\dot{x}_r = u \quad (8.9)$$

For the calculation of the switching functions  $f_p$  and  $f_s$ , and detailed explanation of the neural network extremum seeking control, the reader may refer to [161, 22]. In the next section, the performance of the proposed active object recognition algorithm is analyzed with experiments.

## 8.4 Experimental results

Our experimental setup is made up of a UR5 type Universal Robots arm, a web-camera attached to the end effector of the robot and a rotating platform (Figure 8.4). We have trained the object recognition algorithm with twelve different objects which can be seen in Figure 8.5.

The performance of the algorithm is tested with and without occlusion for each object. For both cases, six experiments are conducted per object with 60 degrees rotation difference (introduced by the rotating platform), 144 experiments in total. The optimization loop is run at 7 Hz. For each run, the initial velocity of the robot is selected randomly (one of the cases of eq. (8.8)), and the neural network ESC is activated. As was explained in Section 8.3.1, the distance to the separating hyperplane of the most probable class is optimized. The most probable class is determined by the probability distributions obtained by the Platt Scaled values. If the most probable class changes during the operation, the optimizer is



**Figure 8.4:** Experimental setup: UR5 type robot arm, webcam attached to the tooltip of the robot and the rotating platform.



**Figure 8.5:** The objects that are used in the experiments.

restarted. In order to avoid rapid switching of the most probable class, a low pass filter is applied to the class probabilities.

An experiment is considered successful, if the algorithm can increase the recognition probability of the correct object up to 95% percent. If this success rate cannot be achieved within 30 seconds, the process is terminated.

**Table 8.1:** Summary of the 144 experiments: Success rate and average execution time of the algorithm with and without occlusion for different termination levels.

		Success	Avg. time (s)
w/o occlu.	Initial view	36.1 %	9.4
	Reached %95 (ESC)	91.6 %	
	Terminated after 30 s (ESC)	95.8%	
w/ occlu.	Initial view	36.1 %	10.4
	Reached %95 (ESC)	83.3 %	
	Terminated after 30 s. (ESC)	97.2 %	

All the experimental results are summarized in Table 8.1. For the experiments without occlusions, the object is correctly recognized directly from the initial view with 95% recognition rate for the 36.1% of the cases. For the other cases, when the proposed active recognition algorithm is run, 95% recognition rate is achieved by changing the viewpoint of the camera by supplying velocity references via ESC algorithm, and the success rate increases to 91.6%. For the 4.2% of the cases, the algorithm fails to reach 95% recognition rate in 30 seconds, and the algorithm is terminated. However, the most probable class is still correct at the end of the process (which sums up to 95.8% success). For the other 4.2% of the cases, the object is guessed wrong, because the highest measured probability during the process belongs to a wrong object.

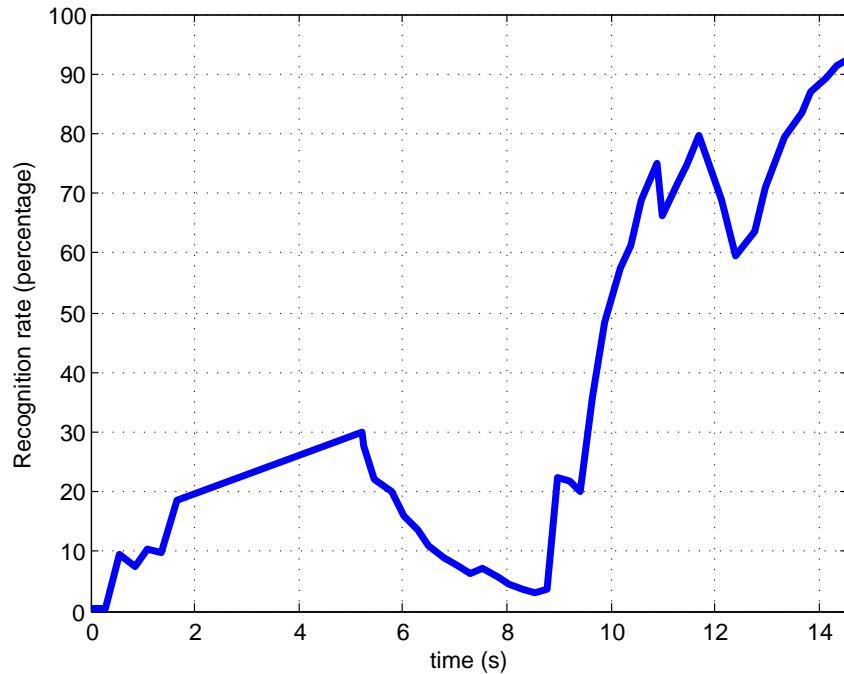
Two samples from the experiments without occlusion are presented in Figures 8.6 and 8.7. In these experiments, the robot starts with a random initial velocity. When the distance to the separation hyperplane is raised above the hysteresis threshold values, the ESC algorithm changes velocity reference and forces the distance to decrease. This causes an increase in the recognition rate of the object and 95% recognition rate is achieved eventually. The average time of obtaining the correct guess is 9.4 seconds.

In order to evaluate the performance of the algorithm under structured noise, occlusion is applied. As the results are compared with the non-occluded case, the significant difference appears to be the decrease while obtaining 95% recognition rate. Although the object is guessed correctly in 97.2% of the cases, the recognition rate cannot be raised to 95% in some runs due to the occluded regions of the object. Also, a slight increase in average convergence time is observed since it takes more time to find a good viewpoint due to occlusions.

Two samples from the experiments with occlusion are given in Figures 8.8 and 8.9. As can be seen from these figures, the algorithm successfully avoid occlusions



(a) Snapshots from an experiment without occlusion. Left image: initial view. Right image: final view.

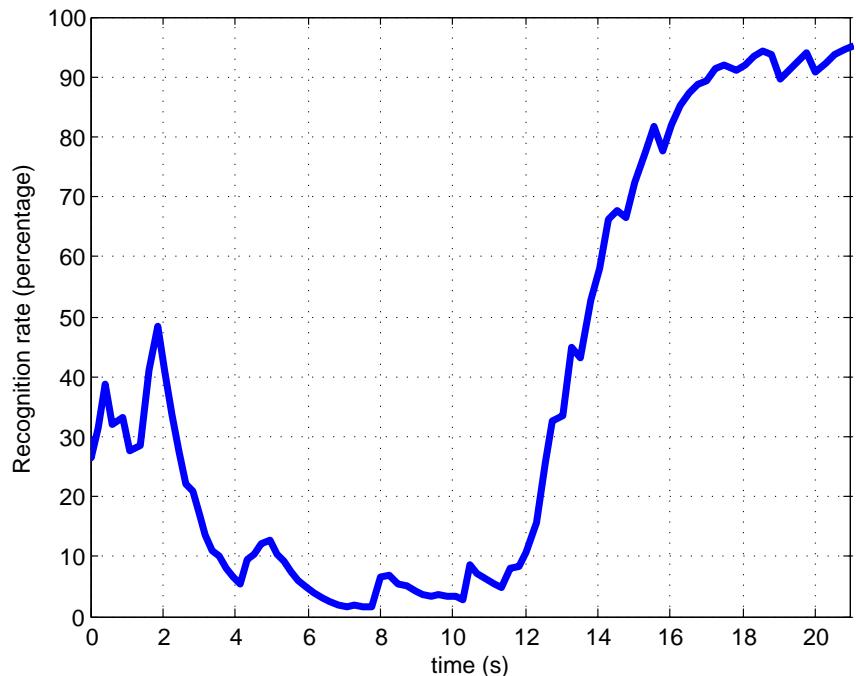


(b) Recognition probability vs. time. The recognition rate increases over time and hits 95%

**Figure 8.6:** Experimental results with “hagel slag” box.



(a) Snapshots from an experiment without occlusion. Left image: initial view. Right image: final view.



(b) Recognition probability vs. time. The recognition rate increases over time and hits 95%.

**Figure 8.7:** Experimental results with “rice dream” box.

and leads the robot to a good viewpoint for recognition. However, the process takes more time and the recognition rate varies more than the non-occluded case, while the algorithm tries to avoid the occluded regions. These large variations are due to the steep scaled values, and might have been a problem for the ESC algorithm, however thanks to the better-behaved hyperplane distance that we used in the optimization process, the algorithm is able to optimize the viewpoint successfully in majority of the cases (as also explained in Section 8.3.3).

For the cases which the algorithm fails to recognize the object, it is observed that the algorithm is stucked in a region where the number of features is low. Due to this low number of features, a healthy guess cannot be maintained by the recognition algorithm, and the optimizer stucks at local minima and fails.

## 8.5 Conclusion

In this chapter a novel active object recognition strategy was presented. This strategy was applied using a feature based object recognition algorithm and a neural network extremum seeking control. The strategy is unique in the sense that it continuously optimizes the viewpoint of the robot without using any offline data. This brings robustness to structured noise effects which are very common in environments where service robots operate. This robustness was validated by the experimental results presented in the paper.

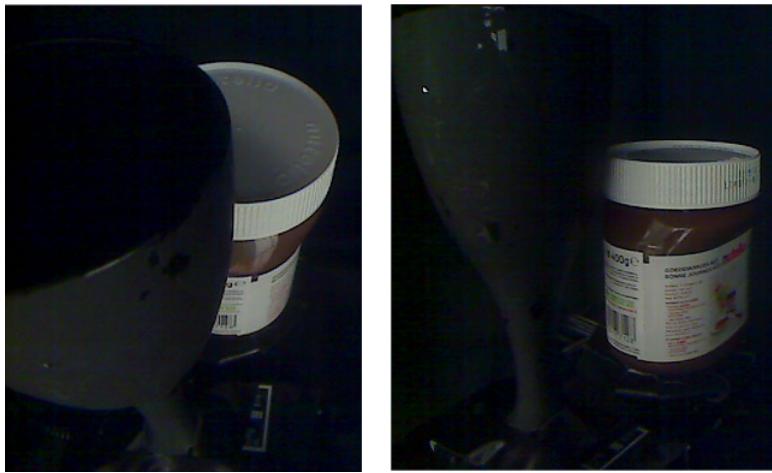
Although this strategy is implemented with a feature based object recognition algorithm, it can also be coupled with other object recognition algorithms. The only requirement is that an objective should be supplied fast enough by the recognition algorithm in order to enable the online process.

It is also important to note that the bottleneck for the convergence speed of the algorithm is the object recognition speed. With a better hardware or a faster recognition algorithm, the optimization loop can be run at faster rates which will directly decrease the convergence time.

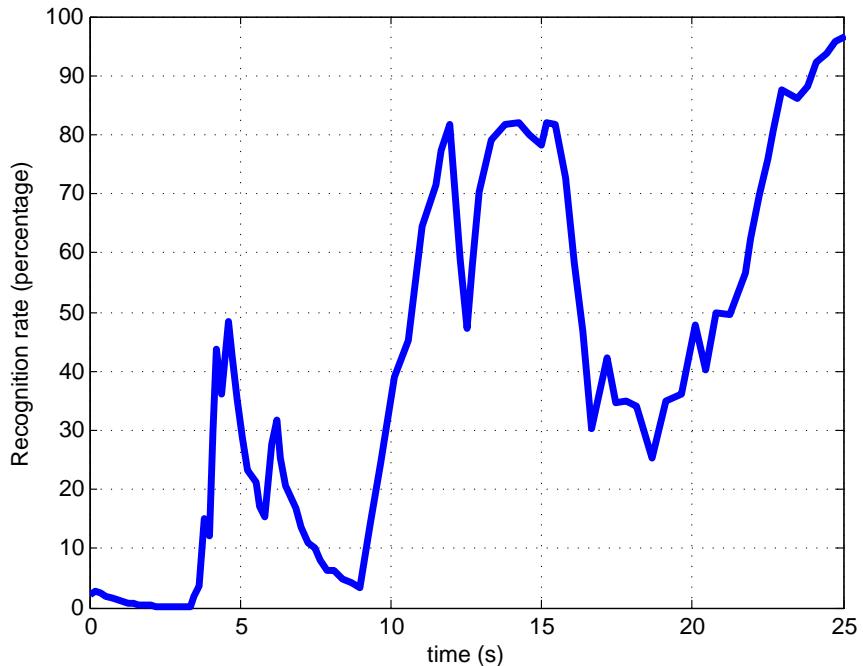
The high robustness and success rate of the proposed algorithm makes it an excellent choice for the robots that operate in unstructured and dynamic environments. We believe that, such an optimization framework can also be quite useful for other problems in robotics such as manipulation and navigation.

## 8.6 Acknowledgements

The authors would like to thank to David Tax from the Pattern Recognition Laboratory, Delft University of Technology, for informative and fruitful discussions on object recognition.



(a) Snapshots from an experiment without occlusion. Left image: initial view. Right image: final view.

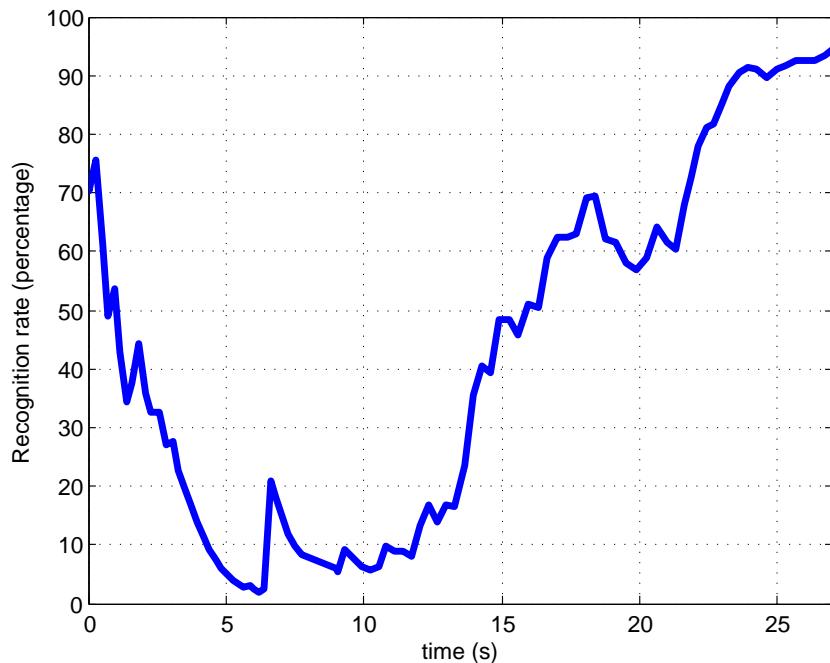


(b) Recognition probability vs. time. The recognition rate increases over time and hits 95%.

**Figure 8.8:** Experimental results with “nutella” jar.



(a) Snapshots from an experiment without occlusion. Left image: initial view. Right image: final view.



(b) Recognition probability vs. time. The recognition rate increases over time and hits 95%.

**Figure 8.9:** Experimental results with “ruijter” box.

# Conclusion

Not having a reliable solution to the problem of grasping unknown objects is one of the major obstacles towards the usage of robots in daily life. The attempts in the literature to solve this problem in a passive manner are successful up to an extent, but fail to supply a holistic solution. These algorithms see grasp synthesis only as a passive data analysis process; they apply strategies for detecting grasping points/regions on a single image of the target object that is supplied to them. On the other hand, the viewpoint of the vision sensor plays a major role for the success of these strategies since it directly determines the missing information about the object's shape. However, no work in the literature deals with changing the viewpoint of the sensor systematically and supplying better data for the detection of grasping points/regions. The active grasp synthesis framework proposed in this thesis combines the data analysis and data improvement steps. By this way, the work load of the data analysis stage is shared, the assumptions that are made in that stage are relaxed, and success of the grasping algorithms is increased.

The proposed active grasp synthesis framework enables strategies to conduct a continuous viewpoint optimization in the grasp synthesis process. The aim of the viewpoint optimization is to improve the quality of the grasp by changing the viewpoint of the sensor. For this optimization, an objective value, in other words a viewpoint quality measure is needed. A natural choice for this value is the quality of the best grasp that can be synthesized from the current viewpoint. By this choice, the grasp synthesis process becomes a two-stage optimization: The passive grasp synthesis algorithm finds the optimum grasp for a given viewpoint, and the viewpoint optimization improves this grasp further. Therefore, the system searches for the best (grasp) of the best (viewpoint).

The presented framework is compatible with almost all the grasp synthesis algorithms in the literature. Any passive grasp synthesis algorithm that can supply a grasp quality value can be integrated to this framework. This is often the

case for algorithms that are not purely based on heuristics, since these algorithms need a quality measure for choosing the best grasp among the grasp candidates. On the other hand, different types of viewpoint quality measures can also be used: The viewpoint optimization can also be conducted to increase the certainty of an object shape hypothesis, to minimize the amount of occlusions on the target object or to optimize any performance related value. If needed, multiple objectives can be combined using weighting functions. This high flexibility brings many possibilities to utilize this framework for manipulation purposes.

Apart from converting a passive grasp synthesis algorithm into an active one, the proposed framework also brings a whole new perspective for designing grasping solutions. Considering the ability of optimizing the viewpoint of the sensor, new grasping strategies can be developed accordingly. For instance, the grasp synthesis algorithm used in Chapter 7 is developed to be used together with a viewpoint optimization strategy. This algorithm is based on surface normals, and it does not make any assumptions for the missing part of the object shape data. Such an algorithm is not suitable to be used in a passive manner, since information obtained from a single viewpoint is not sufficient for it to synthesize successful grasps for the majority of the cases. However, when it is coupled to a viewpoint optimization algorithm, it becomes a very powerful grasp synthesis approach.

The proposed framework is also proven to be useful in other fields that need viewpoint optimization i.e. object recognition. The advantage of using the robot actively is also recognized in that field, and active object recognition algorithms have been developed in the last two decades. These algorithms aim to change the viewpoint of the sensor in order to increase the certainty of the class hypothesis. However, while detecting the next best view, they highly rely on offline data that is acquired in the training phase, and this dependency makes them vulnerable to occlusions and lighting conditions. By using the framework proposed in this thesis, an active object recognition algorithm that does not rely on any offline data for viewpoint optimization was developed. This method used the probability of the most probable class as an objective value, and aimed to maximize it in a continuous procedure. With experiments we showed that such an approach is robust to occlusions and varying lighting conditions, and provides an efficient solution to the problem. This demonstrates the potential of the framework for other disciplines that necessitate viewpoint optimization.

Two types of viewpoint optimization techniques are proposed in this thesis within the presented framework: A local optimization technique and a global optimization technique. For the local technique, the use of continuous extremum seeking control (ESC) algorithms is proposed. These algorithms do not necessitate the knowledge of the objective function or its gradient for optimization purposes. This characteristic makes them suitable for our viewpoint optimization problem, since due to the missing data of the object shape, the objective function or its gradient is not available. In order to choose the most suitable ESC algorithm to our problem, these algorithms are examined with simulations, and their performances under the effect of objective value noise and robot dynamics

are compared. As a result of this analysis, the neural network ESC method is chosen to be used due to its efficiency and robustness to objective value noise. By coupling this method with a 2D grasp synthesis strategy, the abilities of the 2D algorithm is enhanced significantly. This local optimization technique is easy to implement: it does not necessitate any offline training stage, and its parameters are straightforward to tune. However, it can get stuck at local optima, and despite the improvement of the grasping result, the full potential of the grasp synthesis algorithm may not be achieved. These local optima can be escaped by choosing high hysteresis values in the neural network ESC, but this choice degrades the performance of the optimizer.

The viewpoint optimization technique based on supervised learning was designed to avoid the local optima problem. This method requires an offline training stage which aims to obtain a viewpoint optimization policy. This policy maps partial object shape models to exploration directions. To train such a policy supervised learning techniques are used. For this labelled training data are required. These data are composed of models of the partial shape data and an exploration direction which will improve the grasp quality as it is followed by the robot. Since manual labelling is a laborious process, an automated data labelling algorithm is developed. In this algorithm, first partial shape data is acquired by moving the robot randomly. Following that, the robot determines the best exploration direction by exhaustive search. Then, the determined direction is saved together with a model of the partial shape data. The resultant policy determines an exploration direction for a given partial shape model and can be continuously used for viewpoint optimization. This technique is implemented with a point cloud based grasp synthesis algorithm which uses surface normals. In order to use all the acquired data, the point clouds that are acquired during the robot's motion are stitched. For modeling the partial shape data, height accumulated features are used. The labelled data is automatically generated in a simulation environment. The experimental results show that the proposed method is highly efficient for viewpoint optimization purposes and improves the quality of the grasp over time. The ability of the algorithm for escaping local optima is directly related with the amount of training data and the descriptiveness of the object model: The algorithm is guaranteed to find the global optimum for all objects in the training data set. As it is trained with more data and with descriptive models, this ability is extended to other objects.

The efficiencies of the both local and global viewpoint optimization techniques are directly related to the speed of the passive grasp synthesis stage. The framework is designed for continuous viewpoint optimization, and the grasp synthesis algorithm is run in each cycle. The execution times of the viewpoint optimization methods are negligible compared to the grasp synthesis stage, and therefore, the speed of this stage is a bottleneck for the whole procedure. As the grasp synthesis algorithm becomes faster or a computer with higher processing power is used, the robot can be allowed to move faster, and more efficient results can be obtained. Therefore, efficient grasp synthesis methods are more preferable to be used in this

framework rather than computationally expensive approaches.

Apart from the viewpoint optimization framework and methodologies presented within the framework, there are two side products of this thesis. One is the analysis of the extremum seeking control algorithms for robotics applications. These algorithms are especially useful for robots that operate in unstructured environments, since obtaining an objective function for optimization problems in these environments is challenging due to the incomplete environment models. The potential of these algorithm is recognized recently in the robotics community and the analysis results presented in this thesis provide a guideline for choosing the most suitable algorithm for a given the problem. The second side product is the literature survey on grasping algorithms. This survey fills the gap of a recent literature study in the grasping field by providing a comprehensive overview of the grasp synthesis algorithms. In this survey, various grasp synthesis strategies are explained together with their assumptions and related constraints. The limitations based on these assumptions are also discussed.

Grasping unknown objects is a problem that does not have a benchmark that is widely accepted by the community. Each research group that works on this subject uses its own set of objects for experiments. Furthermore, the abilities of the robotic setups vary from group to group, and the grasping scenarios covered by the publications is diverse as well. All these differences between experimental setups make it very challenging to compare the abilities of the proposed algorithms and analyze their performances from the presented results. Moreover, in most publications, the experimental conditions are not fully presented; the information about the relative pose of the target object with respect to the sensor and the positions of the obstacles (if any) are generally missing. As the uncertainty due to the missing information is added to the differences of the experimental setups, detecting the weaknesses of the algorithms and determining the gaps of the field become very troublesome. Nevertheless, all these problems can be solved by a benchmark which have the following information: 1) sets of objects, 2) various (initial) viewpoints to conduct the experiments for each object, 3) various scenarios with different occlusions. The first two properties are essential for a benchmark in grasping unknown objects problem, since they ensure that the algorithms deal with the same amount of information. The last property is important to analyze the abilities of the grasping strategies in challenging conditions.

Operating in dynamic and unstructured environments are challenging for robots and necessitate them to use more of their abilities than in the case of known and static environments: They need to analyze the situation actively and try to gather enough information as efficiently as possible in order to solve a given problem. The grasping problem is no different in that sense, and the proposed viewpoint optimization framework and methodologies fill a crucial gap of the active algorithms in the literature. We believe that the solution of the grasping problem relies on the harmony of the active algorithms working in various levels and stages of the process: viewpoint optimization provides the necessary information for grasp synthesis, the pre-grasp manipulation algorithms are essential to

solve the grasping problem in challenging situations and contact reactive and proximity reactive algorithms have an important role in supporting the grasping process by improving stability of the grasp with tactile or proximity sensors. A robot that is equipped with these active algorithms will have a major advantage of solving the grasping problem even in very challenging situations comparing to the robots that are using passive strategies.

As the grasping problem is solved, another huge obstacle on the way of having service robots in daily life will be removed. The utilization of such a workforce in daily lives will enable many possibilities to solve various problems, and it will be a great progress of the humanity. If this opportunity is used properly, it will be a step towards a utopia. In the end, “progress is the realization of utopias.”





# Bibliography

- [1] Kerncijfers van de bevolkingsprognose, 2010-2060, December 2012.
- [2] A. Albers, S. Brudniok, J. Ottnad, C. Sauter, and K. Sedchaicharn. Upper body of a new humanoid robot - the design of armar iii. In *6th IEEE-RAS International Conference on Humanoid Robots*, pages 308 – 313, 2006.
- [3] K. Scheinberg A.R. Conn and Ph. L. Toint. On the convergence of derivative-free methods for unconstrained optimization. Technical report, IBM T.J. Watson Research C., Ind. Eng. and Operations Res. Dep., Columbia U., Dep. of Math., Facultes Universitaires ND de la Paix, 1996.
- [4] Tal Arbel and Frank P Ferrie. Entropy-based gaze planning. *Image and Vision Computing*, 19(11):779 – 786, 2001.
- [5] K. B. Ariyur and M. Krstic. *Real-Time Optimization by Extremum-Seeking Control*. Wiley, 2003.
- [6] K.B. Ariyur and M. Krstic. Analysis and design of multivariable extremum seeking. In *ACC*, 2002.
- [7] J.W. Li H.G. Cai H. Liu B. Wang, L. Jiang. Grasping unknown objects based on 3d model reconstruction. In *Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 461 – 466, Monterey, CA, July 2005.
- [8] T. Baier-Lowenstein and Jianwei Zhang. Learning to grasp everyday objects using reinforcement-learning with automatic value cut-off. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1551 – 1556, 2007.
- [9] R. Balasubramanian, L. Xu, P. D. Brook, J. R. Smith, and Y. Matsuoka. Physical human interactive guidance: Identifying grasping principles from

- human-planned grasps. *IEEE Transactions on Robotics*, PP(99):1 – 12, 2012.
- [10] R. Balasubramanian, Ling Xu, P.D. Brook, J.R. Smith, and Y. Matsuoka. Human-guided grasp measures improve grasp robustness on physical robot. In *IEEE International Conference on Robotics and Automation*, pages 2294 – 2301, May 2010.
  - [11] A. Bicchi and V. Kumar. Robotic grasping and contact: a review. In *IEEE International Conference on Robotics and Automation, 2000*, volume 1, pages 348 – 353, 2000.
  - [12] J. Bohg, M. Johnson-Roberson, B. Leon, J. Felip, X. Gratal, N. Bergstrom, D. Kragic, and A. Morales. Mind the gap - robotic grasping under incomplete observation. In *IEEE International Conference on Robotics and Automation*, pages 686 – 693, may 2011.
  - [13] Jeannette Bohg and Danica Kragic. Grasping familiar objects using shape context. In *International Conference on Advanced Robotics*, pages 1 – 6, Munich, June 2009.
  - [14] Jeannette Bohg and Danica Kragic. Learning grasping points with shape context. *Robot. Auton. Syst.*, 58(4):362 – 377, Apr. 2010.
  - [15] J. Bohren, R.B. Rusu, E.G. Jones, E. Marder-Eppstein, C. Pantofaru, M. Wise, L. Mosenlechner, Wim Meeussen, and S. Holzer. Towards autonomous robotic butlers: Lessons learned with the pr2. In *IEEE International Conference on Robotics and Automation*, pages 5568 – 5575, 2011.
  - [16] Gary M. Bone, Andrew Lambert, and Mark Edwards. Automated modeling and robotic grasping of unknown three-dimensional objects. In *IEEE International Conference on Robotics and Automation*, pages 292 – 298, May 2008.
  - [17] H. Borotschnig, L. Paletta, M. Prantl, and A. Pinz. Appearance-based active object recognition. *Image and Vision Computing*, 18(9):715 – 727, 2000.
  - [18] Ch. Borst, M. Fischer, and G. Hirzinger. A fast and robust grasp planner for arbitrary 3d objects. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, volume 3, pages 1890 – 1896, May 1999.
  - [19] Ch. Borst, M. Fischer, and G. Hirzinger. Grasp planning: How to choose a suitable task wrench space. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, volume 1, pages 319 – 325, May 2004.

- [20] B. Browatzki, V. Tikhanoff, G. Metta, H. H. Bulthoff, and C. Wallraven. Active object recognition on a humanoid robot. In *IEEE International Conference on Robotics and Automation*, pages 2021 – 2028, 2012.
- [21] Eric Brown, Nicholas Rodenberg, John Amend, Annan Mozeika, Erik Steltz, Mitchell R. Zakin, Hod Lipson, and Heinrich M. Jaeger. Universal robotic gripper based on the jamming of granular material. *Proceedings of the National Academy of Sciences*, 107(44):18809 – 18814, 2010.
- [22] B. Calli, W. Caarls, P. Jonker, and M. Wisse. Comparison of extremum seeking control algorithms for robotic applications. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3195 – 3202, Oct. 2012.
- [23] Berk Calli, Martijn Wisse, and Pieter Jonker. Grasping of unknown objects via curvature maximization using active vision. In *IROS*, 2011.
- [24] Berk Calli, Martijn Wisse, and Pieter Jonker. Grasping unknown objects with active exploration via online viewpoint optimization. 2014.
- [25] S. Caselli, E. Faldella, B. Fringuelli, and F. Zanichelli. A hybrid system for knowledge-based synthesis of robot grasps. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1575 – 1581, 1993.
- [26] I-Ming Chen and J.W. Burdick. Finding antipodal point grasps on irregularly shaped objects. *IEEE Transactions on Robotics and Automation*, 9(4):507 – 512, aug 1993.
- [27] Shengyong Chen, Youfu Li, and Ngai Ming Kwok. Active vision in robotic systems: A survey of recent developments. *The International Journal of Robotics Research*, 30(11):1343 – 1377, 2011.
- [28] J. Cochran and M. Krstic. Nonholonomic source seeking with tuning of angular velocity. *IEEE Transactions on Automatic Control*, 54:717 – 731, 2009.
- [29] Jennie Cochran, Eva Kanso, Scott D. Kelly, Hailong Xiong, and Miroslav Krstic. Source seeking for two nonholonomic models of fish locomotion. *Trans. Rob.*, 25:1166 – 1176, 2009.
- [30] P.I. Corke. A robotics toolbox for matlab. *IEEE Robotics Automation Magazine*, 3:24 – 32, 1996.
- [31] J. Cornella and R. Suarez. On 2d 4-finger frictionless optimal grasps. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 4, pages 3680 – 3685, 2003.

- [32] Noel Curtis and Jing Xiao. Efficient and effective grasping of novel objects through learning and adapting a knowledge base. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2252 – 2257, Nice, September 2008.
- [33] M.R. Cutkosky. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on Robotics and Automation*, 5(3):269 – 279, 1989.
- [34] Barbara Da Roit. Long-term care reforms in the netherlands. In Costanzo Ranci and Emmanuele Pavolini, editors, *Reforms in Long-Term Care Policies in Europe*, pages 97–115. Springer New York, 2013.
- [35] C. Dance, J. Willamowski, L. Fan, C. Bray, and G. Csurka. Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, Prague, 2004.
- [36] Hao Dang, Jonathan Weisz, and P.K. Allen. Blind grasping: Stable robotic grasping using tactile feedback and hand kinematics. In *IEEE International Conference on Robotics and Automation*, pages 5917 – 5922, 2011.
- [37] A.J. Davison and D.W. Murray. Simultaneous localization and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):865 – 880, 2002.
- [38] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052 – 1067, 2007.
- [39] G.C.H.E. de Croon, I.G. Sprinkhuizen-Kuyper, and E.O. Postma. Comparing active vision models. *Image and Vision Computing*, 27(4):374 – 384, 2009.
- [40] J. Defretin, J. Marzat, and H. Piet-Lahanier. Learning viewpoint planning in active recognition on a small sampling budget: a kriging approach. In *Ninth International Conference on Machine Learning and Applications*, pages 169 – 174, 2010.
- [41] F. Deinzer, C. Derichs, H. Niemann, and J. Denzler. A framework for actively selecting viewpoints in object recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(4):765 – 799, 2009.
- [42] J. Denzler and C. M. Brown. Information theoretic sensor data selection for active object recognition and state estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):145 – 157, 2002.
- [43] Renaud Detry, Carl Henrik Ek, Marianna Madry, and Danica Kragic. Learning a dictionary of prototypical grasp-predicting parts from grasping experience. In *IEEE International Conference on Robotics and Automation*, 2013.

- [44] C. Direkoglu and M.S. Nixon. Shape classification using multiscale fourier-based description in 2-d space. In *9th International Conference on Signal Processing*, pages 820 – 823, 2008.
- [45] M.R. Dogar and S.S. Srinivasa. Push-grasping with dexterous hands: Mechanics and a method. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2123–2130, 2010.
- [46] Nesic Dragan. Extremum seeking control: Convergence analysis. *European J. of Cont.*, 15:331 – 347, 2009.
- [47] S. Dragiev, M. Toussaint, and M. Gienger. Gaussian process implicit surfaces for shape estimation and grasping. In *IEEE International Conference on Robotics and Automation*, pages 2845 – 2850, May 2011.
- [48] S. Drakunov, U. Ozguner, P. Dix, and B. Ashrafi. Abs control using optimum search via sliding modes. *Control Systems Technology*, 3:79 – 85, 1995.
- [49] H. Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE Robotics Automation Magazine*, 13(2):99 – 110, 2006.
- [50] S. Dutta Roy, S. Chaudhury, and S. Banerjee. Active recognition through next view planning: a survey. *Pattern Recognition*, 37(3):429 – 446, 2004.
- [51] M. Ehrenmann, R. Zollner, O. Rogalla, and R. Dillmann. Programming service tasks in household environments by human demonstration. In *11th IEEE International Workshop on Robot and Human Interactive Communication*, pages 460–467, 2002.
- [52] R. Eidenberger and J. Scharinger. Active perception and scene modeling by planning with probabilistic 6d object poses. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1036 – 1043, 2010.
- [53] S. Ekvall, P. Jensfelt, and D. Kragic. Integrating active mobile robot object recognition and slam in natural environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5792 – 5797, 2006.
- [54] Staffan Ekvall and Danica Kragic. Learning and evaluation of the approach vector for automatic grasp generation and planning. In *IEEE International Conference on Robotics and Automation*, pages 4715 – 4720, Roma, April 2007.
- [55] S. El-Khoury and A. Sahbani. A new strategy combining empirical and analytical approaches for grasping unknown 3d objects. *Robotics and Autonomous Systems*, 58(5):497 – 507, 2010.

- [56] Yesim H. Esin, Mustafa Unel, and Mehmet Yildiz. Formation control of multiple robots using parametric and implicit representations. In De-Shuang Huang, II Wunsch, DonaldC., DanielS. Levine, and Kang-Hyun Jo, editors, *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, volume 5227 of *Lecture Notes in Computer Science*, pages 558–565. Springer Berlin Heidelberg, 2008.
- [57] F. Farshidi, S. Sorouspour, and T. Kirubarajan. Active multi-camera object recognition in presence of occlusion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2718 – 2723, 2005.
- [58] B. Faverjon and J. Ponce. On computing two-finger force-closure grasps of curved 2d objects. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 424 – 429, Apr. 1991.
- [59] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google’s image search. In *IEEE International Conference on Computer Vision*, volume 2, pages 1816 – 1823, 2005.
- [60] Carlo Ferrari and John Canny. Planning optimal grasps. In *International Conference on Robotics and Automation*, volume 3, pages 2290 – 2295, Nice , France, May 1992.
- [61] D. Fischinger and M. Vincze. Empty the basket - a shape based learning approach for grasping piles of unknown objects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2051 – 2057, 2012.
- [62] S. Frintrop and P. Jensfelt. Attentional landmarks and active gaze control for visual slam. *Robotics, IEEE Transactions on*, 24(5):1054–1065, 2008.
- [63] Hiroshi Fukuda, Naohiro Fukumura, Masazumi Katayama, and Yoji Uno. Relation between object recognition and formation of hand shape: A computational approach to human grasping movements. *Systems and Computers in Japan*, 31(12):11 – 22, 2000.
- [64] L.F. Gallardo and V. Kyrki. Detection of parametrized 3-d primitives from stereo for robotic grasping. In *15th International Conference on Advanced Robotics*, pages 55 – 60, June 2011.
- [65] Nima Ghods. *Extremum Seeking for Mobile Robots*. PhD thesis, University of California, San Diego, 2011.
- [66] C. Goldfeder, P.K. Allen, C. Lackner, and R. Pelossof. Grasp planning via decomposition trees. In *IEEE International Conference on Robotics and Automation*, pages 4679 – 4684, 2007.
- [67] C. Goldfeder, M. Ciocarlie, Hao Dang, and P.K. Allen. The columbia grasp database. In *IEEE International Conference on Robotics and Automation*, pages 1710 – 1716, 2009.

- [68] C. Goldfeder, M. Ciocarlie, J. Peretzman, Hao Dang, and P.K. Allen. Data-driven grasping with partial sensor data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1278 – 1283, 2009.
- [69] E. Gonzalez, A. Adan, V. Feliu, and L. Sanchez. Active object recognition based on fourier descriptors clustering. *Pattern Recognition Letters*, 29(8):1060 – 1071, 2008.
- [70] B. Graf, U. Reiser, M. Hagele, K. Mauz, and P. Klein. Robotic home assistant care-o-bot 3 - product vision and innovation platform. In *IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, pages 139 – 144, 2009.
- [71] A. Haasch, N. Hofemann, J. Fritsch, and G. Sagerer. A multi-modal object attention system for a mobile robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2712 – 2717, Aug. 2005.
- [72] M.R. Hasan, R. Vepa, H. Shaheed, and H. Huijberts. Modelling and control of the barrett hand for grasping. In *15th International Conference on Computer Modelling and Simulation (UKSim)*, pages 230 – 235, April 2013.
- [73] D. Hertog, C. Roos, and T. Terlaky. On the classical logarithmic barrier function method for a class of smooth convex programming problems. *Journal of Optimization Theory and Applications*, 73:1 – 25, 1992.
- [74] A. Herzog, Peter Pastor, Mrinal Kalakrishnan, L. Righetti, T. Asfour, and S. Schaal. Template-based learning of grasp selection. In *IEEE International Conference on Robotics and Automation*, pages 2379–2384, 2012.
- [75] Enrique Hidalgo-Pena, Luis Felipe Marin-Urias, Montes-Gonzalez Fernando, Antonio Marin-Hernandez, and Homero Vladimir Rios-Figueroa. Learning from the web: recognition method based on object appearance from internet images. In *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*, HRI ’13, pages 139 – 140, NJ, USA, 2013.
- [76] K. Hsiao, S. Chitta, M. Ciocarlie, and E.G. Jones. Contact-reactive grasping of objects with partial shape information. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1228 – 1235, Oct. 2010.
- [77] K. Hsiao, Paul Nangeroni, M. Huber, A. Saxena, and A.Y. Ng. Reactive grasping using optical proximity sensors. In *IEEE International Conference on Robotics and Automation*, pages 2098 – 2105, 2009.
- [78] Kaijen Hsiao, L.P. Kaelbling, and T. Lozano-Perez. Grasping pomdps. In *IEEE International Conference on Robotics and Automation*, pages 4685 – 4692, April 2007.

- [79] M. F. Huber, T. Dencker, M. Roschani, and J. Beyerer. Bayesian active object recognition via gaussian process regression. In *Proc. 15th International Information Fusion Conference*, pages 1718 – 1725, 2012.
- [80] Kai Huebner, Kai Welke, Markus Przybylski, Nikolaus Vahrenkamp, Tamim Asfour, Danica Kragic, and Rudiger Dillmann. Grasping known objects with humanoid robots: A box-based approach. In *International Conference on Advanced Robotics*, pages 1 – 6, Munich, June 2009.
- [81] Yan-Bin Jia. Curvature-based computation of antipodal grasps. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1571 – 1577, 2002.
- [82] Yun Jiang, S. Moseson, and A. Saxena. Efficient grasping from rgbd images: Learning using a new rectangle representation. In *IEEE International Conference on Robotics and Automation*, pages 3304 – 3311, May 2011.
- [83] D. Kappler, L. Chang, M. Przybylski, N. Pollard, T. Asfour, and R. Dillmann. Representation of pre-grasp strategies for object manipulation. In *10th IEEE-RAS International Conference on Humanoid Robots*, pages 617 – 624, 2010.
- [84] N.J. Killingsworth, S.M. Aceves, D.L. Flowers, and M. Krstic. Extremum seeking tuning of an experimental hcci engine combustion timing controller. In *American Control Conference*, 2007.
- [85] R. King, R. Becker, G. Feuerbach, L. Henning, R. Petz, W. Nitsche, O. Lemke, and W. Neise. Adaptive flow control using slope seeking. In *Mediterranean Conference on Control and Automation*, 2006.
- [86] E. Klingbeil, D. Rao, B. Carpenter, V. Ganapathi, A.Y. Ng, and O. Khatib. Grasping with application to an autonomous checkout robot. In *IEEE International Conference on Robotics and Automation*, pages 2837 – 2844, May 2011.
- [87] Anna Kochan. Shadow delivers first hand. *Industrial Robot: An International Journal*, 32(1):15 – 16, 2005.
- [88] G. Kootstra, J. Ypma, and B. de Boer. Active exploration and keypoint clustering for object recognition. In *IEEE International Conference on Robotics and Automation*, pages 1005 – 1010, May 2008.
- [89] S. K. Korovin and V. I. Utkin. Use of the slip mode in problems of static optimization. *Automatic and Remote Control*, pages 50 – 60, 1972.
- [90] S.K. Korovin and V.I. Utkin. Using sliding modes in static optimization and nonlinear programming. *Automatica*, 10:525 – 532, 1974.

- [91] Danica Kragic, Marten Bjorkman, Henrik I. Christensen, and Jan-Olof Eklundh. Vision for robotic object manipulation in domestic settings. *Robotics and Autonomous Systems*, 52(1):85 – 100, 2005.
- [92] Gert A. Kragten. *UNDERACTUATED HANDS Fundamentals, Performance Analysis and Design*. PhD thesis, Delft University of Technology, 2011.
- [93] O. B. Kroemer, R. Detry, J. Piater, and J. Peters. Combining active learning and reactive control for robot grasping. *Robot. Auton. Syst.*, 58(9):1105 – 1116, Sep. 2010.
- [94] Miroslav Krstic. Performance improvement and limitations in extremum seeking control. *Systems and Control Letters*, 39:313 – 326, 2000.
- [95] Miroslav Krstic and Hsin-Hsiung Wang. Stability of extremum seeking feedback for general nonlinear dynamic systems. *Automatica*, 36:595 – 601, 2000.
- [96] Frank P Kuhl and Charles R Giardina. Elliptic fourier features of a closed contour. *Computer Graphics and Image Processing*, 18(3):236 – 258, 1982.
- [97] C. Laporte, R. Brooks, and T. Arbel. A fast discriminant approach to active object recognition and pose estimation. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 3, pages 91 – 94, Aug. 2004.
- [98] Quoc V. Le, Ashutosh Saxena, and Andrew Y. Ng. Active perception: Interactive manipulation for improving object detection. Technical report, Computer Science Department Stanford University, Computer Science Department Cornell University, 2010.
- [99] Q.V. Le, D. Kamm, A.F. Kara, and A.Y. Ng. Learning to grasp objects with multiple contact points. In *IEEE International Conference on Robotics and Automation*, pages 5062 – 5069, May 2010.
- [100] Hyun-Ki Lee, Myun-Hee Kim, and Sang-Ryong Lee. 3d optimal determination of grasping points with whole geometrical modeling for unknown objects. *Sensors and Actuators A: Physical*, 107-2:146 – 151, 2003.
- [101] Beatriz Leon, Stefan Ulbrich, Rosen Diankov, Gustavo Puche, Markus Przybylski, Antonio Morales, Tamim Asfour, Sami Moisio, Jeannette Bohg, James Kuffner, and Rudiger Dillmann. Opengrasp: A toolkit for robot grasping simulation. In Noriaki Ando, Stephen Balakirsky, Thomas Hemker, Monica Reggiani, and Oskar Stryk, editors, *Simulation, Modeling, and Programming for Autonomous Robots*, volume 6472 of *Lecture Notes in Computer Science*, pages 109–120. Springer Berlin Heidelberg, 2010.

- [102] C. Leung, Shoudong Huang, and G. Dissanayake. Active slam using model predictive control and attractor based exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026 – 5031, 2006.
- [103] Yi Li, Jean-Philippe Saut, Julien Pette, Anis Sahbani, Philippe Bidaud, and Franck Multon. Fast grasp planning by using cord geometry to find grasping points. In *IEEE International Conference on Robotics and Automation*, 2013.
- [104] V. Lippiello, F. Ruggiero, B. Siciliano, and L. Villani. Preshaped visual grasp of unknown objects with a multi-fingered hand. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5894 – 5899, Oct. 2010.
- [105] Efrain Lopez-Damian, Daniel Sidobre, and Rachid Alami. A grasp planner based on inertial properties. In *IEEE International Conference on Robotics and Automation*, pages 754 – 759, April 2005.
- [106] Andrew Miller and Peter K. Allen. Graspit!: A versatile simulator for robotic grasping. *IEEE Robotics and Automation Magazine*, 11(4):110 – 122, Dec. 2004.
- [107] Andrew T. Miller, Steffen Knoop, Henrik I. Christensen, and Peter K. Allen. Automatic grasp planning using shape primitives. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1824 – 1829, September 2003.
- [108] Brian Mirtich and John Canny. Easily computable optimum grasps in 2-d and 3-d. In *In IEEE International Conference on Robotics and Automation*, volume 1, pages 739 – 747, 1994.
- [109] Parwadi Moengin. Exponential barrier method in solving linear programming problems. *International Journal of Engineering & Technology*, 12(3):33 – 37, 2012.
- [110] Antonio Morales, Gabriel Recatala, Pedro J. Sanz, and Angel P. Del Pobil. Heuristic vision-based computation of planar antipodal grasps on unknown objects. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 583 – 588, Korea, May 2001.
- [111] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7:308 – 313, 1965.
- [112] D. Nesic, Ying Tan, and I. Mareels. On the choice of dither in extremum seeking systems: a case study. In *IEEE Conference on Decision and Control*, 2006.

- [113] V.-D. Nguyen. Constructing force-closure grasps. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 1368 – 1373, 1986.
- [114] Van-Due Nguyen. Constructing stable grasps. *The International Journal of Robotics Research*, 8(1):26 – 37, February 1989.
- [115] N. Niparnan and A. Sudsang. Fast computation of 4-fingered force-closure grasps from surface points. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 4, pages 3692 – 3697, 2004.
- [116] Nusawardhana and S.H. Zak. Extremum seeking using analog nonderivative optimizers. In *ACC*, 2003.
- [117] Nusawardhana and S.H. Zak. Simultaneous perturbation extremum seeking method for dynamic optimization problems. In *Americal Control Conference*, 2004.
- [118] C. Olalla, M.I. Arteaga, R. Leyva, and A. El Aroudi. Analysis and comparison of extremum seeking control techniques. In *ISIE*, 2007.
- [119] L. Paletta and A. Pinz. Active object recognition by view integration and reinforcement learning. *Robotics and Autonomous Systems*, 31(1):71 – 86, 2000.
- [120] R. Pelossof, A. Miller, P. Allen, and T. Jebara. An svm learning approach to robotic grasping. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 3512 – 3518, 2004.
- [121] Doug Perrin, Osama Masoud, Christopher E. Smith, and Nikolaos P. Papanikolopoulos. Unknown object grasping using statistical pressure models. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1054 – 1059, 2000.
- [122] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in large margin classifiers*, pages 61 – 74. MIT Press, 1999.
- [123] J. Ponce and B. Faverjon. On computing three-finger force-closure grasps of polygonal objects. In *Fifth International Conference on Advanced Robotics*, volume 2, pages 1018 – 1023, 1991.
- [124] Mila Popovic, Gert Kootstra, Jimmy Alison Jorgensen, Danica Kragic, and Norbert Kruger. Grasping unknown objects using an early cognitive vision system for general scene understanding. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 987 – 994, Sept. 2011.

- [125] Mila Popovic, Dirk Kraft, Leon Bodenhagen, Emre Bäseski, Nicolas Pugeault, Danica Kragic, Tamim Asfour, and Norbert Krüger. A strategy for grasping unknown objects based on co-planarity and colour information. *Robotics and Autonomous Systems*, 58(5):551 – 565, 2010.
- [126] Markus Przybylski, Tamim Asfour, and Rudiger Dillmann. Planning grasps for robotic hands using a novel object representation based on the medial axis transform. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1781 – 1788, Sept. 2011.
- [127] D. Rao, Q.V. Le, T. Phoka, M. Quigley, A. Sudsang, and A.Y. Ng. Grasping novel objects with depth segmentation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2578 – 2585, 2010.
- [128] Gabriel Recatala, Eris Chinellato, Angel delPobil, Youcef Mezouar, and Philippe Martinet. Biologically-inspired 3d grasp synthesis based on visual exploration. *Autonomous Robots*, 25:59 – 70, 2008.
- [129] Mario Richtsfeld and Markus Vincze. Grasping of unknown objects from a table top. In *Workshop on Vision in Action: Efficient Strategies for Cognitive Agents in Complex Environments*, Marseille, 2008.
- [130] Mario Richtsfeld and Michael Zillich. Grasping unknown objects based on 21/2d range data. In *4th IEEE Conference on Automation Science and Engineering*, pages 691 – 696, Arlington, VA, 2008.
- [131] E. Rimon and J.W. Burdick. Mobility of bodies in contact. i. a 2nd-order mobility index for multiple-finger grasps. *IEEE Transactions on Robotics and Automation*, 14(5):696 – 708, 1998.
- [132] E. Rimon and J.W. Burdick. Mobility of bodies in contact. ii. how forces are generated by curvature effects. *IEEE Transactions on Robotics and Automation*, 14(5):709 – 717, oct 1998.
- [133] P. Robbel and D. Roy. Exploiting feature dynamics for active object recognition. In *11th International Conference on Control, Automation, Robotics and Vision*, pages 2102 – 2108, 2010.
- [134] M.A. Rotea. Analysis of multivariable extremum seeking algorithms. In *American Control Conference*, 2000.
- [135] Maja Rudinac, Berk Calli, and Pieter Jonker. Item recognition, learning, and manipulation in a warehouse input station. In Roelof Hamberg and Jacques Verriet, editors, *Automation in Warehouse Development*, pages 133–152. Springer London, 2012.

- [136] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz. Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems Journal*, (Special Issue on Semantic Knowledge), 2008.
- [137] Marcos Salganicoff, Lyle H. Ungar, and Ruzena Bajcsy. Active learning for vision-based robot grasping. *Machine Learning*, 23(2-3):251 – 278, 1996.
- [138] P.J. Sanz, A.P. Del Pobil, J.M. Inesta, and G. Recatala. Vision-guided grasping of unknown objects for service robots. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 3018 –3025, May 1998.
- [139] M. Sarfraz, Mehmood ul Hassan, and M. Iqbal. Object recognition using fourier descriptors and genetic algorithm. In *International Conference of Soft Computing and Pattern Recognition*, pages 318 – 323, 2009.
- [140] Ashutosh Saxena, Justin Driemeyer, Justin Kearns, Chioma Osondu, and Andrew Y. Ng. Learning to grasp novel objects using vision. In *In 10th International Symposium of Experimental Robotics*, 2006.
- [141] Ashutosh Saxena, Justin Driemeyer, Justin Kearns, Chioma Osondu, and Andrew Y. Ng. Learning to grasp novel objects using vision. In *In 10th International Symposium of Experimental Robotics*, 2006.
- [142] Ashutosh Saxena, Justin Driemeyer, and Andrew Y. Ng. Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, 27:157 – 173, 2008.
- [143] Ashutosh Saxena, Lawson L. S. Wong, and Andrew Y. Ng. Learning grasp strategies with partial shape information. In *Proceedings of the 23rd National Conference on Artificial Intelligence*, volume 3, pages 1491 – 1494. AAAI Press, 2008.
- [144] D. Schiebener, J. Schill, and T. Asfour. Discovery, segmentation and reactive grasping of unknown objects. In *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2012.
- [145] D. Schiebener, A. Ude, J. Morimoto, T. Asfour, and R. Dillmann. Segmentation and learning of unknown objects through physical interaction. In *11th IEEE-RAS International Conference on Humanoid Robots*, pages 500 – 506, Oct. 2011.
- [146] Alexander M. Schmidts, Dongheui Lee, and Angelika Peer. Imitation learning of human grasping skills from motion and force data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1002 – 1007, Sept. 2011.

- [147] E. Schut, S. Sorbe, and J. Hoj. Health care reform and long-term care in the netherlands. Technical report, OECD Economics Department, 2013.
- [148] Katsunari Shibata, Tetsuo Nishino, and Yoichi Okabe. Active perception and recognition learning system based on actor-q architecture. *Systems and Computers in Japan*, 33(14):12 – 22, 2002.
- [149] M.A. Sipe and D. Casasent. Feature space trajectory methods for active computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1634 – 1643, Dec. 2002.
- [150] D. Song, K. Huebner, V. Kyrki, and D. Kragic. Learning task constraints for robot grasping using graphical models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1579 – 1585, Oct. 2010.
- [151] D. Stampfer, M. Lutz, and C. Schlegel. Information driven sensor placement for robust active object recognition based on multiple views. In *IEEE Int Technologies for Practical Robot Applications Conference*, pages 133 – 138, 2012.
- [152] Milos S. Stankovic and Dusan M. Stipanovic. Extremum seeking under stochastic noise and applications to mobile sensors. *Automatica*, 46:1243 – 1251, 2010.
- [153] Jan Sternby. Adaptive control of extremum systems. In *Methods and Applications in Adaptive Control*. Springer Berlin / Heidelberg, 1980.
- [154] R. Suarez and P. Grosch. Mechanical hand ma-i as experimental system for grasping and manipulation. In *IEEE International Conference on Robotics and Automation Video*, 2005.
- [155] R. Suarez, M. Roa, and J. Cornellà. Grasp quality measures. Technical Report IOC-DT-P-2006-10, Universitat Politecnica de Catalunya, Robotica Industrial i de Sistemes, March 2006.
- [156] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning, An introduction*. The MIT Press, Cambridge, MA, USA, 1st edition, 1998.
- [157] Y. Tan, W.H. Moase, C. Manzie, D. Nesic, and I.M.Y. Mareels. Extremum seeking from 1922 to 2010. In *Chinese Control Conference*, 2010.
- [158] Geoffrey Taylor, , Geoffrey Taylor, and Lindsay Kleeman. Grasping unknown objects with a humanoid robot. In *Australasian conference on Robotics and Automation*, pages 191 – 196, 2002.
- [159] M. Taylor, A. Blake, and A. Cox. Visually guided grasping in 3d. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 761 – 766, May 1994.

- [160] M.C.M. Teixeira and S.H. Zak. Analog nonderivative optimizers. In *American Control Conference*, 1997.
- [161] M.C.M. Teixeira and S.H. Zak. Analog neural nonderivative optimizers. *IEEE Transactions on Neural Networks*, 9(4):629 – 638, Jul 1998.
- [162] Sebastian Thrun. Simultaneous localization and mapping. In Margaret E. Jefferies and Wai-Kiang Yeap, editors, *Robotics and Cognitive Approaches to Spatial Mapping*, volume 38 of *Springer Tracts in Advanced Robotics*, pages 13 – 41. Springer Berlin Heidelberg, 2008.
- [163] Sebastian Thrun and Michael Montemerlo. The graph slam algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research*, 25(5-6):403 – 429, 2006.
- [164] W. van der Windt, R.C.K.H. Smeets, and E.J.E. Arnold. De arbeidsmarkt van verpleegkundigen, verzorgenden en sociaalagogen 2008-2012. *Regio-Marge 2008*, 2008.
- [165] Shrihari Vasudevan, Stefan Gachter, Viet Nguyen, and Roland Siegwart. Cognitive maps for mobile robots—an object based approach. *Robotics and Autonomous Systems*, 55(5):359 – 371, 2007.
- [166] T. Vidal-Calleja, A.J. Davison, J. Andrade-Cetto, and D.W. Murray. Active control for single camera slam. In *IEEE International Conference on Robotics and Automation*, pages 1930 – 1936, 2006.
- [167] G.C. Walsh. On the application of multi-parameter extremum seeking control. In *American Control Conference*, 2000.
- [168] B. Wang, L. Jiang, J.W. Li, H.G. Cai, and H. Liu. Grasping unknown objects based on 3d model reconstruction. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 461 – 466, Monterey, CA, July 2005.
- [169] Hsin-Hsiung Wang, Miroslav Krstic, and Georges Bastin. Optimizing bioreactors by extremum seeking. *International Journal of Adaptive Control and Signal Processing*, 13:651 – 669, 1999.
- [170] D. Wilkes and J.K. Tsotsos. Active object recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 136 – 141, Jun 1992.
- [171] B. Wittenmark and R.J. Evans. Extremal control of wiener model processes. In *IEEE Conference on Decision and Control*, 2002.
- [172] B. Wittenmark and A. Urquhart. Adaptive extremal control. In *IEEE Conference on Decision and Control*, 1995.

- [173] Kimitoshi Yamazaki, Masahiro Tomono, and Takashi Tsubouchi. Picking up an unknown object through autonomous modeling and grasp planning by a mobile manipulator. In Christian Laugier and Roland Siegwart, editors, *Field and Service Robotics*, volume 42, pages 563 – 571. Springer Berlin / Heidelberg, 2008.
- [174] Kimitoshi Yamazaki, Masahiro Tomono, Takashi Tsubouchi, and Shinichi Yuta. A grasp planning for picking up an unknown object for a mobile manipulator. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2143 – 2149, May 2005.
- [175] Kimitoshi Yamazaki, Masahiro Tomono, Takashi Tsubouchi, and Shinichi Yuta. A grasp planning for picking up an unknown object for a mobile manipulator. In *IEEE International Conference on Robotics and Automation*, pages 2143 – 2149, May 2005. Intelligent Robot Laboratory, University of Tsukuba.
- [176] H. Yu and U. Ozguner. Extremum-seeking control strategy for abs system with time delay. In *American Control Conference*, 2002.
- [177] Chunlei Zhang, Daniel Arnold, Nima Ghods, Antranik Siranosian, and Miroslav Krstic. Source seeking with non-holonomic unicycle without position measurement and with tuning of forward velocity. *Systems and Control Letters*, 56:245 – 252, 2007.
- [178] Chunlei Zhang and R. Ordonez. Non-gradient extremum seeking control of feedback linearizable systems with application to abs design. In *Conference on Decision and Control*, 2006.
- [179] Chunlei Zhang and Raul Ordonez. Robust and adaptive design of numerical optimization-based extremum seeking control. *Automatica*, 45:634 – 646, 2009.
- [180] Yinghua Zhang, Jinglin Shen, Mario Rotea, and Nicholas Gans. Robots looking for interesting things: Extremum seeking control on saliency maps. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1180 – 1186, Sept. 2011.
- [181] Xiang Sean Zhou, D. Comaniciu, and A. Krishnan. Conditional feature sensitivity: a unifying view on active recognition and feature selection. In *Ninth IEEE International Computer Vision Conference*, pages 1502 – 1509, 2003.

---

---

## Acknowledgments

It was August 2009 when I first arrived to the Netherlands to have an interview for a PhD position at TU Delft. I directly took the train from the airport to Delft. I was told by Pieter Jonker that a guy called Oytun Akman would take me from the train station. Probably my curious looks revealed my identity and a huge guy with his undersized foldable bike started to approach me. After a small chat he said “Ok, it is better if you leave your stuff to your hotel.” Then, I asked, “What hotel?”. As a disorganized person, I came to the Netherlands just with my backpack and didn’t even think about arranging an accommodation. Soon, we figured out that neither TU Delft did since I didn’t even ask for it. Oytun called his wife, Melis Akman, to ask if a hairy guy, who he met a couple of minutes ago, could stay with them that night. We went to their house and talked until late in the night, and I crashed on their couch. That day, I felt lucky to find these friendly people, but little did I know that it was the beginning of wonderful friendships. I had the interview the next day and it went well; I was accepted to TU Delft! After the arrangement of my papers, I started working at TU Delft on December 2009. A couple of days after my arrival, I had an e-mail from Caner Hamarat. Even though we were neighbours from the dormitories of Sabanci University, we hardly knew each other: he once visited our room to ask for a cigarette, and we forced him to dance (twist) in exchange (and he did); I didn’t see him again. In his e-mail, he was saying that he and his girlfriend, Nalan Liv, was working in TU Delft too. We met shortly after his e-mail and never stopped meeting again. These people, Oytun, Melis, Caner and Nalan, were with me all the time, at all the stages of my stay in the Netherlands. When life was boring or sad they made it joyful, when it was joyful, they made it even better. I sometimes thought that life was easy; but now I notice that it was because they were helping me all the time, even sometimes without me noticing it.

In the second week of my arrival, I was invited to a Turkish night. As I arrived, I met Nihal Yeniad, Alper Alkan, Sevinç Goral Alkan, Nazlı Cila, Olgu Çaliskan, İlhan Polat, Esra Bektaş, Caner Anaç, Neşen Sürmeli Anaç, Umut Çorbacıoğlu (x2), Merve Bedir, Özge Aydin, Neşe Direk, Özge Sahin, Bilge İmamoğlu. It was

such a fun with a lot of chatting, music, singing, laughter, dancing... (I believe everyone who joined one of these Turkish nights remember it with a smile in their face). After that night, I didn't have any worries about my stay in the Netherlands anymore. With such cheerful, warm-hearted and energetic people, I already knew that it will be a lot of fun. Soon after, I have met Alex Çarıkçı, Emre Erol, Nihal Erol, Yunus Sarıçay, Narin Tezcan, Sinan Tüfekçioğlu, Vanya Uluç, Gözde Dere, Ali Güney each of which has a very special place in my heart. Meetings with them was always so refreshing and I was always looking forward to the next one. Many things have changed in five years: some got married, had kids, some had new jobs, moved from the Netherlands etc. Inevitably, these changes made it harder to meet frequently. However, the importance of these wonderful people in my life never changed and I feel very lucky to meet them by the first days of my stay.

The last stages of Ph.D. are supposed to be painful. But it was totally bearable for me thanks to Burak Sözen, Çağrı Tekinay, Argun Çençen, Nazlı Cila, Güncem Gürtekin, Çiğdem Demirel, Esin Dağlıoğlu, Onur Dağlıoğlu, Feyza Sarı, Taner Sezgin, Sine Çelik, or in short "the Şamandiring family". We met too late, but built up so many beautiful and cheerful memories in such a short period of time. We was so nice to lose the sense of time with you.

As I was leaving the Netherlands, I felt it was too early to be far away from all these beautiful people I love. I will never forget: the enlightening and joyful talks with Alper and Sevinc, the beautiful Den Haag tours with Alex, the amazing bike rides with Caner and Nalan, the confusing Mogwai concert with Melis, the parties we have been to with Merve, Neşe and Özge, all the amazing concerts we have been to with Yunus and Nazlı (Yunus, I am always in for the Berlin plan! One day!), the radio sessions with Caner Anac, taking photos at 3 a.m. with Oytun on my birthday, the after work beers with Esra, the surreal disco night at Emre and Nihal's home, tea parties with Nihal and Narin, the basketball fun we had with Olgun and the night at France with the Samandiring family!

My working environment was a peaceful, easygoing and a fruitful one. I was very lucky to have Pieter Jonker as my promoter, and Martijn Wisse as my supervisor. By providing complementary feedback, they did not only teach me a lot about how to conduct a research project, but also broaden my view as a scientist: Pieter showed me the joy of doing research, and taught me to think ahead and to dream about the future. Martijn saved me from my disorganized thinking, and taught me to be a systematic researcher.

I would like to thank to the whole Delft Biorobotics Laboratory team for helping me with their inspiring comments and discussions. I would especially like to thank to Wouter Caarls (never got tired from my neverending questions), Tim Vercruyssen and Erik Schuitema for being such good room mates, for their extensive help in every stage of my thesis project, and also being much more than colleagues, but friends; it was a joy to work and chat with you. Also, thanks to Mukunda Bharatheesha, Loubna Bouarfa, Xin Wang and Michiel Plooij for their precious friendship and support. I would also like to thank to Guus Liqui Lung and Jan van Frankenhuyzen for helping me to realize the test setups used

in this thesis. Also thanks to the Delft Robotics team: Maarten, Kanter, Bas, Hans, Ronald, Mihai, Arie and Martijn for teaching me so much during our short collaboration.

The times abroad are hard while missing friends. I would like to thank to Firat Ant for always reminding me that I have a brother in Turkey. His lifelong friendship is irreplaceable. I never felt faraway from Emre Düzgün, Irem Düzgün, Cihan Başkır, Dogan Başkır, Çağrı Ulucenk, Emrah Deniz Kunt, Deniz Erdoğan Kunt, Cevdet Göktepe, Adil Küsem, Hakan Kaynar. They were always there for me, made the distances closer and never left me alone. It always made me so happy to keep in touch with them during my entire stay.

Also special thanks to people who helped me finalizing this thesis: when I was away from the Netherlands: Thanks to Nalan Liv and Nazlı Cila for accepting being my paronyms, helping me out of the thesis submission procedure and printing the thesis, thanks again to Nazlı for designing the beautiful cover of this thesis in such a short time, and thanks to Caner Hamarat and Burak Sozgen for saving me so much time and effort while moving.

Finally, my family... This thesis is dedicated to them as their endless love, support and understanding made so many beautiful things in my life possible. I love you.

*Berk*



---

---

## About the Author

Berk Çalli was born in İzmir, Turkey on February 23, 1984. In 2001 he obtained his high-school diploma from Özel Çakabey Lisesi in İzmir, Turkey. In 2002, he was accepted to Sabancı University (SU), İstanbul, Turkey. He studied in the mechatronics B.Sc. program in SU. After completing his B.Sc. studies in 2006, he continued his M.Sc. studies under the supervision of Assoc. Prof. Dr. Kemalettin Erbatur and Prof. Dr. Mustafa Ünel in the same department and received his M.Sc. degree on the subject of integrated force and vision based control for robotic arms in 2008. He continued to work at Sabancı University as a researcher for one more year with Prof. Mustafa Ünel and worked on variable delay compensation in bilateral teleoperation.

In 2009, he started his PhD study on robotic manipulation under the supervision of Prof.dr.ir P.P. Jonker and Dr.ir. M. Wisse at the Delft Biorobotics Lab, Delft University of Technology. His PhD research was focused on the utilization of active vision algorithms for the robotic manipulation problem. The resulting algorithms are further applied to the active object recognition problem.

On October 2014, Berk joined Yale University GRAB Laboratory as a postdoctoral associate. He now focuses on developing grasping strategies for underactuated robotic hands and benchmarking in robotic manipulation.