Optimization-based Robot Grasp
Synthesis and Motion Control

Örebro Studies in Technology 61

ROBERT KRUG

Optimization-based Robot Grasp Synthesis and Motion Control

Title: Optimization-based Robot Grasp
Synthesis and Motion Control
Publisher: Örebro University 2014
www.oru.se/publikationer-avhandlingar

Print: Örebro University, Repro 05/2014

ISSN 1650-8580
ISBN 978-91-7529-026-3

Abstract

Robert Krug (2014): Optimization-based Robot Grasp Synthesis and Motion Control. Örebro Studies in Technology 61.

This thesis investigates the questions of where to grasp and how to grasp a given object with an articulated robotic grasping device. To this end, aspects of grasp synthesis and hand motion planning and control are investigated. Grasp synthesis is the process of determining a palm pose with respect to the target object, as well as a hand joint configuration and/or grasp contact points such that a successful grasp execution is allowed. Existing methods tackling the grasp synthesis problem can be categorized in analytical and empirical approaches. Analytical approaches are based on geometric, kinematic and/or dynamic formulations, whereas empirical methods aim at mimicking human strategies.

An overarching idea throughout this thesis is to circumvent the curse of dimensionality, which is inherent in high-dimensional planning problems, by incorporating empirical data in analytical approaches. To this end, tools from the field of constrained optimization are used (i) to synthesize grasp families based on available prototype grasps, (ii) to incorporate heuristics capturing human grasp strategies in the grasp synthesis process and (iii) to encode demonstrated grasp motions in primitive motion controllers.

The first contribution is related to the computation and analysis of grasp families which are represented as Independent Contact Regions (ICR) on a target object's surface. To this end, the well-known concept of the Grasp Wrench Space for a single grasp is extended to be applicable for a set of grasps. Applications of ICR include grasp qualification by capturing the robustness of a grasp to position inaccuracies and the visual guidance of a demonstrator in a teleoperating scenario. In the second main contribution of this thesis, it is shown how to reduce the grasp solution space during the synthesis process by accounting for human approach strategies. This is achieved by imposing appropriate constraints to a corresponding optimization problem. A third contribution in this dissertation is made to reactive motion planning. Here, primitive controllers are synthesized by estimating the free parameters of corresponding dynamical systems from multiple demonstrated trajectories. The approach is evaluated on an anthropomorphic robot hand/arm platform. Also, an extension to a Model Predictive Control (MPC) scheme is presented which allows to incorporate state constraints for auxiliary tasks such as obstacle avoidance.

Keywords: Robot Grasping, Grasp Synthesis, Grasp Planning, Motion Control, Model Predictive Control, Independent Contact Regions, Obstacle Avoidance, Motion Planning.

Robert Krug, School of Science and Technology
Örebro University, SE-701 82 Örebro, Sweden, robert.krug@oru.se

Acknowledgments

"If we knew what it was we were doing, it would not be called research, would it?" This tongue-in-cheek quote, attributed to Albert Einstein, nicely sums up the wonderful and challenging experience I had while working towards my PhD degree in the AASS Research Center at Örebro University. I discovered that research is a team effort, were the reward not only lies in finding solutions to long-pondered problems, but that one also grows along the way on a professional, as well as on a personal level.

Over the past years, many people were instrumental – directly or indirectly – in setting me on the path of an academic career. First and foremost, I would like to thank my supervisors Achim Lilienthal and Dimitar Dimitrov for allowing me to work under their guidance at AASS. Thanks Achim for always having my back. Your encouragement and faith in me throughout the past years have been extremely helpful. Mitko, I am grateful for your unfading enthusiasm and for showing me what true passion towards research means. You were always available for fruitful discussions and always found time to read and comment on my articles. I would not have made it without your help and I promise to never use crazy notation again. Next, I want to extend my thanks to all my co-workers and friends at the University for making this place the friendly, productive, dynamic and inspiring environment that it is. Thank you Todor for being a great colleague, and for sharing more-or-less exotic culinary experiences between Taipei, Pisa and Bremen. Marco, Luigi and Lina, thanks for shared adventures on and off the snow. Fabrizia, I am grateful for convincing me that those "little numbers" are not everything (although I still think that they are more important than you believe). Thanks Silvia and Maurizio for being great flatmates, and Marcello for providing shelter in times of need.

Finally, I want to express my appreciation for the people closest to me for their love and support: my family, who always provided a safe haven on my (far too rare) visits home and especially my mother, who always wholeheartedly supported my education – it looks like the chances that I make something of myself are intact. Last but not least, thank you Delia for your unwavering patience and encouragement, especially during the stressful period of compiling this thesis – *Always on My Mind!*

"At bottom, robotics is about us. It is the discipline of emulating our lives, of wondering how we work."

Rod Grupen, Discover Magazine, 2008

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1
Introduction

In 1920, Czech writer Karel Čapek published the play R. U. R. (Rossum's Universal Robots), which deals with the ethical implications of using artificially created people as cheap labor. This work is regarded as the origin of the term *robot* which comes from the Czech word robota, meaning "hard work". And indeed, nowadays robots have become an indispensable centerpiece of automated manufacturing processes. In industrial settings, where cost-effectiveness is paramount, they reliably carry out a plethora of tasks such as welding, painting, machining, material transport, assembly and packaging. In this context, an often cited acronym characterizes the tasks a robot should perform as the three D's – Dull, Dangerous and Dirty. In recent years robots have become available as consumer products in other domains such as service and entertainment. Successful examples include Sony's Aibo [1], which is a series of robotic pets, and iRobot's Roomba [2], a robotic vacuum cleaner.

An important aspect in many robot applications across all domains is the interaction with the environment. The interface between a robot manipulator (arm) and the environment is provided in form of end-effectors. Today, the majority of end-effectors, such as suction cups and parallel-jaw grippers, is simple and tailored to carry out specific tasks on specific objects. In order to avoid the need for changing end-effectors on a task-to-task basis, versatile and dexterous end-effectors are required. A solution is offered in form of articulated multi-fingered hands [3]. These are grasping devices which possess the ability to reconfigure themselves for performing different grasps. Such mechanisms were built first in the early 1980's. Among them are the Stanford/JPL hand [4] and the Utah/MIT hand [5], the latter of which is shown in Fig. 1.1(a). One line of research has focused on devising anthropomorphic (human-like) devices which attempt to mimic the human hand with its unsurpassed dexterous grasping and manipulation capabilities. An example is the hand/arm system in [6], which is engineered by the German Aerospace Center (DLR) and depicted in Fig. 1.1(b). Devices in this mould are advantageous for applications such as teleoperation, prosthetics and for service robots in a human environment. However, anthro-

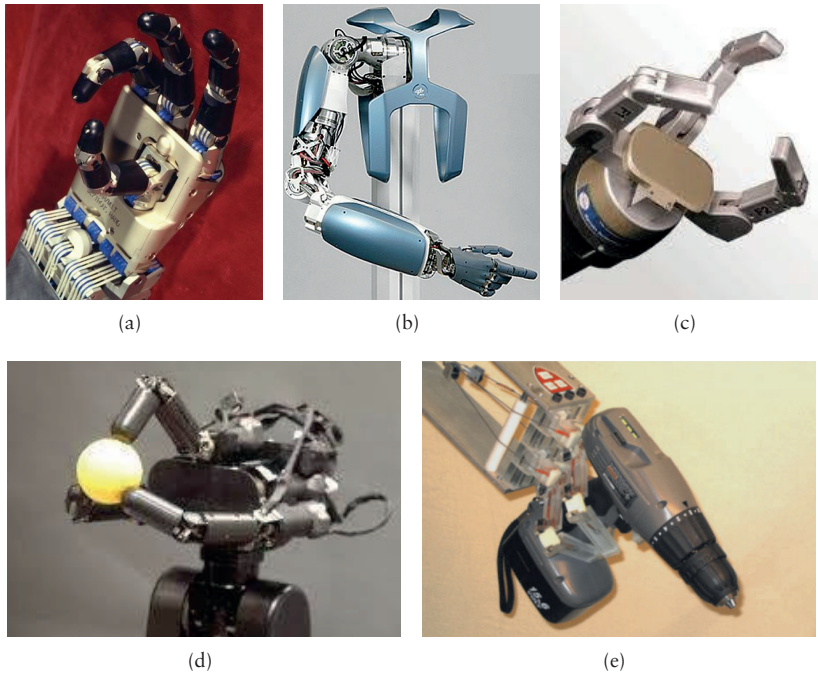


Figure 1.1: *Articulated Grasping Devices:* (a) The Utah/MIT hand, one of the first multi-fingered hands. (b) The DLR hand/arm system is one of the most advanced anthropomorphic platforms today. (c) The 3-fingered Barrett hand features break-away transmissions in the distal joints which allows for robust grasping. (d) The lightweight high-speed hand by Namiki et al. [7] allows for real-time visual feedback control. (e) The underactuated SDM hand is a low-cost compliant grasping device using a single actuator only.

pomorphism is neither necessary nor sufficient to achieve dexterity. There exist many impressive grasping and manipulation devices with alternative mechanical structures. Examples include the Barrett hand [8] (see Fig. 1.1(c)), which has become a popular research tool, and the high-speed hand in [7], which can perform highly dynamic tasks such as catching objects and is depicted in Fig. 1.1(d). Underactuated grippers comprising less actuators than Degrees of Freedom (DoF), such as the SDM hand [9] shown in Fig. 1.1(e), provide interesting and cost-effective alternatives. Here, the mechanisms are designed such that certain desired grasping/manipulation features are preserved.

In most robotic applications today, behaviors and motions are pre-programmed. In order for robots to leave the structured environments of industrial or laboratory settings and to succeed in uncontrolled scenarios, it has become

clear that they need to be endowed with a sufficient level of autonomy. To purposefully interact with its environment, and as a prerequisite for any subsequent manipulation, a robot needs to be able to autonomously grasp objects in a robust manner which is the focus of this dissertation.

1.1 The Challenges of Autonomous Robot Grasping

The DARPA Robotics Challenge (DRC) is a prestigious competition funded by the US Defense Advanced Research Projects Agency with the aim to push the boundaries regarding supervised autonomy in an emergency-response scenario for mobile, mostly humanoid robots. In this context, supervised autonomy means that there is a human teleoperator in the loop which can issue commands, albeit under the constraint of a limited bandwidth. The DRC trials held in 2013, the year before writing this thesis, included manipulation tasks such as opening a door or closing a valve. Even the most successful robots used up at least half of their 30 minutes time limit per challenge and a significant number of attempts failed. The purpose of the above example is to highlight the substantial difficulty of achieving even only partial autonomy in robot grasping and manipulation and the big gap between the capabilities of fictional robots and currently existing systems. With respect to humans, Neuroscience has shown that the largest fraction (30-40%) of the motor cortex, i. e., the region of the brain responsible for movement planning and execution, is dedicated to the control of the hand [3]. For a robot, successfully grasping an object entails solving the problems of object perception and grasp synthesis, as well as hand and manipulator motion planning.

Object perception estimates the pose of the target object and, if not known a priori, its geometry from potentially incomplete and noisy sensor data. Solving this problem is aggravated by factors like occlusions of the target object by the environment or the robot itself, and varying light conditions across different scenarios which necessitate different calibrations/setups of the employed range sensing devices. Once a representation of the object is built by means of the available sensor inputs, it is necessary to address the grasp synthesis problem. Here, the goal is to determine a hand palm pose with respect to the object, as well as a joint configuration and/or grasp contact points such that a successful grasp can be achieved by an appropriate hand closing motion. This process is not trivial, especially considering uncertainties in the target object's pose and the achievable positioning accuracy of the robot platform. The purpose of hand motion planning is to generate a coordinated grasp movement which is particularly relevant when complex hands with many DoF are considered. Finally, manipulator motion planning is concerned with finding a collision-free path leading the grasping device from the initial pose to the grasping pose.

Problem Statement

At this point, the general problem of interest in this dissertation can be stated.

Problem: *Given the pose and geometry of an object to be grasped with an articulated robotic grasping device, determine an appropriate set of contact points, palm pose and gripper joint configuration such that a coordinated grasp closing motion results in a stable grasp.*

Loosely speaking, the addressed question is *where* to grasp and *how* to grasp a given object. To this end, aspects of grasp synthesis and hand motion planning are investigated. The experiments presented in this work were conducted by using existing solutions for object perception and manipulator motion planning. A central tenet in this thesis is to circumvent the curse of dimensionality, which is inherent in high-dimensional planning problems, by incorporating empirical data in analytical approaches. Most of the proposed algorithms encapsulate a notion of optimality in the context of the tackled sub-problem. Therefore, the use of tools from numerical optimization is a second central aspect in this dissertation.

1.2 Outline

The rest of this thesis is organized as follows.

Chapter 2 puts the dissertation in the context of the two EU-funded projects in whose scope the presented work was carried out. Furthermore, the necessary background, as well as the utilized hardware and robots are discussed.

Chapter 3 provides an overview of relevant related work in the fields of grasp synthesis and reactive motion generation.

Chapter 4 introduces algorithms for synthesizing contact-level grasp families based on a prototype grasp and a notion of expected disturbances.

Chapter 5 gives a numerical evaluation and application examples of the methods presented in the previous chapter. The range of applications includes grasp qualification, guided teleoperation, interactive grasp transfer and finger gait planning.

Chapter 6 presents an optimization-based grasp synthesis and execution scheme which is tailored to the specifics of an underactuated grasping device. The approach incorporates empirical data in form of grasp strategies observed in humans.

Chapter 7 discusses a reactive real-time framework for generating coordinated hand motions. The method is based on data provided in form of human

demonstrations and can be extended to incorporate auxiliary goals such as obstacle avoidance.

Chapter 8 finally concludes this dissertation and summarizes the major contributions and directions of future research based on the presented work.

1.3 Contributions

The major contributions of this thesis, as outlined in the previous section, can be summarized as follows:

Grasp synthesis algorithms which extract a family of similar contact-level grasps from a provided prototype and allow to prioritize specified fingers.

An open-source C++ library implementing the aforementioned algorithms.

Practical applications of contact-level grasp families ranging from grasp qualification to visually guided teleoperation, interactive grasp transfer and finger gait planning.

An optimization-based grasp synthesis framework which incorporates heuristics based on human grasp strategies.

A grasp execution routine using the active surfaces of a gripping device for in-hand manipulation to increase the stability of an initial grasp.

A reactive motion generation framework whose output resembles human demonstrations.

A control scheme which allows for real-time obstacle avoidance.

1.4 Publications

The core of the work presented in this dissertation has either been published in various peer-reviewed articles, or is under review at the time of writing. The following list summarizes all the publications accomplished during the course of working towards this thesis, as well as the particular chapters of this work that each article contributed to. Author's copies of the publications are available online at `http://www.aass.oru.se/Research/Learning/rtkg.html`.

- Robert Krug, Krzysztof Charusta and Dimitar Dimitrov, "Constructing Independent Contact Regions based on the Exertable Wrench Space: Theory, Implementation and Applications to Robot Grasping", *International Journal of Robotics Research (IJRR)*, 2014, under review.
  Main part of Chapters 4 and 5

- Robert Krug and Dimitar Dimitrov, "Model Predictive Motion Control based on Generalized Dynamical Movement Primitives". *Journal of Intelligent & Robotic Systems (JINT), Special Issue on the* 16^{th} *International Conference on Advanced Robotics*, 2014, under review.
  Main part of Chapter 7

- Robert Krug, Todor Stoyanov, Manuel Bonilla, Vinicio Tincani, Narunas Vaskevicius, Gualtiero Fantoni, Andreas Birk, Achim J. Lilienthal and Antonio Bicchi, "Improving Grasp Robustness via In-Hand Manipulation with Active Surfaces", In *IEEE Int. Conf. on Robotics and Automation (ICRA) – Workshop on Autonomous Grasping and Manipulation: An Open Challenge*, 2014, under review.
  Part of Chapter 6

- Robert Krug, Todor Stoyanov, Manuel Bonilla, Vinicio Tincani, Narunas Vaskevicius, Gualtiero Fantoni, Andreas Birk, Achim J. Lilienthal and Antonio Bicchi, "Velvet Fingers: Grasp Planning and Execution for an Underactuated Gripper with Active Surfaces", In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014, to appear [10].
  Main part of Chapter 6

- Robert Krug and Dimitar Dimitrov, "Representing Movement Primitives as Implicit Dynamical Systems learned from Multiple Demonstrations", In *Proc. of the Int. Conf. on Advanced Robotics (ICAR)*, 2013 [11].
  Part of Chapter 7

- Robert Krug, Dimitar Dimitrov, Krzysztof Charusta and Boyko Iliev, "Prioritized Independent Contact Regions for Form Closure Grasps", In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011 [12].
  Part of Chapter 4

- Robert Krug, Dimitar Dimitrov, Krzysztof Charusta and Boyko Iliev, "On the Efficient Computation of Independent Contact Regions for Force Closure Grasps", In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010 [13].
  Part of Chapter 4

Not included in this dissertation:

- Krzysztof Charusta, Robert Krug, Dimitar Dimitrov and Boyko Iliev, "Independent Contact Regions Based on a Patch Contact Model", In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012 [14].

- Krzysztof Charusta, Robert Krug, Todor Stoyanov, Dimitar Dimitrov and Boyko Iliev, "Generation of Independent Contact Regions on Objects Reconstructed from Noisy Real-World Range Data", In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012 [15].

- Erik Berglund, Boyko Iliev, Rainer Palm, Robert Krug, Krzysztof Charusta and Dimitar Dimitrov, "Mapping between different kinematic structures without absolute positioning during operation", In *Electronics letters*, 2012 [16].

- Erik Berglund, Boyko Iliev, Rainer Palm, Robert Krug, Krzysztof Charusta and Dimitar Dimitrov, "Mapping between different kinematic structures without absolute positioning", In *IEEE Int. Conf. on Robotics and Automation (ICRA) – Workshop on Autonomous Grasping*, 2011 [17].

In all articles for which I am first author, I performed the relevant software implementations and tests, as well as the major part of analyzing and reporting the obtained results. In the paper in [10], Todor Stoyanov and Narunas Vaskevicius were responsible for object modeling, object database maintenance and perception. Manuel Bonilla and Vinicio Tincani provided support for manipulator motion planning and performing test runs. For the works with Charusta et al. [14, 15], I collaborated in the design of algorithms and dissemination. To the works of Berglund et al. [17, 16], I contributed in form of discussions.

Chapter 2
Background

The research presented in this dissertation was carried out in the context of two EU-FP7 projects: “Handle” and “RobLog”. The former aimed at endowing a complex anthropomorphic hand/arm platform with skilful grasping and manipulation capabilities. Here, the focus was on understanding how humans conduct everyday tasks and to find compact representations of the underlying strategies in a Teaching by Demonstration (TbD) setting. Roblog, the second project, is still ongoing at the time of writing this thesis. It is aimed at autonomous unloading of shipping containers in a logistics scenario. Surprisingly, this is a task which is still mostly carried out by manual labor, despite of its strenuous and hazardous nature. Here, the goal is to use an underactuated gripper with a low number of actuated degrees of freedom to robustly grasp geometrically simple objects such as boxes, barrels or coffee sacks.

Despite their fundamentally different nature, it showed that the underlying idea of this thesis – combining empirical with analytical approaches based on optimality criteria – is applicable to common sub-problems arising in both of these projects. Defining a contact-level grasp in terms of discrete contact points on the surface of the target object and assuming one contact point per fingertip allows to solve the grasp analysis problem independent of the grasping device [18]. This makes the underlying theory applicable to any dexterous hand. However, the subsequent problems of finding suitable palm poses and hand joint configurations to execute that grasp inherently depend on the specifics of the considered gripping device and necessitate dedicated solutions. The resulting overall problem tackled in this thesis can be categorized as follows:

- Grasp contact point synthesis,
- Palm pose synthesis,
- Hand joint configuration synthesis,
- Hand motion planning and control.

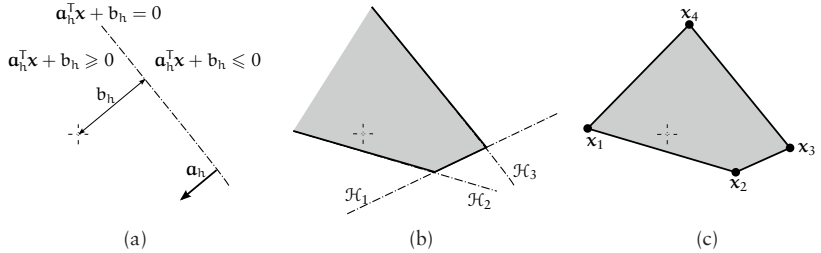


Figure 2.1: *Polyhedra and Polytopes*: (a) According to (2.1), a hyperplane $\mathcal{H}_h$ is defined by its normal unit vector $\mathbf{a}_h$ and offset b_h. Also indicated are the associated closed inner and outer half-spaces in Equations (2.2) and (2.3) respectively. (b) Shown is the polyhedron formed by the intersection of interior half-spaces $\mathcal{H}_1^+ \cap \mathcal{H}_2^+ \cap \mathcal{H}_3^+$ corresponding to the three depicted hyperplanes. (c) A polytope formed by the convex hull in (2.4) over the points in $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_4\}$.

In accordance with the above structure and starting with Chapter 4, at the beginning of each chapter an indication is given as to which specific sub-problem is addressed.

In principal, successful grasping requires the solution of an additional problem – determining appropriate contact forces [19]. In this work, instead of explicitly calculating and controlling these interaction forces, the experiments described in Sections 7.5.2 and 6.3.3 were generated via simple stiffness-based interaction control [20] using low-gain joint level position controllers.

2.1 Polyhedra

Here we summarize some of the relevant concepts and terminology regarding polyhedra, which are the fundamental geometric objects used in this thesis. Due to their importance in a wide range of applications they have been studied extensively [21, 22, 23]. Let us define a hyperplane in k-dimensional space by means of the equality

$$\mathcal{H}_h = \left\{\boldsymbol{x} \in \mathbb{R}^k : \mathbf{a}_h^T\boldsymbol{x} + b_h = 0\right\}, \tag{2.1}$$

where $\mathbf{a}_h \in \mathbb{R}^k$ denotes a unit normal vector and $b_h \in \mathbb{R}$ is a scalar offset as shown in Fig. 2.1(a). Figure 2.1(b) exemplary depicts a polyhedron which can be defined as the intersection of closed half-spaces associated to a finite number of u hyperplanes in (2.1). In this context, a closed inner half-space is given via the inequality

$$\mathcal{H}_h^+ = \left\{\boldsymbol{x} \in \mathbb{R}^k : \mathbf{a}_h^T\boldsymbol{x} + b_h \geqslant 0\right\}. \tag{2.2}$$

Similarly, a closed outer half-space is denoted as

$$\mathcal{H}_h^- = \left\{\boldsymbol{x} \in \mathbb{R}^k : \mathbf{a}_h^T\boldsymbol{x} + b_h \leqslant 0\right\}. \tag{2.3}$$

Collecting its hyperplane normals in the matrix $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_u]^T \in \mathbb{R}^{u \times k}$ and the corresponding offsets in the vector $\mathbf{b} = [b_1, \ldots, b_u]^T \in \mathbb{R}^u$ allows to denote a polyhedrons $\mathcal{H}$-representation as the pair $(\mathbf{A}, \mathbf{b})$. If a polyhedron is bounded, it is referred to as a polytope.

Consider the finite set of k-dimensional vectors $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_c\}$, $\mathbf{x}_i \in \mathbb{R}^k$. A cone is defined as the set of all conic combinations of elements in $\mathcal{X}$

$$\text{cone}(\mathcal{X}) = \left\{ \sum_{i=1}^{c} \mathbf{x}_i \alpha_i \in \mathbb{R}^k : \alpha_i \geqslant 0, \ i = 1, \ldots, c \right\},$$

where the coefficients $\alpha_i \in \mathbb{R}$ are positive real numbers. In a similar fashion, the convex hull can be formalized as the set of all convex combinations of the elements in $\mathcal{X}$

$$\text{conv}(\mathcal{X}) = \left\{ \sum_{i=1}^{c} \mathbf{x}_i \alpha_i \in \mathbb{R}^k : \sum_{i=0}^{c} \alpha_i = 1, \ \alpha_i \geqslant 0, \ i = 1, \ldots, c \right\}. \tag{2.4}$$

In the above definition, the coefficients $\alpha_i \in \mathbb{R}$ additionally have to sum up to one. The convex hull over a set of points $\mathcal{X}$ forms a convex polytope as illustrated in Fig. 2.1(c). A face of a k-dimensional polytope is described as any intersection of this polytope with a half-space, such that none of the polytope's interior points lie on the boundary of that half-space. Faces are denominated according to their dimensions: a vertex is a 0-dimensional face, an edge is a 1-dimensional face, a ridge is a $k-2$-dimensional face and a facet is a $k-1$-dimensional face. Each ridge connects two facets. In this thesis, it is assumed that convex polytopes are are given in simplicial form, i. e., a facet is spanned by k vertices. The reason for this assumption will become clear in Section 4.6.2. Furthermore, we denote the face lattice as the partially ordered set of all faces of a convex polytope, the ordering is by set inclusion.

2.2 Grasps and Independent Contact Regions

In this dissertation, the surface of a target object is discretized and represented as a set of points whose indices are collected in the set $\mathcal{O} = \{1, \ldots, o\}$. For a specific grasping device, a f-fingered grasp

$$G = (\mathbf{P}, \mathbf{q}) \tag{2.5}$$

is denoted as a pair of hand joint configuration vector $\mathbf{q} = [q_1, \ldots, q_t]^T$ and palm pose $\mathbf{P} \in \mathbb{R}^6$, which can be written in terms of three Euler angles and a position vector. Additionally, we define a contact-level grasp

$$\mathcal{G} = \{g_i \in \mathcal{O} : i = 1, \ldots, f\} \tag{2.6}$$

as a set of contact point indices. This definition does not depend on the specifics of any grasping device. For given hand kinematics and geometry, a contact-level

grasp $\mathcal{G}$ can be expressed via the forward kinematics map as a function of a grasp G, *i. e.*, $\mathcal{G} = \mathcal{G}(\mathrm{G}(\mathbf{P}, \mathbf{q}))$.

One prevalent concept in this dissertation is to associate a contact-level grasp with a set of ICR on the target object's surface, such that one such region is associated with each fingertip (see Fig. 4.1 for an illustrative example). These regions are constructed such that each member of the family of contact-level grasps formed by placing every finger anywhere within its respective region is guaranteed to qualify for pre-specified tasks. This representation provides robustness in front of object modeling, perception and finger positioning inaccuracies since it is unrealistic to assume that a gripping device can contact the object precisely at prescribed locations. The computation of these regions is detailed in Chapter 4 and allows to incorporate empirical data in form of a provided prototype grasp in order to synthesize a family of similar grasps.

2.3 Dynamical Movement Primitives

The reactive motion generation framework presented in Chapter 7 is based on the concept of Dynamical Movement Primitives (DMP), which was originally proposed by Ijspert, Nakanishi and Schaal [24]. Essentially, a DMP is a Dynamical System (DS) which constitutes a policy over the state space and acts as an online trajectory generator for one DoF. Usually, the DS parameters are learned from empirical data provided in form of demonstrated trajectories. During execution, a motion profile is generated via integrating the corresponding DS. This allows to incorporate state feedback in real-time and thus a DMP can be seen as a mid-level controller with the ability to instantaneously react to state disturbances. To execute the generated trajectory, an additional low-level tracking controller generating appropriate motor commands is necessary.

2.4 Hardware and Robots used in this Dissertation

Human demonstrations served as inputs to the grasp synthesis algorithms in Chapters 4 and 5, as well as the reactive motion generation method presented in Chapter 7. Demonstrated data was collected by means of the sensorized glove depicted in Fig 2.2(a). Additionally, a magnetic 6D pose sensor was employed to extract palm poses for teleoperation purposes. Most of the algorithms which are proposed and evaluated in this dissertation can be easily generalized to different robots. The hand motion controllers proposed in Chapter 7 were tested on the Shadow Robot platform [25] shown in Fig. 2.2(b), which is operated by the Institut des Systèmes Intelligents (ISIR) lab at University Pierre and Marie Curie (UPMC) in Paris, France. It comprises a pneumatic 4 DoF arm and a hand with 20 actuated DoF. Five ATI-Nano17 6D force/torque sensors embedded in the fingertips enable tactile sensing. For evaluating the grasping pipeline described in Chapter 6 we employed the underactuated Velvet Fingers gripper [26, 27], which implements active surfaces by conveyor belts on each

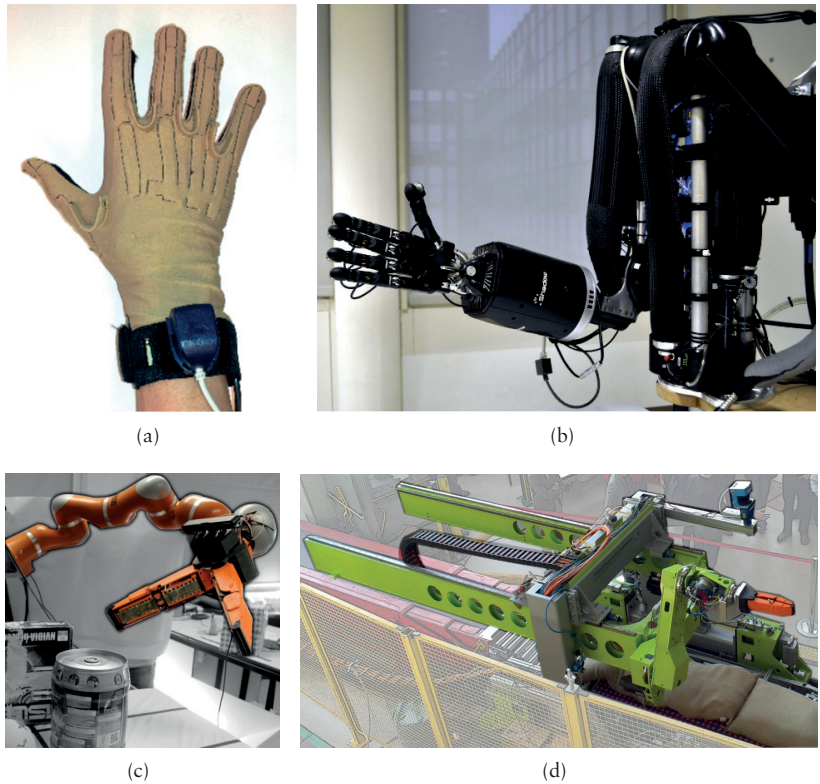


Figure 2.2: *Utilized Hardware and Robots:* (a) An Immersion Cyberglove-18 was used to record joint angles while demonstrating static prototype grasps and grasp trajectories. Additionally, in combination with the attached WinTracker 6D pose sensor, the glove was used to teleoperate the Shadow Robot platform. (b) In the Handle project, the anthropomorphic Shadow Robot hand/arm platform was used. (c) The Velvet Fingers gripper mounted on a KUKA lightweight arm used in the RobLog project. (d) Also utilized in RobLog was the Parcelrobot equipped with the Velvet Fingers Gripper.

of its two fingers. To this end we used the platform depicted in Fig. 2.2(c) at the Centro E. Piaggio, University of Pisa, Italy. The system comprises the Velvet Fingers gripper with 3 actuated DoF (one for open/close movement and one per finger for conveyor belt actuation) and a 7-DoF KUKA lightweight arm. Perception is done with an ASUS Xtion structured light camera mounted on the gripper. An additional set of test runs was done on the Parcelrobot platform which can be seen in Fig. 2.2(d). It is located at the Bremer Institut für Produktion und Logistik GmbH (BIBA) in Bremen, Germany. Equipped with

the Velvet Fingers gripper, it is a dedicated system for automatically unloading shipment containers filled with randomly packed goods. The kinematic structure comprises linear and rotary axes that cover a cylindrical workspace.

2.5 Abbreviations

Abbreviations which are used throughout this manuscript are summarized in Table 2.1.

Table 2.1: *Table of Abbreviations*: Abbreviations commonly used throughout this thesis

DoF	Degree of Freedom
TbD	Teaching by Demonstration
ICR	Independent Contact Region
CoM	Center of Mass
GWS	Grasp Wrench Space
TWS	Task Wrench Space
OWS	Object Wrench Space
EWS	Exertable Wrench Space
LP	Linear Program
NLP	Nonlinear Program
QP	Quadratic Program
SQP	Sequential Quadratic Program
TSDF	Truncated Signed Distance Field
GMM	Gaussian Mixture Model
GBF	Gaussian Basis Function
LWR	Locally Weighted Regression
MSE	Mean Square Error
STD	Standard Deviation
ODE	Ordinary Differential Equation
DS	Dynamical System
GAS	Global Asymptotic Stability
DMP	Dynamical Movement Primitive
MPC	Model Predictive Control

2.6 Notations

Some common notations and symbols used throughout this dissertation are shown in Table 2.2. Common symbols are used whenever possible in all chapters, with concepts introduced inline with the text.

Table 2.2: *Table of Notations*: Notations commonly used throughout this thesis

	General Notation
$a, \ldots, z$	Scalar quantities
$\alpha, \ldots, \omega$	
$\mathbf{a}, \ldots, \mathbf{z}$	m-dimensional column vectors
$\boldsymbol{\alpha}, \ldots, \boldsymbol{\omega}$	
$\mathbf{A}, \ldots, \mathbf{Z}$	$m \times n$-dimensional matrix quantities
$\mathbf{A}, \ldots, \boldsymbol{\Omega}$	
$\mathbf{x}^T, \mathbf{Y}^T$	Transpose of the vector $\mathbf{x}$; transpose of the matrix $\mathbf{Y}$
$\|\mathbf{x}\|_2$	Euclidean norm of vector $\mathbf{x}$ – i. e., $\|\mathbf{x}\|_2 = \sqrt{x_1^2 + \ldots + x_m^2}$
$\|\mathbf{x}\|_1$	L_1 norm of vector $\mathbf{x}$ – i. e., $\|\mathbf{x}\|_1 = \|x_1\| + \ldots + \|x_m\|$
$\mathbf{x}^\star$	Solution of an optimization problem with decision variable $\mathbf{x}$
	Set Notation
$\mathbb{R}$	Set of real numbers
$\mathbb{Z}$	Set of integer numbers
$\mathcal{A}, \ldots, \mathcal{Z}$	Generic sets
$\mathcal{A} \subseteq \mathcal{B}$	$\mathcal{A}$ is a subset of $\mathcal{B}$
$\mathcal{A} \cup \mathcal{B}$	Union of the sets $\mathcal{A}$ and $\mathcal{B}$
$\mathcal{A} \cap \mathcal{B}$	Intersection of the sets $\mathcal{A}$ and $\mathcal{B}$
$\mathcal{A} \oplus \mathcal{B}$	Minkowski sum of the sets $\mathcal{A}$ and $\mathcal{B}$ – i. e., the set formed by adding each element in $\mathcal{A}$ to each element in $\mathcal{B}$
$\mathcal{A} \setminus \mathcal{B}$	Set difference between $\mathcal{A}$ and $\mathcal{B}$ – i. e., all elements of $\mathcal{A}$ which are not in $\mathcal{B}$

Chapter 3
Related Work

3.1 Grasp Synthesis

Synthesizing grasps which are appropriate for the considered robotic platform has long been in the focus of research and is a central aspect in this dissertation. Methods operating on a grasp contact level are often referred to as *analytic* approaches [28]. Here, stable grasps are constructed by defining fingertip locations on the surface of the target object [29, 30]. Often, these methods rely on precise knowledge of hand kinematics, object geometry and the relative pose of hand and object. Ferrari and Canny [31] form grasp quality criteria which are subsequently used for synthesizing optimal grasps. Borst et al. [32, 29] suggest heuristics for the fast generation of large sets of stable grasps based on random sampling. In [33], the authors synthesize grasps by optimizing a differentiable quality metric. A review on the topic is provided by Bicchi [34].

In this thesis, to incorporate the notion of uncertainty in the synthesis process, we adopt the ICR paradigm which was introduced by Nguyen [35]. He defined the set of optimal independent regions with the largest minimal radius, which yield a force-closure grasp if each finger is placed anywhere within its respective region. Ponce et al. [36] extended the concept to the computation of independent regions for 3-fingered grasps on planar objects and 4-fingered grasps on polyhedral objects [37]. The number and distribution of points shaping ICR is not unique and depends on the underlying construction principle. To overcome the combinatorial nature of this problem, Pollard [38, 39] centered the ICR computation around an affine transformation of the GWS corresponding to an initial prototype grasp. This is a central idea in the algorithms we present in Chapter 4. The construction procedure in [39] is based on geometric reasoning. It incorporates a task related grasp quality measure to synthesize sets of whole-hand grasps which are similar to a prototype grasp comprising a large number of contacts on discretized 3-dimensional objects. Charusta et al. [15] consider a patch contact model in the region construction procedure, a way to compute continuous ICR on target objects represented in closed-form was

introduced in [40]. In a work closely related to ours, Roa and Suárez [41] suggested a fast approach which grows independent regions for precision grasps on discretized objects, but the presented computation scheme is overly restrictive. An extension of [41] which loosens the dependence of the resulting regions on the necessary seed grasp was recently proposed in [42]. A detailed comparison between the works in [41, 42] and the work conducted in this thesis is given in Section 4.6.3.

As argued in the previous chapter, for a practical grasp synthesis system it is not sufficient to only plan contact locations. Also appropriate palm poses and hand joint configurations need to be found. In the RobLog project, a *data-driven* approach [43, 44, 45] which accounts for empirical data in the synthesis process has been adopted. Data-driven methods generate grasp hypotheses for a given object in a knowledge database and often provide a minimalistic grasp definition, *e. g.*, only the approach vector [46]. Combined with appropriate heuristics for grasp execution and/or learning schemes [47], in many cases this has been shown to be robust to uncertainties inherent in a robotic system. To simplify the grasp synthesis and subsequent retrieval from the database, it has been suggested to approximate the target object with primitives or superquadrics [48, 49, 50]. A commonly used strategy to compute the underlying grasp hypotheses is to sample the surface or bounding-box normals of the target object and to use them as approach vectors in a simulation where the fingers are closed once the gripper's palm contacts the object [51, 52]. A wrench-based geometric quality criterion, such as the one in [31], is usually used to rank the grasps. Alternatively, as discussed in Chapter 6, suitable pre-grasps can be created by minimizing an appropriate energy function as demonstrated in [53, 54]. Again, the final grasp quality evaluation is usually done after auto-closure of the fingers in a static simulation (*i. e.*, using a spatially fixed object and only performing forward kinematics and collision checks while ignoring interaction forces) [54, 45]. However, for underactuated simple grippers as the one used in the RobLog project, this strategy is unsuitable because it fails to accurately predict the final grasp configuration which depends on the interaction between gripper and object. We refer to Bohg et al. [55] for a complete recent review on data-driven grasp synthesis approaches.

The goal of the RobLog project is to autonomously unload shipment containers filled with randomly packed goods. This requires grasp synthesis and execution in cluttered scenes, which is subjected to intrinsic difficulties due to the fact that many pre-planned grasps are not reachable in such environments. In chapter 6 we adopt an approach similar to the one in Berenson et al. [56], who address this problem by online computation of a grasp score based on heuristics. The approach in [57, 58] allows for simultaneous contacts of the gripping device with multiple objects. Pushing actions are used to manipulate otherwise non-graspable objects. Saxena et al. [59] present a vision-based approach which accounts for uncertainty in the target object's location during planning and grasp selection.

3.2 Reactive Motion Planning and Obstacle Avoidance

Dynamical systems have become a popular framework for encoding motions. Our motion generation system, which was used in the Handle project to generate hand joint motions, is described in Chapter 7 and is based on the DMP framework [24]. Here, the underlying DS (usually referred to as the transformation system) consists of a predefined stable linear DS which is modulated by a nonlinear forcing function that decays over time ensuring Global Asymptotic Stability (GAS). Arbitrarily many DoF can be synchronized via a phase variable (whose evolution is governed by the so called canonical system) which acts as a substitute of time. The learning problem is usually solved by fixing the nonlinear parameters of the forcing function and fitting only the linear parameters with Locally Weighted Regression (LWR) [60]. The DMP framework (see [61] for a recent review) can be used to generate point-to-point motions as well as periodic movements and lends itself well to reinforcement learning techniques [62, 63, 64, 65, 66]. Although DMP offer a compact way of capturing the dynamics of a single demonstration, the actual underlying dynamics can differ substantially in regions of the state space not covered by this demonstration. Hence, it is desirable to account for multiple different demonstrations to increase generalization.

In this thesis, we use embedded optimization to generalize over a set of demonstrations capturing a movement. Most works aiming at generalization of DMP are based on statistical learning techniques. Pastor et al. [67] build a library of template primitives which can be used for sequencing movements. Matsubara et al. [68] learn DMP from multiple demonstrations and combine them using a style parameter. In [69], a statistical movement representation using Gaussian Mixture Regression is proposed. Ude et al. [70] suggest to keep multiple demonstrated trajectories in memory and to synthesize new primitives using LWR in order to compute local models. This approach was extended in [71] to make it feasible for online computation by directly representing demonstrations as DMP and utilizing Gaussian Process Regression to compute new DMP parameters depending on a given desired goal point. Similarly, in [72] striking movements for table tennis are learned by mixing primitives via a gating network.

An alternative DS model structure was proposed by Gribovskaya et al. [73]. Here, the authors define a locally stable DS via a probabilistic representation of the demonstrations as a Gaussian Mixture Model (GMM). Their system is time-independent which, depending on the application, can increase robustness in the presence of temporal perturbations. Furthermore, only one DS is learned which potentially allows to capture coupling effects between different DoF. Extending the work in [73], Khansari-Zadeh et al. [74] introduce the Stable Estimator of Dynamical Systems (SEDS) approach. Here, the parameters of

the GMM are estimated by solving a Nonlinear Programming Problem (NLP). As in [73], SEDS learns a single time-independent coupled DS with additional constraints guaranteeing that the system is GAS. However, as stated by the authors in [74], with increasing number of DoF the learning problem can become intractable.

In a reactive planning setting based on DS, obstacles are typically dealt with locally by augmenting the DS formulation with repelling potential fields [75, 76]. Alternatives include the use of coupling feed-forward terms [77] and appropriate modulation of the original DS depending on the distance of the current state to the obstacles [78, 79]. With increasing maturity of online optimization algorithms and solvers, it is becoming feasible to formulate obstacles directly as constraints in the state space [80, 81]. In Chapter 7, we present a MPC-based approach to motion control in the presence of obstacles. Approaches in this mould require online solution of optimization problems during motion execution, in order to ensure that the constraints are obeyed at each point in time. Variants of this concept have recently been successfully applied to on-line path planning schemes for autonomous/semi-autonomous vehicles [82, 83].

Chapter 4
Synthesizing Grasp Families from Prototypes

Grasp contact point synthesis
Palm pose synthesis
Hand joint configuration synthesis
Hand motion planning and control

This chapter deals with the construction of contact-level grasp families, which are represented as independent contact regions on the discretized surface of a target object. The computation is based on an available prototype grasp and an appropriate task specification. Contributions include an in-depth analysis about the geometric relations in the context of independent contact regions and the development of efficient parallelizable algorithms to compute these regions using convex optimization techniques. Furthermore, compared to existing approaches, the dependence on the necessary initial grasp is loosened for special cases. This allows to prioritize desired fingers/regions and to produce ICR which are shaped to befit the considered application.

4.1 Introduction

An important goal of the robot grasp selection process is to choose contact points which are suitable for the application at hand. Here, evaluating the "goodness" of a given multi-fingered grasp while accounting for the capabilities of the grasping device is an important issue and there have been many quality measures proposed in the literature (see [84] for a survey). For a large class of grasps the *force closure* property is desirable. Loosely speaking, force closure means the ability of the grasp to immobilize the grasped object influenced by an

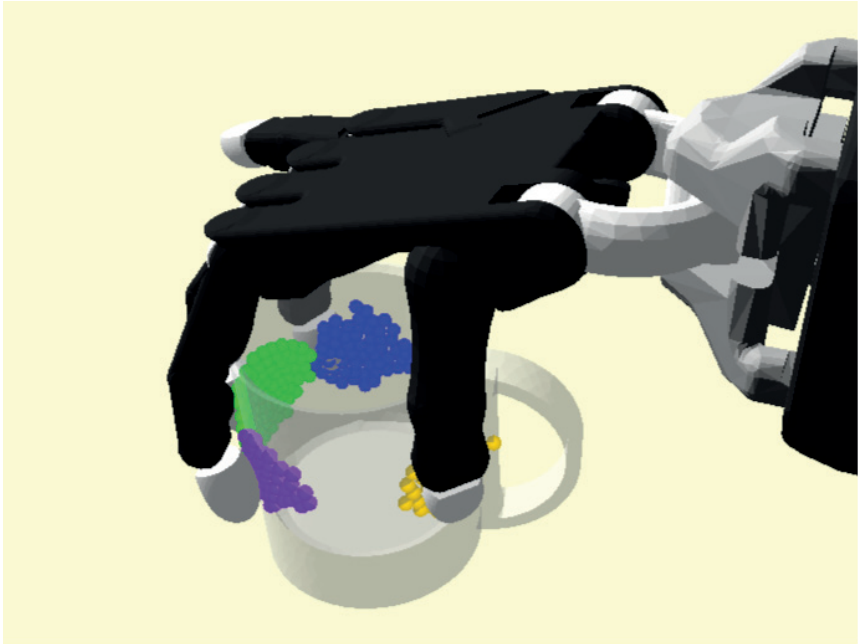


Figure 4.1: *Exemplary Contact Regions*: Shown are the ICR constructed from a 4-fingered prototype grasp. The grasp is provided by a human demonstrator via teleoperation in a virtual environment as described in Section 5.3.2.

arbitrary external disturbance, if the manipulator is capable of exerting sufficiently large forces through frictional contacts on the object [85]. Reuleaux [86] coined the term *form closure* for the related ability of a grasping/fixturing device to fully prevent motions of an object via unilateral frictionless contact constraints. For analysis purposes, contact force vectors and resulting torque vectors are commonly concatenated to wrench vectors. Mishra et al. [87] showed that a grasp is force/form closure, if the convex hull spanned by the contact wrenches stemming from bounded contact forces contains a neighborhood of the origin. The wrench set described by this convex hull is commonly referred to as the Grasp Wrench Space (GWS).

In many cases force closure is just a necessary, and not a sufficient requirement - a good grasp should be task oriented and be able to efficiently withstand forces, which are likely to occur during the performed task as stated by Li and Sastry [88]. In their work, the problem of incorporating knowledge of a task into the grasp analysis is addressed by formulating an ellipsoidal wrench set in order to describe probable disturbances. However, they conclude that it is difficult to model the ellipsoid with regard to specific tasks. If nothing about the task is known, a common measure is the radius of the largest origin-centered

insphere of the GWS, which was proposed by Kirkpatrick et al. [89]. Several works have integrated disturbance forces acting on the target object in the grasp evaluation [38, 90, 30].

From the viewpoint of a grasping device, not only the ability to resist disturbances, but also the robustness of a grasp to modeling, perception and positioning inaccuracies is an important factor. In this chapter, we present algorithms constructing ICR, which are designed to cope with uncertainties, while taking pre-specified disturbances into account. An illustrative example is shown in Fig. 4.1. To overcome the combinatorial nature of the problem, our computation schemes are centered around affine transformations of the GWS corresponding to a provided initial prototype grasp as discussed in Section 4.6.2. This prototype grasp can be computed by one of the contact-level planners discussed in Section 3.1, or be provided in form of a human demonstration. Demonstrated grasps usually are of high quality and naturally incorporate task specific constraints. Furthermore, anthropomorphic robotic hands are, at least to some extent, designed to replicate such grasps. Here, we do not account for the problem of finding appropriate palm poses and hand joint configurations corresponding to the computed regions. In the context of ICR, these problems have been addressed in [91]. Roa et al. [92] provide a solution to the related problem of ensuring that the regions are reachable by a given grasping device.

The rest of this chapter is organized as follows: below we state the tackled problem and assumptions before detailing the utilized contact models in Section 4.3. The presented framework builds upon the geometric relations regarding the grasp and task formulations which we discuss in Section 4.4. In Section 4.5 we introduce an extension of the GWS to families of grasps before we proceed to describe the developed ICR computation schemes in Section 4.6. Finally, we draw conclusions in Section 4.7.

4.2 Problem Description and Assumptions

Nomenclature

	Indices
c	Contact point index, $c \in \{1, \ldots, o\}$
i	Grasp contact index, $i \in \{1, \ldots, f\}$
j	Primitive wrench index, $j \in \{1, \ldots, l\}$
h	Hyperplane index, $h \in \{1, \ldots, u\}$
t	Sub-task index, $t \in \{1, \ldots, n\}$
	General
$\mathbf{p}_c$	Contact point on an object's surface, $c \in \{1, \ldots, o\}$
$\mathbf{n}_c$	Contact normal at $\mathbf{p}_c$, $c \in \{1, \ldots, o\}$
$\mathbf{f}_c$	Contact force at $\mathbf{p}_c$, $c \in \{1, \ldots, o\}$
$\boldsymbol{w}_j(\mathbf{p}_c)$	j-th primitive contact wrench at $\mathbf{p}_c$, $j \in \{1, \ldots, l\}$
k	Dimension of the wrench space, $k \in \{3, 6\}$

μ	Static friction coefficient, $\mu \geqslant 0$
ρ	Soft finger torque coefficient, $\rho \geqslant 0$
$(\mathbf{A}, \mathbf{b})$	$\mathcal{H}$-representation of a GWS, $\mathbf{A} \in \mathbb{R}^{u \times k}$, $\mathbf{b} \in \mathbb{R}^u$
$(\mathbf{E}, \mathbf{s})$	$\mathcal{H}$-representation of an EWS, $\mathbf{E} \in \mathbb{R}^{u \times k}$, $\mathbf{s} \in \mathbb{R}^u$
	Sets
$\mathcal{O}$	Contact point index set, $\mathcal{O} = \{1, \ldots, o\}$
$\mathcal{G}$	Contact-level grasp, $\mathcal{G} = \{g_i \in \mathcal{O} : i = 1, \ldots, f\}$
$\mathcal{R}_i$	Index set of contacts forming the i-th ICR, $\mathcal{R}_i \subseteq \mathcal{O}$
$\mathcal{W}_c$	Set of primitive wrenches generated at $\mathbf{p}_c$
$\mathcal{W}_{g_i}$	Primitive wrench set generated at $\mathbf{p}_{g_i}$
$\mathcal{W}_{\mathcal{R}_i}$	Primitive wrench set corresponding to contacts in $\mathcal{R}_i$
$\mathcal{Z}_{i,j}$	Set formed by the indices of those facets of GWS_{init} which contain $\boldsymbol{w}_j(\mathbf{p}_{g_i})$
$\mathcal{S}_{i,j}$	Search zone associated to $\boldsymbol{w}_j(\mathbf{p}_{g_i})$
$\mathcal{V}_h$	Set of wrenches spanning the h-th facet of GWS_{init}
$\mathcal{V}_{h,i}$	Set of wrenches in $\mathcal{V}_h$ generated at $\mathbf{p}_{g_i}$
$\mathcal{U}_{h,i}$	Set of wrenches in $\mathcal{V}_h$ not generated at $\mathbf{p}_{g_i}$
$\mathcal{F}_h$	Finger indices associated to the h-th facet of GWS_{init}
$\mathcal{R}_{\mathcal{F}_h}$	Contacts in regions $\mathcal{R}_i$ corresponding to fingers in $\mathcal{F}_h$

The surface of a target object is assumed to be discretized and described as a polygonal mesh (or polygonal chain in the case of planar objects) with vertices $\mathbf{p}_c$ which, henceforth, are referred to as contact points. The contact point indices are collected in the set $\mathcal{O} = \{1, \ldots, o\}$. Each point $\mathbf{p}_c$ has an associated inward-pointing unit normal $\mathbf{n}_c$ and neighboring points, defined as the ones connected to $\mathbf{p}_c$ by an edge of the mesh. Thus, this representation can be seen as a graph, where nodes represent mesh vertices $\mathbf{p}_c$ and edges define the neighboring relation between these vertices.

A f-fingered contact-level grasp $\mathcal{G} = \{g_i \in \mathcal{O} : i = 1, \ldots, f\}$ and corresponding independent contact regions $\mathcal{R}_i \subseteq \mathcal{O}$ are specified in terms of contact point indices. For convenience, contact-level grasps are simply referred to as grasps in the rest of this chapter. A constructive definition of the regions $\mathcal{R}_i$ is stated in Equation (4.12) in Section 4.6.1. We define the notion of *viable* grasps as

Definition 4.2.1 (Viability)**.**
Viable grasps are force closure grasps which comprise one contact drawn from each region $\mathcal{R}_i$ and are suitable to resist expected disturbances (see Section 4.4.2).

It is assumed that the target object is sufficiently discretized to capture local curvature, ensuring that grasps with contacts on mesh facets spanned by the discrete points forming regions $\mathcal{R}_i$ are also viable grasps. Furthermore, we presume that quasi static conditions prevail.

The aim is to develop efficient algorithms for the synthesis of grasp families, represented as regions $\mathcal{R}_i$, based on user-input provided in form of an initial

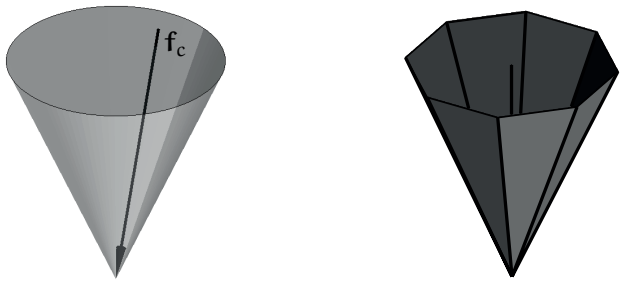


Figure 4.2: *Friction cone Discretization*: (a) To avoid slippage, contact forces $\mathbf{f}_c$ have to lie in the cone formalized in (4.1). (b) Approximation of the friction cone as a convex polytope resulting from an exemplary discretization with $l = 9$ forces.

viable grasp $\mathcal{G}_{\text{init}}$. Also, we want to investigate what is the set of disturbances that every viable grasp is guaranteed to resist, if desired regions $\mathcal{R}_i$ themselves are defined by a user beforehand.

4.3 Contact Modeling

We first consider *frictional point contacts* between the target object and the fingers of the gripping device. Static friction is taken into account via Coulomb's model, which states that slippage between two contacting surfaces does not occur if the following condition is satisfied

$$\|\left(\mathbf{I} - \mathbf{n}_c\mathbf{n}_c^T\right)\mathbf{f}_c\|_2 \leqslant \mu(\mathbf{n}_c^T\mathbf{f}_c), \tag{4.1}$$

where $\mathbf{f}_c$ is the contact force, $\mu \geqslant 0$ denotes the static friction coefficient and $\mathbf{I}$ is the (appropriately dimensioned) identity matrix. We discretize the nonlinear friction cone in (4.1) with l forces, such that one force acts along the contact point normal $\mathbf{n}_c$ and the remaining ones are distributed equidistantly around the cone's base as depicted in Fig. 4.2. The discretization forces at a contact point $\mathbf{p}_c$ are denoted in matrix notation as $\mathbf{F}_c = [\mathbf{f}_1(\mathbf{p}_c), \ldots, \mathbf{f}_l(\mathbf{p}_c)]$ which allows to express a contact force $\mathbf{f}_c$ as a conic combination of forces in $\mathbf{F}_c$ or formally $\mathbf{f}_c = \mathbf{F}_c\boldsymbol{\alpha}_c : \boldsymbol{\alpha}_c \geqslant \mathbf{0}$. The force $\mathbf{f}_c$ creates a torque $\boldsymbol{\tau}_c = (\mathbf{p}_c \times \mathbf{f}_c)$. Force and torque vectors can be concatenated to form a wrench

$$\boldsymbol{w}_c = \begin{bmatrix} \mathbf{f}_c \\ \boldsymbol{\tau}_c/\lambda \end{bmatrix} \in \mathbb{R}^k, \quad \lambda = \max_c(\|\mathbf{p}_c\|_2), \tag{4.2}$$

where the wrench dimension $k = 3$ for planar target objects and $k = 6$ in case of 3-dimensional objects. Dividing the torque parts by the largest possible torque arm λ guarantees scale invariance [38]. The wrenches generated by the

forces in $\mathbf{F}_c$ are referred to as *primitive wrenches*. For a given contact point $\mathbf{p}_c$, the set of primitive wrenches is defined as

$$\mathcal{W}_c = \{\boldsymbol{w}_1(\mathbf{p}_c), \cdots, \boldsymbol{w}_l(\mathbf{p}_c)\}. \tag{4.3}$$

The *soft finger contact* model according to [3] allows for additional torsional moments around the local contact normal $\mathbf{n}_c$. To this end, the set of primitive wrenches in 4.3 is augmented with the wrenches $\left[\mathbf{n}_c^T, \rho\mathbf{n}_c^T/\lambda\right]^T$ and $\left[\mathbf{n}_c^T, -\rho\mathbf{n}_c^T/\lambda\right]^T$, where $\rho \geqslant 0$ is a positive scalar. In the soft finger contact model, scaling the wrench vectors by the largest possible torque arm λ does not grant scale-invariance any more. This is due to the fact, that the additional wrenches do not depend on the object geometry. Still, scaling imparts invariance to the chosen units of length.

In the case of the *frictionless point contact* model, the friction coefficient μ is zero and $\mathbf{f}_c$ acts along the surface normal. In this case, the set $\mathcal{W}_c$ just contains one wrench generated by the respective normal force.

Form closure grasps on 3-dimensional objects require a minimum of seven frictionless contacts, since fewer than seven wrenches cannot positively span $\mathbb{R}^6$. For some objects with rotational symmetries it is not possible to achieve the form closure property with frictionless contacts. Regarding force closure grasps, considering the frictional hard- and soft finger contact models, a respective number of three and two contacts is always necessary. Some objects require four frictional hard-finger contacts which are always sufficient [93].

4.4 Physically Relevant Grasp and Task Modeling

In this section we want to address the following questions which, according to Borst et al. [30], are paramount in static grasp synthesis:

- What forces can a given grasp exert on the object?
- Which disturbances are expected to act on the object?
- How well can the grasp resist the expected disturbances?

In the scope of this thesis we adhere to the nomenclature established in the literature and refer to wrench sets such as the GWS as wrench spaces, although they are not actual vector spaces since the axioms of the identity element of addition and the inverse elements of addition do not necessarily hold.

4.4.1 The Grasp Wrench Space

With respect to the first of the above questions, the GWS describes the set of all wrenches generated by forces that can be applied to an object through a grasp

$\mathcal{G}$. Formally, the discrete GWS is defined as the polytope given by the convex hull over the primitive wrenches associated to all grasp contacts in $\mathcal{G}$

$$\mathrm{GWS} = \mathrm{conv}\Big(\{\boldsymbol{w} : \boldsymbol{w} \in \mathcal{W}_c,\ \forall c \in \mathcal{G}\}\Big). \tag{4.4}$$

The above definition implies that the summed magnitudes of the contact forces $\mathbf{f}_c$, which generate the wrenches in (4.2) forming the primitive wrench sets according to (4.3), is bounded. In this work, we consider a bound $\|\mathbf{f}_c\|_2 = g$ on the grasp contact forces. Since the applied forces are proportional to the current in the grasping device's actuators, this can be seen as a limitation due to a common power source and an appropriate value for g can be derived from the kinematic structure and torque limitations of the considered device [31]. It is worth noting that all primitive wrenches spanning a GWS lie on its boundary. However, not necessarily all of these wrenches constitute vertices of the GWS, *e. g.*, when a grasp $\mathcal{G}$ contains multiple points on the same planar surface, some of its associated wrenches may lie on ridges (see Section 2.1) of the corresponding GWS. This is considered in the implementations used to generate the results presented in Chapter 5. In the following derivations however, to avoid notational clutter and without loss of generality, we assume that all primitive wrenches in (4.4) constitute vertices of the GWS.

The related concept of the Object Wrench Space (OWS) was introduced by Pollard in [38]

$$\mathrm{OWS} = \mathrm{conv}\Big(\{\boldsymbol{w} : \boldsymbol{w} \in \mathcal{W}_c,\ \forall c \in \mathcal{O}\}\Big). \tag{4.5}$$

It is defined as the convex hull over the primitive wrenches associated to all points of the mesh describing the geometry of an object. As such, it describes the best possible grasp for a given object and each GWS is a subset of the OWS.

4.4.2 The Task Wrench Space

In the presented framework, tasks are represented as sets of wrenches necessary to counteract expected disturbance forces/torques acting on the object, *i. e.*, task wrenches are the mirror image of expected disturbance wrenches. Given n tasks $\mathcal{T}_t$, we denote the Task Wrench Space (TWS) as the convex hull over the Minkowski sum of these tasks

$$\mathrm{TWS} = \mathrm{conv}\Big(\{\mathcal{T}_1 \oplus \ldots \oplus \mathcal{T}_n\}\Big). \tag{4.6}$$

Similar to the formulation of the GWS in (4.4), in the above definition it is assumed that the sum magnitude of the forces generating the wrenches forming a sub-task $\mathcal{T}_t$ is bounded.

If the exact task specification is unknown, it is reasonable to assume that the grasp at least should be able to withstand the disturbances introduced by

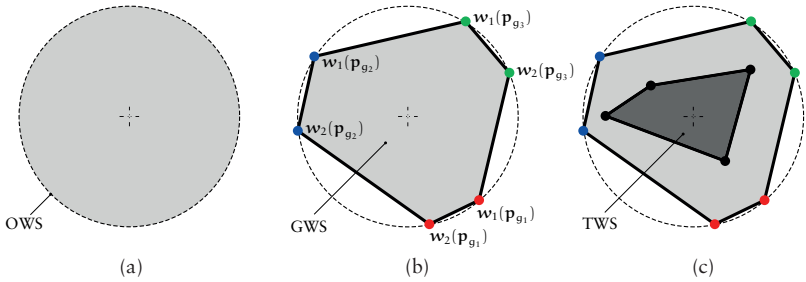


Figure 4.3: *Wrench Spaces*: Illustrated is an abstract 2-dimensional wrench space. (a) The OWS is formed by the convex hull over all primitive wrenches that can be applied to the object through grasp contacts. (b) Shown is a GWS corresponding to a three-fingered grasp with friction cone discretization $l = 2$, i. e., two primitive wrenches are generated at each grasp contact point $\mathbf{p}_{g_i}$. (c) To fulfill the task requirements, the grasp has to be able to apply the task wrenches, i. e., the GWS has to contain the TWS.

gravity and contacts between the object and the environment [30]. A popular way of representing these requirements is to formulate the TWS as the largest origin-centered insphere of the GWS corresponding to the considered grasp. This sphere contains the largest wrench the grasp is able to counteract in any direction and relates to the grasp quality criterion proposed by Kirkpatrick et al. [89]. Yet, this gives only weak protection against disturbance forces on the extreme parts of the object geometry and might pose unnecessary restrictions by protecting against disturbances which are unlikely to occur. This is due to the fact that torques are typically caused by forces acting on the boundary of the object and therefore a general TWS is not uniform [30]. Pollard [38] suggested a physically better motivated way to describe a task $\mathcal{T}_t$ by wrenches resulting from any possible disturbance of magnitude d that can act on any point $\mathbf{p}_c$ on the objects surface. This way of modeling a task incorporates the object geometry in the grasp evaluation and is equivalent to mirroring and scaling the OWS in (4.5) and we denote the corresponding set of wrenches as $\mathcal{T}^{\mathcal{O}}$.

Combining multiple independent sub-tasks usually involves the computationally expensive Minkowski sum according to Equation (4.6). However, Borst et al. [30] have shown that disturbances caused by gravitational forces of any direction acting in the object's Center of Mass (CoM) with magnitude m can be easily incorporated in $\mathcal{T}^{\mathcal{O}}$ if the CoM is used as wrench origin. The resulting gravitational wrenches have zero torque parts and, as well as $\mathcal{T}^{\mathcal{O}}$, are tightly enclosed by a sphere in the force domain. Thus, it is possible to simply scale the force domain of $\mathcal{T}^{\mathcal{O}}$ by a factor $1 + m/d$ in order to account for the combined effects of forces corresponding to surface disturbances and gravity.

An example of the wrench sets constituting the OWS, GWS and TWS is shown in Fig. 4.3. For simplicity, a hypothetical 2-dimensional illustration of

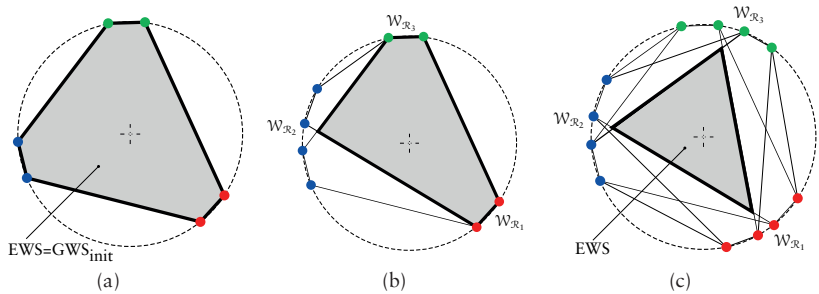


Figure 4.4: *The Exertable Wrench Space:* (a) The EWS equals the GWS_{init} since each region $\mathcal{R}_i$ only contains the initial grasp contacts in $\mathcal{G}_{\text{init}}$. (b) Adding new points to region $\mathcal{R}_2$, i. e., adding new primitive wrenches to its corresponding set $\mathcal{W}_{\mathcal{R}_2}$ according to (4.9) allows for new grasps to be formed by combinations of contact points in the independent regions. (c) The EWS is defined as the intersection of all grasp wrench spaces corresponding to viable grasps, facets of the EWS are coplanar to facets of the limiting GWS.

wrenches is adopted, although they are 6-dimensional in the general case and 3-dimensional for planar grasps.

4.5 The Exertable Wrench Space

Here, we introduce a generalization of the GWS which is associated to a single grasp, to the Exertable Wrench Space (EWS) for a family of grasps. Let us for the moment assume that a family of grasps is already defined in terms of f ICR. If the sets $\mathcal{R}_1, \ldots, \mathcal{R}_f$ forming these regions are are disjoint, the number of viable grasps ν according to Definition 4.2.1 is given by the product of the cardinalities of $\mathcal{R}_i$

$$\nu = |\mathcal{R}_1| \times \ldots \times |\mathcal{R}_f|. \tag{4.7}$$

In the case of partially overlapping sets $\mathcal{R}_i$, which is admissible in the scope of this work, (4.7) gives an upper bound. This allows for a formal definition of the EWS as the intersection of all grasp wrench spaces corresponding to viable grasps.

$$\text{EWS} = \left(\text{GWS}_1 \cap \ldots \cap \text{GWS}_\nu\right). \tag{4.8}$$

Analogue to the GWS, which is composed of the wrenches a *single* grasp can exert, the EWS represents the set of wrenches which *every* viable grasp can apply to the target object. Figure 4.4 shows the relation between the EWS and the contact points in the ICR. Let us denote the wrenches associated with the contacts in $\mathcal{R}_i$ as

$$\mathcal{W}_{\mathcal{R}_i} = \{\boldsymbol{w} : \boldsymbol{w} \in \mathcal{W}_c,\ \forall c \in \mathcal{R}_i\}. \tag{4.9}$$

Every point added to a region $\mathcal{R}_i$ contributes wrenches to its associated set $\mathcal{W}_{\mathcal{R}_i}$ in (4.9) and limits the EWS.

The mirror image of the EWS provides an exact formulation of the disturbance wrenches each of the viable grasps is guaranteed to withstand. However, from Equations (4.7) and (4.8) it follows that a brute-force approach via computing and intersecting the convex hulls of all possible grasps is prohibitive in the general case, since it requires the computation and intersection of ν convex hulls.

4.6 Computation of Independent Contact Regions

Now we consider the problem of computing ICR when given a viable example grasp $\mathcal{G}_{\text{init}}$ and a task specification in form of a TWS as discussed in Section 4.4.2. In the light of the above, it is evident that these regions need to be constructed such that the corresponding EWS contains the TWS, i. e., each viable grasp needs to be able to at least exert the task wrenches. Thus, the goal is to approximate the EWS and compute regions $\mathcal{R}_i$ which fulfil this requirement.

Let the $\mathcal{H}$-representation of the prototype's GWS in (4.4) be given as $(\mathbf{A}, \mathbf{b})$, where $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_u]^T \in \mathbb{R}^{u \times k}$ is a matrix containing the inward-pointing unit normals to the bounding hyperplanes. Vector $\mathbf{b} = [b_1, \ldots, b_u] \in \mathbb{R}^u$ contains the corresponding distances to the origin (see Section 2.1). Analogously, the matrix $\mathbf{E} = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_u]^T \in \mathbb{R}^{u \times k}$ and the vector $\mathbf{s} = [s_1, \ldots, s_u] \in \mathbb{R}^u$ describe the $\mathcal{H}$-representation of the EWS.

4.6.1 Exploiting Task Redundancy

By definition, the GWS of a viable prototype grasp contains the TWS of interest. Therefore, grasp $\mathcal{G}_{\text{init}}$ is redundant with respect to the task. The idea is to trade-off this redundancy for the possibility to generate a family of viable grasps, such that each member of this family still respects the task requirements [15]. This trade-off can be formalized with an affine transformation of the GWS associated with $\mathcal{G}_{\text{init}}$, i. e., an approximation of the EWS by a new polytope related to the GWS. The process of transforming the GWS leads to the definition of zones in the wrench space. Pollard [38, 39] provided the key idea of "populating" independent regions $\mathcal{R}_i$ with those contacts which contribute wrenches lying in these search zones. Here, we interpret the underlying principle in the context of *visibility*, which is a well known concept in the field of computational geometry [23].

Consider a convex polytope $\mathcal{P}$. By convexity, $\mathcal{P}$ is fully contained in every closed inner half-space in (2.2) which is defined by a hyperplane corresponding to any of its facets. A facet h is said to belong to the visible region of a point $\hat{\mathbf{x}}$, if that point lies in the outer half-space in (2.3) which is defined by the hyperplane $\mathcal{H}_h$ supporting that facet, i. e., $\hat{\mathbf{x}}$ "sees" facet h.

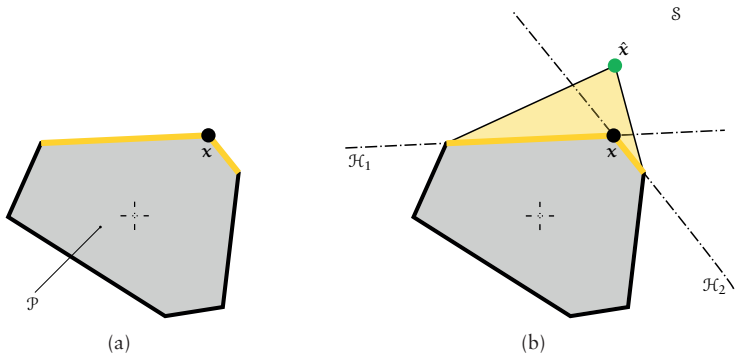


Figure 4.5: *Visible Region*: (a) Depicted is a polytope $\mathcal{P}$, facets which are seen from vertex $\mathbf{x}$ are highlighted in yellow. (b) Zone $\mathcal{S}$ is formed by the intersection of closed outer half-spaces defined by $\mathcal{H}_1$ and $\mathcal{H}_2$ according to (2.3). Since point $\hat{\mathbf{x}}$ lies in $\mathcal{S}$, the polytope $\{\mathcal{P} \setminus \mathbf{x}, \hat{\mathbf{x}}\}$ fully contains $\mathcal{P}$ and thus $\hat{\mathbf{x}}$ can safely substitute $\mathbf{x}$.

Proposition 4.1 (Visibility)**.**
(a) $\mathcal{P} \subseteq \{\mathcal{P} \setminus \mathbf{x}, \hat{\mathbf{x}}\}$ if all facets which are visible from $\mathbf{x}$ are also visible from $\hat{\mathbf{x}}$.
(b) The convex hull over multiple sets containing $\mathcal{P}$ also contains $\mathcal{P}$.

Proposition 4.1(a) is illustrated in Fig. 4.5. It states, that the polytope resulting from replacing a vertex $\mathbf{x}$ of $\mathcal{P}$ with a point $\hat{\mathbf{x}}$ will fully contain $\mathcal{P}$, if the visible region of $\mathbf{x}$ on $\mathcal{P}$ is seen by $\hat{\mathbf{x}}$ as well. Let zone $\mathcal{S}$ be the intersection of all outer half-spaces defined by hyperplanes corresponding to facets which contain vertex $\mathbf{x}$ of $\mathcal{P}$. An equivalent interpretation of 4.1(a) is, that $\mathcal{P}$ is preserved if substitution point $\hat{\mathbf{x}}$ lies in zone $\mathcal{S}$. Note that it is possible for $\hat{\mathbf{x}}$ to see more facets of $\mathcal{P}$ than $\mathbf{x}$. Proposition 4.1(b) is a direct consequence of convexity.

In this light, the requirements for a point on a target object's surface to be included in one of the independent regions $\mathcal{R}_i$ are illustrated in Fig. 4.6. The core concept is based on finding substitutes for wrenches inside their respective search zones, such that the EWS which contains the task is preserved. To formalize the notion of search zones, we need to define f l sets holding the indices of those facets of the prototype grasp's GWS which contain the j-th primitive wrench generated at the i-th grasp contact

$$\mathcal{Z}_{i,j} = \left\{ h \in \mathbb{Z}_+ : \mathbf{a}_h^T \boldsymbol{w}_j(\mathbf{p}_{g_i}) + b_h = 0 \right\}. \tag{4.10}$$

Assuming that an EWS is available (methods for appropriately approximating the EWS under consideration of given tasks will be discussed in Section 4.6.2), the definition in (4.10) allows to describe search zones in the wrench space

$$\mathcal{S}_{i,j} = \left\{ \boldsymbol{w} \in \mathbb{R}^k : \boldsymbol{e}_h^T \boldsymbol{w} + s_h \leqslant 0, \ \ \forall h \in \mathcal{Z}_{i,j} \right\}. \tag{4.11}$$

Here, each primitive wrench spanning the prototype grasp's GWS is associated

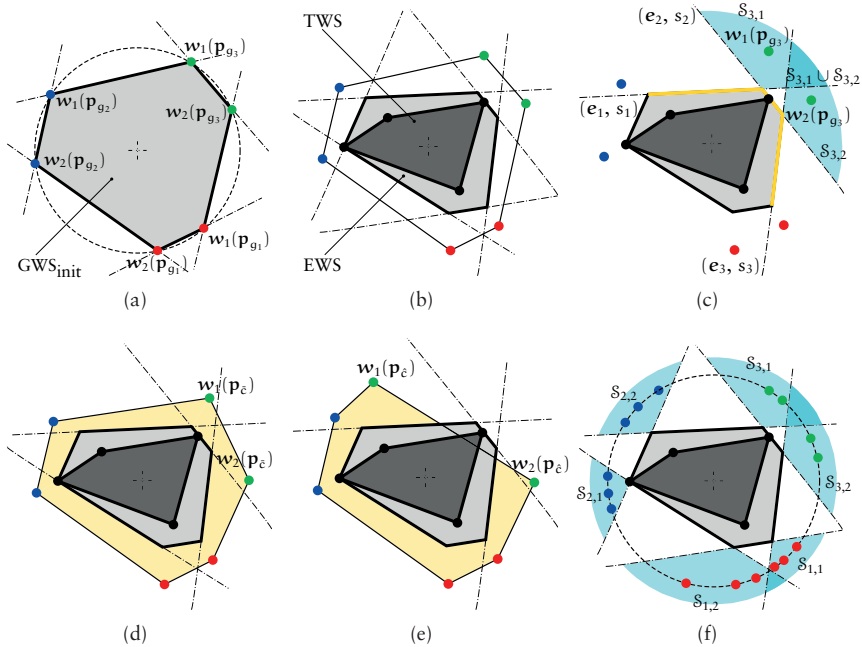


Figure 4.6: *ICR Construction*: Shown is the ICR construction procedure in an abstract 2-dimensional wrench space for a 3-fingered grasp with friction cone discretization $l = 2$. (a) The prototype grasp's GWS and it's confining hyperplanes. (b) The light shaded area illustrates an EWS approximation computed as an affine transformation of the prototype's GWS as discussed in Section 4.6.2. The EWS contains the provided TWS which is signified by the dark shaded region. (c) The union of EWS facets which are visible from $\boldsymbol{w}_1(\mathbf{p}_{g_1})$ and $\boldsymbol{w}_2(\mathbf{p}_{g_1})$ is highlighted in yellow. Corresponding search zones are formed according to (4.11). According to (2.3), $\mathcal{S}_{3,1}$ is the intersection of the closed outer half-spaces of EWS planes $(\boldsymbol{e}_1, s_1)$ and $(\boldsymbol{e}_2, s_2)$, $\mathcal{S}_{3,2}$ contains the intersection of closed outer half-spaces associated to $(\boldsymbol{e}_2, s_2)$ and $(\boldsymbol{e}_3, s_3)$. (d) The same EWS facets are visible from the wrenches generated at $\mathbf{p}_{\bar{c}}$ and $\mathbf{p}_{g_1}$. Thus, as a consequence of Proposition 4.1, contact point $\mathbf{p}_{\bar{c}}$ can safely replace $\mathbf{p}_{g_1}$ in a grasp and $\bar{c}$ qualifies for inclusion in region $\mathcal{R}_1$. (e) Not all EWS facets which are visible from the wrenches generated at $\mathbf{p}_{g_1}$ are seen by the wrenches corresponding to $\mathbf{p}_{\hat{c}}$. Therefore, $\hat{c}$ does not qualify for inclusion in $\mathcal{R}_1$. (f) The cyan shaded areas illustrate the search zones $\mathcal{S}_{i,j}$ corresponding to all prototype grasp contacts g_i. As stated in (4.12), regions $\mathcal{R}_i$ are formed by contacts which can contribute convex combinations of primitive wrenches lying in search zones $\mathcal{S}_{i,1}$ and $\mathcal{S}_{i,2}$ (or the intersection of these zones).

with one search zone $\mathcal{S}_{i,j}$ in (4.11). With reference to Fig. 4.6(f), note that any wrench lying inside $\mathcal{S}_{i,j}$ can substitute a prototype grasp contact wrench $\boldsymbol{w}_j(\mathbf{p}_{g_i})$ which results in a GWS that still contains the EWS and thus the TWS.

Algorithm 1: *ICRComputation* - Full

Input: EWS approximation $(\mathbf{E}, \mathbf{s})$, search zone index sets $\{\mathcal{Z}_{i,j}\}$, target object $\{\mathbf{p}_c, \mathbf{n}_c\}$
Output: set of f regions $\mathcal{R}_i$

```
for c ← 1 to o do                                   /* Loop over all contact points of the object */
    for i ← 1 to f do                                /* Loop over all f fingers */
        if InclusionTest(c, {Z_{i,1}, ..., Z_{i,l}}, EWS, target object) then
            R_i ← c                                  /* c qualifies as a member of R_i */
return {R_1, ..., R_f}
```

Algorithm 2: *ICRComputation* - BFS

Input: EWS approximation $(\mathbf{E}, \mathbf{s})$, prototype grasp $\mathcal{G}_{init}$, search zone index sets $\{\mathcal{Z}_{i,j}\}$, target object $\{\mathbf{p}_c, \mathbf{n}_c\}$
Output: set of f regions $\mathcal{R}_i$

```
queue ← ∅
for i ← 1 to f do                                   /* i.e., for each contact point g_i ∈ G_init */
    R_i ← ∅
    R_i ← g_i   /* Prototype grasp contact g_i qualifies by definition as member of R_i */
    queue ← push neighbors of g_i                   /* Enqueue the neighbors of contact g_i */
    while queue ≠ ∅ do
        c ← pop queue                               /* Dequeue the first element in the list */
        if InclusionTest(c, {Z_{i,1}, ..., Z_{i,l}}, EWS, target object) then
            R_i ← c                                 /* c qualifies as a member of R_i */
            queue ← push neighbors of c             /* Enqueue the neighbors of contact c */
return {R_1, ..., R_f}
```

This observation motivates the following Proposition for qualifying contacts as ICR members.

Proposition 4.2 (Inclusion condition - CC)**.**
Index $c \in \mathcal{O}$ *is included in region* $\mathcal{R}_i$ *if there exist convex combinations of corresponding primitive wrenches* $\mathcal{W}_c$ *lying in each of the* l *search zones* $\mathcal{S}_{i,j}$ *associated with* g_i**.**

We can use the above Proposition to state a formal definition of the regions $\mathcal{R}_i$ which form ICR

$$\mathcal{R}_i = \big\{ c \in \mathcal{O} : \exists \boldsymbol{\alpha}_j \in \mathbb{R}^l \text{ s.t. } \boldsymbol{W}_c \boldsymbol{\alpha}_j \in \mathcal{S}_{i,j}, \quad j = 1, \ldots, l, \;\; \boldsymbol{\alpha}_j \geqslant 0, \;\; \|\boldsymbol{\alpha}_j\|_1 = 1 \big\}. \tag{4.12}$$

Evaluating the inclusion condition in Proposition 4.2 requires the solution of a feasibility problem which can be formulated as a Linear Program (LP) [39]. In [41], Roa and Suárez formulated a computationally cheaper but less stringent condition. They only check the discrete primitive wrenches associated with a contact point for inclusion in the corresponding search zones which leads to the following Corollary

Algorithm 3: *InclusionTest* - CC

Input: contact c to be tested for inclusion in $\mathcal{R}_i$, search zone index sets $\{\mathcal{Z}_{i,1}, \ldots, \mathcal{Z}_{i,l}\}$, EWS approximation $(\mathbf{E}, \mathbf{s})$, target object $\{\mathbf{p}_c, \mathbf{n}_c\}$
Output: boolean indicating success or failure

/* Check for convex combinations of wrenches in $\mathcal{W}_c$ lying in each search zone */
for $j \leftarrow 1$ *to* l **do**
  Solve the following linear program:

$$\begin{aligned} &\underset{\boldsymbol{\alpha}_j \in \mathbb{R}^l,\, z \in \mathbb{R}}{\text{minimize}} && z \\ &\text{subject to} && \mathbf{e}_h^T \boldsymbol{W}_c \boldsymbol{\alpha}_j + s_h \leqslant z, \quad \forall h \in \mathcal{Z}_{i,j} \\ &&& \|\boldsymbol{\alpha}_j\|_1 = 1, \quad \boldsymbol{\alpha}_j \geqslant \mathbf{0}, \quad z \leqslant 0 \end{aligned}$$

  if $z > 0$ **then**
    return false /* Test for inclusion in $\mathcal{R}_i$ has failed */
return true /* Test for inclusion in $\mathcal{R}_i$ has succeeded */

Corollary 4.1 (Inclusion condition - PW)**.**
Index $c \in \mathcal{O}$ *is included in region* $\mathcal{R}_i$ *if there exist primitive wrenches in* $\mathcal{W}_c$ *lying in each of the* l *search zones* $\mathcal{S}_{i,j}$ *associated with* g_i*.*

To summarize, the ICR construction procedure consists of approximating the EWS (see Section 4.6.2) under consideration of a given TWS, forming $f\,l$ search zones according to (4.11) and sequentially test contact points for inclusion in regions $\mathcal{R}_i$ according to either Proposition 4.2 or Corollary 4.1. In Algorithm 1, all o contacts are checked for inclusion in every region $\mathcal{R}_i$, which might lead to non-neighboring points to be included in the same region $\mathcal{R}_i$. Recall that the target object is represented as a graph. Therefore, for each region $\mathcal{R}_i$, it is alternatively possible to explore contact points via a simple breadth-first tree search with the respective initial grasp contact point $\mathbf{p}_{g_i}$ as root node [41] as shown in Algorithm 2. Both algorithms require time $O(o\,f)$, can be conducted independently for each finger and thus are parallelizable . The contact point inclusion test routines according to Proposition 4.2 and Corollary 4.1 are represented in Algorithm 3 and Algorithm 4 respectively. In Algorithm 3, up to l linear programs need to be solved which can be done in polynomial time. The inclusion test according to Algorithm 4 requires time $O(l^2)$.

Constructing ICR by finding wrenches in the intersection of closed outer half-spaces forming the search zones in (4.11) and mapping them back to contact points yields a family of grasps which is similar to the provided prototype as stated in [39]. Here, we state an explicit definition of similarity opposed to the informal one given in [39]

Definition 4.6.1 (Similarity)**.**
Two grasps are similar if the face lattices of their corresponding grasp wrench spaces are isomorphic.

Algorithm 4: *InclusionTest* - PW

Input: contact c to be tested for inclusion in $\mathcal{R}_i$, search zone index sets $\{\mathcal{Z}_{i,1}, \ldots, \mathcal{Z}_{i,l}\}$, EWS approximation $(\mathbf{E}, \mathbf{s})$, target object $\{\mathbf{p}_c, \mathbf{n}_c\}$
Output: boolean indicating success or failure

```
/* Check if there exist wrenches w_j(p_c) lying in each search zone */
for q ← 1 to l do
    success ← false
    for j ← 1 to l do
        /* Check if the j-th wrench at contact c lies in the q-th search zone S_{i,q} */
        r ← max({e_h^T w_j(p_c) + s_h, ∀h ∈ Z_{i,q}})
        if r ⩽ 0 then
            success ← true
            break
    if ¬success then
        return false                    /* Test for inclusion in R_i has failed */
return true                             /* Test for inclusion in R_i has succeeded */
```

Loosely speaking, the face lattice of a convex polytope describes its topology as discussed in Section 2.1. Thus, the above definition implies that the GWS of similar grasps comprise the same topological structure in a sense that there exists a homeomorphism between their faces. A more detailed discussion can be found in [22].

4.6.2 Affine Transformations to Approximate the EWS

So far, we have postponed the issue of how to approximate the EWS when given a prototype grasp and a TWS to preserve. This is a prerequisite for the Algorithms discussed in the previous section and we address this problem by deriving appropriate affine transformations of the prototype grasp's GWS.

Parallel Shifting

A straightforward way of performing this transformation was introduced by Pollard [38, 39]. In her work, the hyperplanes defining the GWS of a prototype are parallely shifted inwards until they are tangent to the TWS. Here, we formulate the shifting procedure as a LP which minimizes the offset s_h of the h-th hyperplane

$$\begin{aligned} &\underset{s_h \in \mathbb{R}_+}{\text{minimize}} \quad s_h \\ &\text{subject to} \quad \mathbf{T}^T \boldsymbol{e}_h + \mathbb{1} s_h \geqslant \mathbf{0}, \end{aligned} \tag{4.14}$$

where $\mathbf{T} \in \mathbb{R}^{k \times z}$ is a column matrix of the z task wrenches contained in the TWS in (4.6) and $\mathbb{1}$ denotes an appropriately dimensioned column vector of ones. This transformation leaves the GWS facet normals unchanged, i. e.,

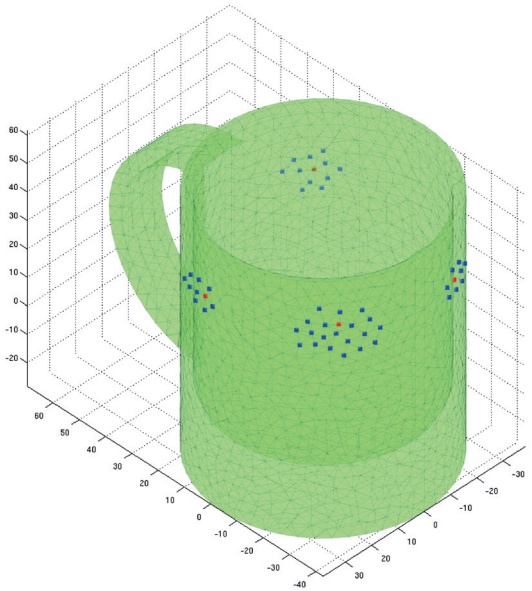


Figure 4.7: *ICR Example - Parallel Shifting*: Exemplary ICR construction for a 4-fingered prototype grasp based on an EWS approximation by parallel shifting. Red squares represent the prototype's grasp contact points, blue squares the regions $\mathcal{R}_i$. The regions were constructed considering frictional hard-finger contacts. The illustrated example incorporates a TWS formed by a scaled task $\mathcal{T}^O$ to account for the combined effects of gravity and disturbances acting on the object's surface as described in Section 4.4.2. Utilized parameters: $\mu = 0.8$, $l = 8$, $g = 10$, $m = 1.5$, $d = 0.6$.

Algorithm 5: *EWSApproximation* - Parallel Shift

Input: prototype grasp $\mathcal{G}_{\text{init}}$, task wrenches $\mathbf{T}$, target object $\{\mathbf{p}_c, \mathbf{n}_c\}$
Output: EWS approximation $(\mathbf{E}, \mathbf{s})$

Compute the GWS of $\mathcal{G}_{\text{init}}$ in $\mathcal{H}$-representation $(\mathbf{A}, \mathbf{b})$ according to (4.4)
$\mathbf{E} \leftarrow \mathbf{A}$ /* Parallel shifting does not affect the facet normals */
for $h \leftarrow 1$ *to* u **do** /* Loop over all u facets of the GWS */
  Compute the shifted offset s_h by solving the LP in (4.14)
return $(\mathbf{E}, \mathbf{s})$

$\boldsymbol{e}_h = \boldsymbol{a}_h$, $h = 1, \ldots, u$. Note that (4.14) allows to omit forming the convex hull over the task wrenches in (4.6). The procedure of approximating the EWS by parallel shifting is summarized in Algorithm 5. It requires the computation of a convex hull over fl primitive wrenches. Considering the QuickHull algorithm [94], this step has a complexity of $O\left((fl)^3/6\right)$ for six-dimensional input. According to [21], an upper bound for the number of facets of the resulting

GWS can be given as $u \leqslant (f\,l)^3/6$. Subsequently for each of the u facets the LP in (4.14) needs to be solved, which can be done in polynomial time. In order to incorporate a spherical TWS as discussed in Section 4.4.2, the iterative solution of the LP in Algorithm 5 can be omitted, since the offset s_h in (4.14) equal the radius of the considered sphere. An example of constructing ICR considering an EWS approximated via parallel shifting is shown in Fig. 4.7.

Prioritizing Fingers

Using parallel shifting to approximate the confining hyperplanes of the EWS and forming associated search spaces provides a computationally efficient way of generating ICR. However, it is by no means unique. The only requirement is that the intersection of inner half-spaces associated with the EWS hyperplanes forming search spaces contains the TWS. This requirement is fulfilled by infinitely many hyperplanes if the constraint of parallelism is relaxed and inclination of these hyperplanes is allowed.

In this line of thought, we propose a method to prioritize single fingers in the ICR construction procedure. Given again a prototype grasp $\mathcal{G}_{\text{init}}$ and a TWS, the goal is to enlarge the region $\mathcal{R}_i$ corresponding to a specific finger by reducing the number of points in the remaining $f-1$ regions. The key idea is to approximate the EWS by "tilting" those facets of the prototype's GWS which contain wrenches corresponding to the prioritized finger until they are tangent to the TWS. To this end, for each of the h facets of the prototype's GWS, we formulate the set of primitive wrenches spanning the h-th facet

$$\mathcal{V}_h = \left\{ \boldsymbol{w} \in \{\mathcal{W}_{g_1}, \ldots, \mathcal{W}_{g_f}\} : \mathbf{a}_h^T \boldsymbol{w} + b_h = 0 \right\}. \tag{4.15}$$

Due to the prerequisite that polytopes are represented in simplicial form, the set in (4.15) contains k primitive wrenches. Those vertices of the h-th facet which correspond to finger i form the set

$$\mathcal{V}_{h,i} = \left\{ \boldsymbol{w} \in \mathcal{W}_{g_i} : \mathbf{a}_h^T \boldsymbol{w} + b_h = 0 \right\}. \tag{4.16}$$

Finally, Equations (4.15) and (4.16) allow to define the set

$$\mathcal{U}_{h,i} = \mathcal{V}_h \setminus \mathcal{V}_{h,i}. \tag{4.17}$$

It contains those primitive wrenches which are vertices of the h-th facet but are not associated to finger i.

The core concept of the tilting procedure is illustrated in Fig. 4.8. Here, the EWS is approximated by the maximum-margin separating hyperplanes between wrenches in the TWS and elements in the sets $\mathcal{V}_{h,i}$ in (4.16). An efficient solution to this problem is provided in the framework of Support Vector Machines (SVM) [95]. It requires solving the Quadratic Programing (QP) problem stated below

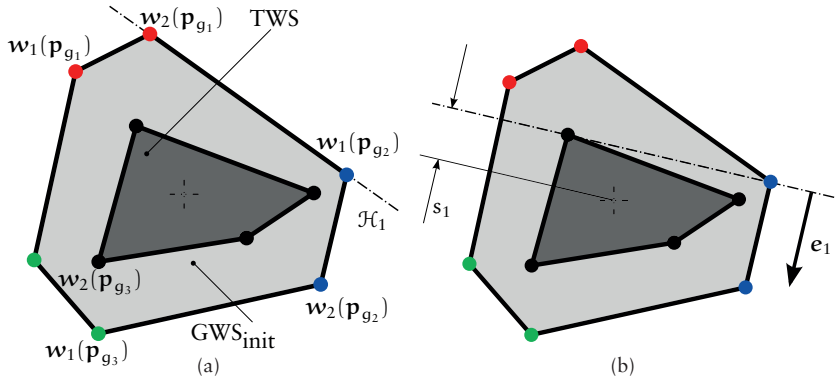


Figure 4.8: *EWS Approximation - Prioritize Finger*: Illustrated is the exemplary construction of one approximating hyperplane of the EWS with finger $i = 1$ prioritized. (a) Shown are the prototype grasp's GWS, the considered TWS and hyperplane $\mathcal{H}_1$ which defines a facet containing a primitive wrench associated with grasp contact point p_{g_1}. (b) The EWS hyperplane according to (4.18) and (4.19). It separates the wrenches in the TWS from $\mathcal{V}_{1,1} = w_2(p_{g_1})$ in (4.16) with maximum margin while pivoting around $\mathcal{U}_{1,1} = w_1(p_{g_2})$ in (4.17).

$$\begin{aligned} \underset{e'_h \in \mathbb{R}^k,\, s'_h \in \mathbb{R}_+}{\text{minimize}} \quad & \frac{1}{2}\|e'_h\|_2^2 \\ \text{subject to} \quad & \\ -V_{h,i}^T e'_h - \mathbb{1} s'_h &\geqslant \mathbb{1}, \\ T^T e'_h + \mathbb{1} s'_h &\geqslant 0, \\ U_{h,i}^T e'_h + \mathbb{1} s'_h &= 0. \end{aligned} \tag{4.18}$$

In the above problem, the constraint column matrices $V_{h,i} \in \mathbb{R}^{k \times |\mathcal{V}_{h,i}|}$ and $U_{h,i} \in \mathbb{R}^{k \times |\mathcal{U}_{h,i}|}$ contain the elements of the respective sets according to (4.16) and (4.17), matrix $T \in \mathbb{R}^{k \times z}$ holds the task wrenches in the TWS and $\mathbb{1}$ denotes appropriately dimensioned column vectors. Vector e'_h and scalar s'_h denote the normal and offset of the h-th approximated EWS facet scaled by the norm of e'_h, i. e.,

$$e_h = \frac{e'_h}{\|e'_h\|_2}, \quad s_h = \frac{s'_h}{\|e'_h\|_2}. \tag{4.19}$$

Algorithm 6 summarizes the EWS approximation procedure. As in Algorithm 5, the computation of a convex hull over the primitive grasp wrenches is necessary. Subsequently, for all u facets of this GWS, a convex QP needs to be solved which requires polynomial time. Approximating the h-th EWS facet via the maximum-margin hyperplane according to (4.18) and (4.19) not neces-

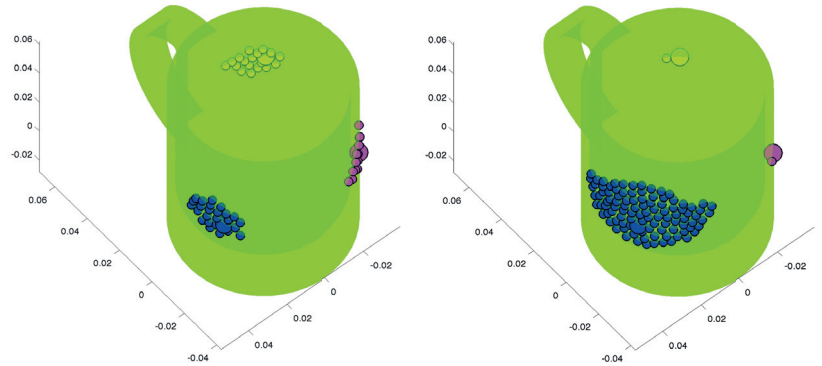


Figure 4.9: *ICR - Parallel Shifting vs. Prioritizing Fingers*: ICR construction for a 3-fingered prototype grasp with frictional hard-finger contacts. Prototype grasp contact points $\mathbf{p}_{g_i}$ are indicated by large spheres. The considered TWS contains the zero-wrench and one wrench generated by the gravitational force which acts in negative vertical direction. (a) ICR constructed from an EWS approximated by parallel shifting according to Algorithm 5. (b) ICR constructed with an EWS approximation according to Algorithm 6. The finger corresponding to the region depicted in blue is prioritized which results in an enlarged region. As a trade-off, regions associated with non-prioritized fingers contain fewer contacts. Utilized parameters: $\mu = 0.5$, $l = 9$, $g = 1$, $m = 0.1$.

Algorithm 6: *EWSApproximation* - Prioritize Finger

Input: prototype grasp $\mathcal{G}_{\text{init}}$, task wrenches $\mathbf{T}$, finger i to prioritize, target object $\{\mathbf{p}_c, \mathbf{n}_c\}$
Output: EWS approximation $(\mathbf{E}, \mathbf{s})$

Compute the GWS of $\mathcal{G}_{\text{init}}$ in $\mathcal{H}$-representation $(\mathbf{A}, \mathbf{b})$ according to (4.4)
for $h \leftarrow 1$ *to* u **do** /* Loop over all u facets of the GWS */
  Form the sets $\mathcal{V}_{h,i}$ and $\mathcal{U}_{h,i}$ according to (4.16) and (4.17) from the GWS
  Solve the QP in (4.18) /* Compute the normal direction */
  Compute $\boldsymbol{e}_h$ according to (4.19)
  Compute s_h by solving the LP in (4.14) /* Compute the offset by parallel shifting */
return $(\mathbf{E}, \mathbf{s})$

sarily makes this facet tangent to the TWS. Therefore, and in order to possibly open up search zones corresponding to regions associated with non-prioritized fingers, the final offset s_h of facet h is computed via parallel shifting along the tilted normals. Thus, Algorithm 6 also necessitates the solution of u LP's according to (4.14). An exemplary comparison of ICR resulting from EWS approximations by parallel shifting according to Algorithm 5 and prioritizing one finger according to Algorithm 6 is depicted in Fig. 4.9.

Prioritizing Contact Points

As a final method to approximate the EWS we suggest a method which accepts as user-input a TWS and desired regions $\mathcal{R}_i$ in addition to a prototype grasp $\mathcal{G}_{\text{init}}$. The motivation is to loosen the conditioning on the prototype's GWS by directly specifying independent regions whose shape and distribution befit the considered application. Here, for reasons which will become clear in the following, we only consider frictionless hard-finger contacts. Considering the sets of k-dimensional wrenches $\mathcal{W}_{\mathcal{R}_I}$ in (4.9) associated to the contacts in regions $\mathcal{R}_i$, let us formulate

Proposition 4.3 (EWS facets)**.**
In the case of frictionless contact constraints, the facets of the EWS lie on hyperplanes each of which is spanned by wrenches from k *different sets* $\mathcal{W}_{\mathcal{R}_i}$.

According to (4.8), the EWS is formed by the intersection of all GWS associated to viable grasps. Therefore, the facets of the EWS lie on hyperplanes which also contain certain limiting GWS (see Fig. 4.4(c)). Recall, that the facets of polytopes are given in simplicial form. Thus, in the frictionless case where only a single wrench is generated at any grasp contact point, the hyperplanes containing these facets have to be spanned by wrenches from k different sets $\mathcal{W}_{\mathcal{R}_i}$ by definition which validates the above proposition.

Here, the core idea is to approximate the h-th confining hyperplane of the EWS by a maximum-margin separation of the task wrenches from the wrenches associated to contacts in those predefined regions $\mathcal{R}_i$, which are associated to the k fingers contributing wrenches which span the h-th facet of GWS_{init} (see Fig. 4.10). To this end, we define the sets containing the indices of those fingers which contribute primitive wrenches spanning the h-th facet of GWS_{init}

$$\mathcal{F}_h = \big\{ i \in \{1, \dots, f\} : \mathcal{V}_{h,i} \neq \emptyset \big\}. \tag{4.20}$$

In the formulation above, the set $\mathcal{V}_{h,i}$ according to (4.16) holds those vertices of the h-th facet which correspond to finger i. Expression (4.20) allows to formulate contact sets

$$\mathcal{R}_{\mathcal{F}_h} = \left\{ c \in \bigcup^{i} \mathcal{R}_i,\ \forall i \in \mathcal{F}_h \right\}, \tag{4.21}$$

which contain the union of independent regions associated with the fingers in $\mathcal{F}_h$. Again, the problem of finding the maximum-margin separating hyperplane can be formulated by a linearly constrained convex QP

$$\begin{aligned} &\underset{\boldsymbol{e}'_h \in \mathbb{R}^k,\, s'_h \in \mathbb{R}_+}{\text{minimize}} && \frac{1}{2}\|\boldsymbol{e}'_h\|_2^2 \\ &\text{subject to} \\ &-\boldsymbol{w}(\mathbf{p}_c)^{\mathsf{T}} \boldsymbol{e}'_h - s'_h \geqslant 1, && \forall c \in \mathcal{R}_{\mathcal{F}_h} \\ &\mathbf{T}^{\mathsf{T}} \boldsymbol{e}'_h + \mathbb{1} s'_h \geqslant \mathbf{0}. \end{aligned} \tag{4.22}$$

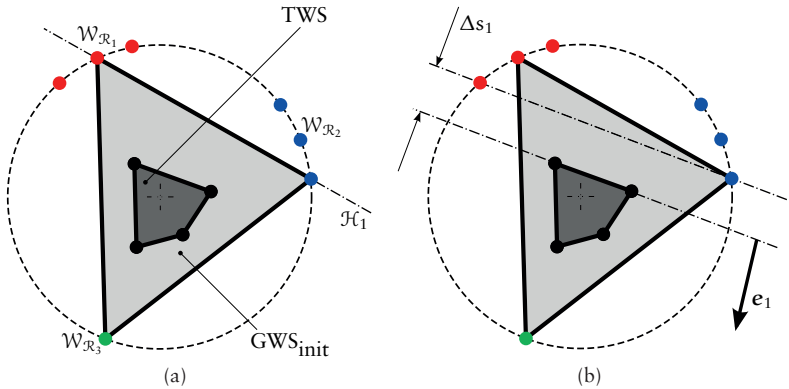


Figure 4.10: *EWS Approximation - Prioritize Contacts*: Illustrated is the exemplary construction of one approximated hyperplane of the EWS with predefined regions $\mathcal{R}_i$ for frictionless contact constraints. (a) Shown are the prototype grasp's GWS, the considered TWS, the wrench sets $\mathcal{W}_{\mathcal{R}_i}$ in (4.9) corresponding to the predefined regions $\mathcal{R}_i$ and hyperplane $\mathcal{H}_1$ which defines a facet of GWS_{init}. (b) Depicted is the maximum-margin hyperplane $(\boldsymbol{e}_1, s_1)$ yielded by solving (4.22). For the presented example, the set $\mathcal{F}_h = \{1, 2\}$ according to (4.20) and thus $\mathcal{R}_{\mathcal{F}_h}$ in (4.21) is $\mathcal{R}_1 \cup \mathcal{R}_2$. The separating margin amounts to $\Delta s_1 = 1/\|\boldsymbol{e}_1'\|_2$ and thus is maximized by minimizing $\|\boldsymbol{e}_1'\|_2$ as in (4.22).

Algorithm 7: *EWSApproximation* - Predefined Regions

Input: prototype grasp $\mathcal{G}_{\text{init}}$, task wrenches $\mathbf{T}$, desired regions $\{\mathcal{R}_1, \ldots, \mathcal{R}_f\}$, target object $\{\mathbf{p}_c, \mathbf{n}_c\}$
Output: EWS approximation $(\mathbf{E}, \mathbf{s})$

Compute the GWS of $\mathcal{G}_{\text{init}}$ in $\mathcal{H}$-representation $(\mathbf{A}, \mathbf{b})$ according to (4.4)
for $h \leftarrow 1$ *to* u **do** /* Loop over all u facets in GWS_{init} */
  From the GWS, compute the sets $\mathcal{F}_h$ and $\mathcal{R}_{\mathcal{F}_h}$ according to (4.20) and (4.21)
  Solve the QP in (4.22) /* Compute the normal direction */
  if *QP in* (4.22) *is not feasible* **then**
    return false /* TWS and the wrenches associated with $\mathcal{R}_{\mathcal{F}_h}$ are not separable */
  Compute $\boldsymbol{e}_h$ and s_h according to (4.19)
return $(\mathbf{E}, \mathbf{s})$

In the above problem, $\boldsymbol{w}(\mathbf{p}_c) \in \mathbb{R}^k$ is the wrench generated by the normal force at contact point $\mathbf{p}_c$, matrix $\mathbf{T} \in \mathbb{R}^{k \times z}$ contains the task wrenches and $\mathcal{R}_{\mathcal{F}_h}$ is the contact set defined in (4.21). The final hyperplane representation $(\boldsymbol{e}_h, s_h)$ can be computed from the scaled quantities in (4.22) via Equation (4.19). An illustration of the construction procedure is depicted in Fig. 4.10.

The corresponding affine transformation yielding the approximated EWS is summarized in Algorithm 7. Invalid user-specified ICR are detected if not

all wrench sets corresponding to $\mathcal{R}_{\mathcal{F}_h}$ are linearly separable from the TWS. Again, as an initial step, the convex hull over the prototype's grasp contact wrenches needs to be computed. Since, in the considered frictionless case only a single wrench is generated at each grasp contact, this requires time of order $O(f^3/6)$ [94] for 6-dimensional input. An upper bound for the maximum number of facets of GWS_{init}, and thus for the QP's to be solved in Algorithm 7, can be given as $u \leqslant f^3/6$ [21]. Forming the index sets $\mathcal{F}_h$ and $\mathcal{R}_{\mathcal{F}_h}$ according to (4.20) and (4.21) respectively, can be done in linear time. Figure A.2 in Appendix A shows an example of sequentially computing the hyperplanes confining the EWS given the regions $\mathcal{R}_i$ predefined on the discretized ellipse shown in Fig. A.1.

The above described way of approximating the confining hyperplane of the EWS does not guarantee for Proposition 4.3 to hold true. It is possible that the number of wrenches which act as support vectors of the h-th approximated hyperplane is smaller than k. The computation just tries to maximize the margin between the TWS and the respective wrench sets corresponding to $\mathcal{R}_{\mathcal{F}_h}$ in (4.21). This can cause the EWS approximation to be unnecessarily limited in some areas (see Fig. A.2(c)). Furthermore, if not all viable grasps are similar according to Definition 4.6.1 (which cannot be verified a priori), the grasp wrench spaces confining the actual EWS can comprise vertices generated by different fingers than those indexed by $\mathcal{F}_h$ in (4.20). Again, this can result in unnecessarily restrictive approximating hyperplanes. However, the following is guaranteed: by construction, all wrenches corresponding to $\mathcal{R}_{\mathcal{F}_h}$ lie in the closed outer half-space of the associated maximum-margin hyperplane. Thus, the definition of independent regions $\mathcal{R}_i$ according to (4.12) – contacts qualify for inclusion in region $\mathcal{R}_i$ if their corresponding wrenches lie in the intersection of closed outer half-spaces associated with prototype grasp contacts $\mathbf{p}_{g_i}$ – holds true. Hence, the method is conservative in a sense that no false positives are generated (i. e., no grasps are classified as being able to preserve a given TWS if they are not).

Expecting a user to specify sets $\mathcal{R}_i$ which yield a non-empty approximated EWS is a stringent requirement. Therefore, instead of predefining ICR, a user can provide a logic for sequentially including points in regions $\mathcal{R}_i$ in order to "grow" them from the corresponding initial grasp points. Consider adding points to region $\mathcal{R}_i$ by exploring the mesh which represents the target object's surface according to a simple breadth-first tree search with $\mathbf{p}_{g_i}$ as root node. Each time a point is added, Algorithm 7 is used to update the EWS approximation. If the point is feasible (i. e., the EWS approximation contains the TWS), the neighbors of the point are enqueued in the search. The procedure stops when no more feasible points are found. Regions $\mathcal{R}_i$ can be prioritized according to their position in the sequence. In particular, choosing the sequence $i = (1, \ldots, f)$ provides an alternative way of computing ICR which, compared to the method of parallel shifting, yields larger regions for the same TWS as illustrated in Fig. 4.11(a).

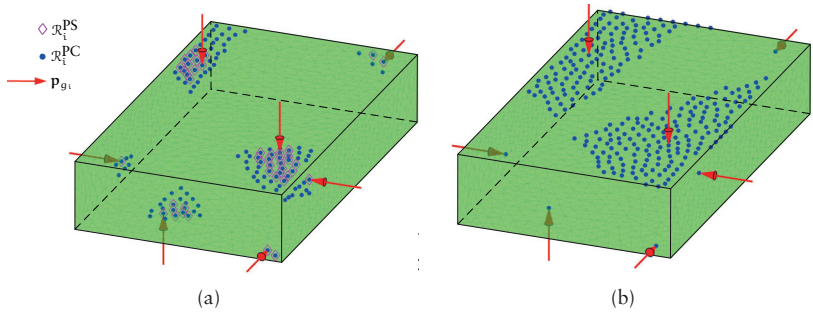


Figure 4.11: *ICR - Parallel Shifting vs. Prioritizing Contacts*: Shown are ICR for a 7-fingered prototype grasp with contact points $\mathbf{p}_{g_i}$ considering frictionless point contact constraints and a TWS consisting of the zero-wrench. (a) Comparison between regions $\mathcal{R}_i^{PS}$ constructed from an EWS approximation via parallel shifting, and regions $\mathcal{R}_i^{PC}$ computed by sequentially adding contacts and approximating the EWS via Algorithm 7. (b) Exemplary growing ICR with five contacts fixed beforehand results in enlarging the two remaining regions.

If the position of certain initial grasping points $\mathbf{p}_{g_i}$ is known precisely, e. g., when there are several locator pins to hold a workpiece [41], it is possible to exclude the corresponding regions $\mathcal{R}_i$ from the sequence. This allows for more points to be included in the remaining regions. An example is shown in Fig. 4.11(b), where contacts are only added to two out of seven regions $\mathcal{R}_i$.

At this point, we want to elaborate on the restriction to the frictionless point contact model when approximating the EWS via prioritizing contact points. Incorporating frictional contact constraints necessitates to separate the task wrenches from a convex combination of the primitive wrenches corresponding to each contact in the set $\mathcal{R}_{\mathcal{F}_h}$ in (4.21). As a consequence, the QP in (4.22) needs to include the vectors of convex combination coefficients $\boldsymbol{\alpha}_c \in \mathbb{R}^l$ among the decision variables resulting in

$$\begin{aligned} &\underset{\boldsymbol{e}'_h \in \mathbb{R}^k,\, s'_h \in \mathbb{R}_+,\, \{\boldsymbol{\alpha}_c \in \mathbb{R}^l :\, c \in \mathcal{R}_{\mathcal{F}_h}\}}{\text{minimize}} && \frac{1}{2}\|\boldsymbol{e}'_h\|_2^2 \\ &\text{subject to} \\ &-\boldsymbol{\alpha}_c^T \boldsymbol{W}_c^T \boldsymbol{e}'_h - s'_h \geqslant 1, \quad \forall c \in \mathcal{R}_{\mathcal{F}_h} \\ &\mathbf{T}^T \boldsymbol{e}'_h + \mathbb{1} s'_h \geqslant 0, \end{aligned}$$

where $\boldsymbol{W}_c \in \mathbb{R}^{k \times l}$ collects the l primitive wrenches $\boldsymbol{w}_j(\mathbf{p}_c)$ generated at contact $\mathbf{p}_c$. However, this results in a non-convex quadratic constraint in the above QP. As a consequence, solving the problem requires increased computational effort and, in general, it is only possible to find a local minimum. Therefore, we decided not to pursue this approach in this work.

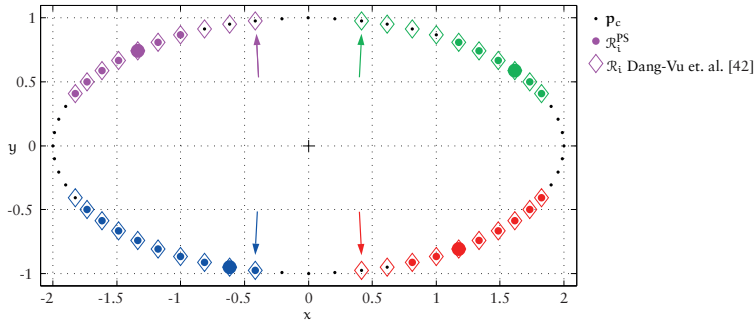


Figure 4.12: *ICR Comparison*: Illustrated are ICR computed for a 4-fingered prototype grasp $\mathcal{G}_{\text{init}}$ with frictional hard-finger constraints. Large dots indicate the prototype grasp's contact points $\mathbf{p}_{g_i}$, a spherical TWS of radius 0.8 times the radius of the largest origin-centered insphere of GWS_{init} is considered. Regions computed according to Algorithms 2, 4 and 5 are indicated by magenta dots, magenta diamonds signify ICR produced by the construction method according to [42]. Arrows highlight the contact points $\mathbf{p}_{\hat{g}_i}$ of the invalid grasp whose GWS is shown in Fig. 4.13(d). Employing the approach in [41] yields empty regions $\mathcal{R}_i$ for this example. Utilized parameters: $\mu = 0.8$, $g = 1$, $l = 2$.

4.6.3 Comparison with Existing Works

Here, we want to put the presented methods in the context of existing works. We provide a comparison with previous approaches which rely on the same prerequisites, i. e., available user-input in form of a prototype grasp $\mathcal{G}_{\text{init}}$ and a set of task wrenches suitable to counter expected disturbances.

From Pollard [38, 39], we adopt the ideas of spanning the search zones in (4.11) belonging to primitive wrenches of $\mathcal{G}_{\text{init}}$, and to qualify contacts for inclusion in independent regions $\mathcal{R}_i$ in (4.12) by checking whether convex combinations of their associated primitive wrenches lie in corresponding search zones. In her work, she provides no explicit notion of the EWS and only considers implicit approximations via parallelly shifting the facets of GWS_{init} a predefined margin.

Roa and Suárez [41] compute ICR by constructing search zones based on parallelly shifting the facets of GWS_{init} until they are tangent to a sphere of a given radius. In order to qualify contacts for inclusion in regions $\mathcal{R}_i$, they introduce the efficient but less stringent alternative to exclusively check if corresponding discrete primitive wrenches exist in search zones as stated in Corollary 4.1. Compared with [38, 39] and our work, they construct search zones in a conservative but overly restrictive manner which is sensitive to the choice of friction coefficient μ in the applied contact model. For the i-th finger, as exemplary illustrated in Fig. 4.6(c), the search zones in [41] are formed by the

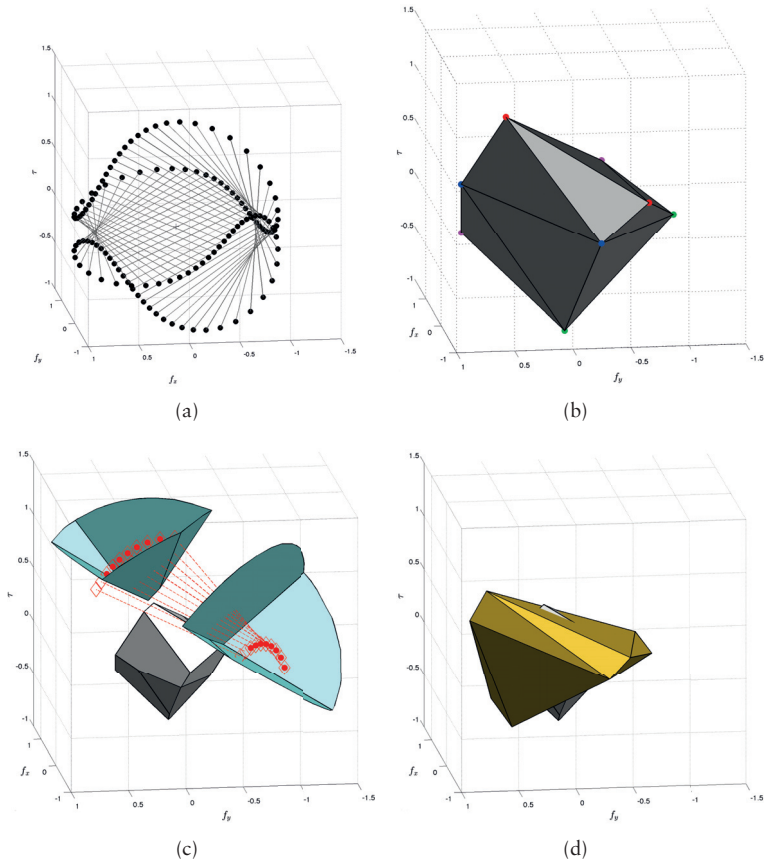


Figure 4.13: *ICR Comparison - Wrench Space*: Illustrated are the geometric relations in the wrench space for the example on the ellipse shown in Fig. 4.12. (a) The OWS according to (4.5) of the considered ellipse. (b) The prototype's GWS. Vertices are color-coded to match the corresponding grasp contact points $\mathbf{p}_{g_i}$ in Fig. 4.12. (c) For one region $\mathcal{R}_i$, polyhedral search zones according to (4.11) are depicted in cyan. They are formed such that contained wrenches satisfy Proposition 4.1. Note, that the intersection of these search zones as in [41] is empty. Red dots indicate the corresponding primitive wrenches whose associated contacts form region $\mathcal{R}_i^{PS}$ in Fig. 4.12. Not all of the primitive wrenches indicated by red diamonds, which are declared eligible by the method in [42], fulfill Proposition 4.1. As a consequence, the EWS (and thus the TWS contained in the EWS) is not necessarily a subset of the GWS of every member of the resulting grasp family. One example is the GWS corresponding to the grasp with contact points $\mathbf{p}_{\hat{g}_i}$ in Fig. 4.12. As shown in (d), it does not fully contain the EWS.

intersection of the l search zones in (4.11) as $\{\mathcal{S}_{i,1} \cup \ldots \cup \mathcal{S}_{i,l}\}$. This is admissible, since a wrench lying in this intersection also lies in each search zone $\mathcal{S}_{i,j}$ and thus the corresponding contact qualifies for inclusion in $\mathcal{R}_i$. However, it leads to the paradox that the number of elements in regions $\mathcal{R}_i$ decreases with increasing friction coefficient. This is due to the fact that the intersection of search zones "moves away" from the OWS and is therefore often empty as shown in Fig. 4.13(c).

In a recent work, Dang-Vu et al. [42] suggest a method to compute regions $\mathcal{R}_i$ which lessens the restrictiveness of [41]. They use this ICR as seeds for an algorithm which extends them based on a criterion proposed in [37]. However, their way of computing these seed ICR via search zones constructed by parallel shifting of GWS_{init}'s confining hyperplanes is not conservative. It potentially allows for members of the resulting grasp family to violate the posed task requirements. As sketched in Fig. 4.6(e), in [42] a contact qualifies for inclusion in region $\mathcal{R}_i$ if it contributes primitive wrenches lying in outer half-spaces of every confining EWS hyperplane associated to the i-th finger of the prototype grasp. This allows to violate the visibility concept formulated in Proposition 4.1. Figure 4.12 shows an exemplary comparison between the seed ICR computed according to [42] and our ICR computation based on an EWS approximation via parallel shifting. The corresponding geometric relations in the wrench space are depicted in Fig. 4.13, which makes evident that the approach in [42] can lead to grasps which are not viable according to Definition 4.2.1. For clarity, in the presented example only $l = 2$ primitive wrenches are generated at a contact $\mathbf{p}_c$, *i. e.*, the corresponding wrench set $\mathcal{W}_c$ does not contain the wrench generated by the contact normal force as stated in Section 4.3 which, however, does not qualitatively change the result.

4.7 Discussion

In this chapter, following the central tenet of this thesis, we incorporate empirical user input in form of a prototype grasp to synthesize grasp families represented as ICR. The introduced algorithms are largely based on the solution of convex optimization problems and can be efficiently parallelized. The suggested methods allow to incorporate physically motivated tasks in the synthesis process. Furthermore, it is shown how to incorporate additional user-input in order to allow prioritization of single fingers or desired grasp contact regions.

The ICR paradigm operates on a grasp contact-level and thus does not constitute a solution to the full grasp synthesis problem, which involves determining an appropriate palm pose and hand joint configuration. However, it can be useful in various auxiliary roles during the synthesis process as shown in the following chapter.

Chapter 5
Evaluation and Applications of Independent Contact Regions

Grasp contact point synthesis
Palm pose synthesis
Hand joint configuration synthesis
Hand motion planning and control

The main contribution presented in this chapter is an open-source C++ library called icrcpp, in which most of the algorithms discussed in the previous chapter are implemented. Also given are numerical evaluations of these algorithms. Furthermore, we demonstrate applications of the ICR paradigm in the grasp synthesis context including employment as a grasp quality and scoring metric, visual guidance for teleoperation, interactive grasp transfer and as a support tool in finger gait planning.

5.1 Introduction

Modeling its wrench space provides an elegant way to evaluate a grasps stability properties in a physically meaningful way in terms of disturbances which can be resisted. A multitude of wrench space-based quality criteria have been proposed in the literature (see [84] for an overview). In this context, rather than representing a single grasp, ICR constitute a family of similar grasps constructed in a way that each member of this family is guaranteed to resist specified disturbance wrenches. Therefore, the ICR paradigm augments the expressiveness of wrench-based criteria with a notion of uncertainty in finger placement which is represented by the number and distribution of points contained

Figure 5.1: *icrcpp - Evaluation Objects*: All objects comprise approximately 2500 vertices and 5000 triangles.

in the individual independent regions. Recently, ICR were employed in a grasp planning framework [96] in order to account for finger placement uncertainties.

An intuitive way to obtain high quality grasps is to utilize human demonstrations. Aleotti and Caselli [97] showed how to incorporate a wrench space-based quality criterion in a TbD scenario where a human operator teleoperates a model of the grasping device in a virtual environment. They propose the Functional Wrench Space (FWS), which is designed to capture the operator's intent by describing the union of GWS corresponding to a set of demonstrated grasps. In this line of thought, Roa et al. [92] used ICR to provide visual feedback to the operator during the demonstration process. Pollard [39] utilized the concept to demonstrate whole-hand grasps with many contacts and transferred them to similar novel objects. Here, the underlying search zones in the wrench space were used to project a class of grasps equivalent to the demonstrated one onto the new object. However, the key question of how to choose an appropriate reference frame location for this novel object was not addressed in [39].

Below, we present a C++ library which efficiently implements the concepts discussed in the previous chapter and provide a numerical evaluation. In Section 5.3 we present ICR application examples including employment as a grasp quality and scoring measure, visual guidance in virtual and real teleoperation scenarios, interactive grasp transfer and finger gait planning. Finally, in Section 5.4, we discuss the obtained results.

Nomenclature

i	Grasp contact index, $i \in \{1, \ldots, f\}$
$\mathcal{O}$	Contact point index set, $\mathcal{O} = \{1, \ldots, o\}$
$\mathcal{G}$	Contact-level grasp, $\mathcal{G} = \{g_i \in \mathcal{O} : i = 1, \ldots, f\}$
$\mathcal{R}_i$	Index set of contacts forming the i-th ICR, $\mathcal{R}_i \subseteq \mathcal{O}$
μ	Static friction coefficient, $\mu \geqslant 0$
l	Friction cone discretization, $l \in \mathbb{Z}_+$

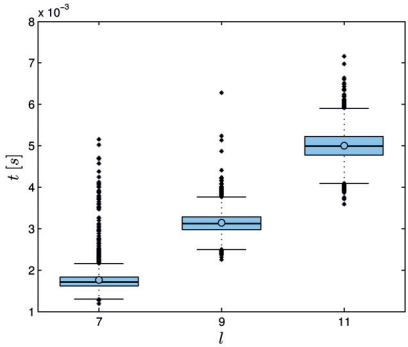


Figure 5.2: *EWS Approximation via Parallel Shifting - Computation Times*: Boxplot showing the computation times for approximating the EWS via Algorithm 5 for a friction coefficient $\mu = 0.5$ and varying friction cone discretizations l.

5.2 Numerical Evaluation

In this section we present the icrcpp library which is available under the GNU General Public License at the url in [98]. Currently implemented are Algorithms 1 and 2, for constructing ICR using either the convex combination-based inclusion test in Algorithm 3 or the discrete primitive wrench inclusion test according to Algorithm 4. Furthermore, the icrcpp library includes Algorithm 5 for approximating the EWS via parallel shifting and Algorithm 6 for an approximation which prioritizes single fingers as described in Section 4.6.2. For evaluation purposes, random grasps were generated on the six object models from the KIT database [99] shown in Fig. 5.1. A standard PC equipped with 6 GB memory and a 3.40 GHz Intel i7-2600 CPU was used to generate the presented results. The Gurobi optimization kit [100], which is freely available for academic purposes, is used by the icrcpp library to solve optimization problems.

For bench-marking, we created nine sets containing 1000 4-fingered prototype grasps each for varying friction coefficients μ and friction cone discretizations l on all test objects. A first set of experiments is centered around ICR construction based on an EWS approximated by parallel shifting the hyperplanes of GWS_{init} via Algorithm 5. The considered TWS contained the zero-wrench and the wrench generated by gravity acting in negative vertical direction at the center of mass, the ratio of gravity and grasp force magnitude was chosen to be $m/g = 0.1$. Figure 5.2 illustrates the resulted computation times for a constant friction coefficient $\mu = 0.5$ and friction cone discretizations $l = 7$, $l = 9$ and $l = 11$ respectively. In no case the required time exceeded 8 ms.

Subsequently, we used Algorithm 2 to construct ICR, respectively using Algorithm 3 or Algorithm 4 to perform the contact inclusion test. The yielded

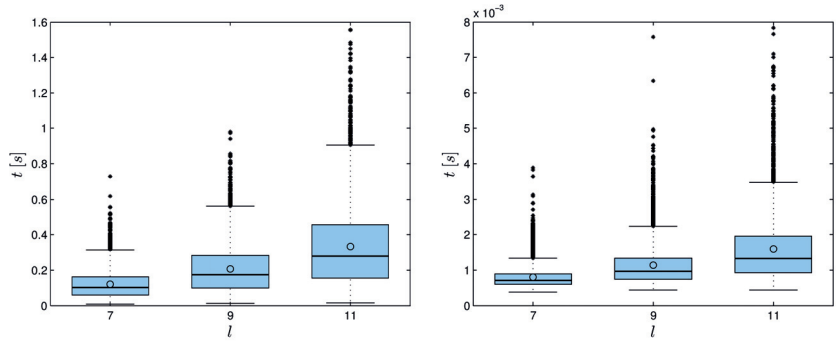


Figure 5.3: *ICR Evaluation - Computation Times*: Boxplots showing the cumulative computation times for Algorithm 2. (a) The convex combination-based inclusion test according to Algorithm 3. (b) The primitive wrench-based inclusion test according to Algorithm 4.

computation times are depicted in Fig. 5.3. It can be seen that the computational effort for solving the LP in the convex combination-based inclusion test is two orders of magnitude larger than only checking discrete primitive wrenches which is in the millisecond range.

As a second measure, we compared the overall number of ICR contacts $\sum |\mathcal{R}_i|$, $i = 1, \ldots, f$ for the two inclusion test variants while varying friction cone discretization l and friction coefficient μ respectively. The outcome is summarized in Fig. 5.4. It is evident that the influence of the friction cone discretization on the ICR size is minor compared to the impact of friction which is significant. For a friction cone discretization $l = 9$ for example, increasing $\mu = 0.5$ to $\mu = 0.8$ increases the number of elements in the regions $\mathcal{R}_i$ by about 80%. In general, the convex combination-based inclusion test yields roughly 15% larger regions than its discrete counterpart. The complete numerical results are collected in Table 5.1.

In a second set of experiments, we evaluated the ICR construction based on an EWS approximated by prioritizing fingers as described in Section 4.6.2. A number of 1000 3-fingered prototype grasps was randomly generated for all six test objects depicted in Fig. 5.1 for a constant friction coefficient $\mu = 0.5$ and a friction cone discretization $l = 9$. For each of these grasps, regions $\mathcal{R}_i$ were computed based on an EWS approximation by prioritizing each of the four fingers, one at a time, according to Algorithm 6. The same TWS as in the previous experiments was employed. Figure 5.5 shows the results. Approximating the EWS via tilting the hyperplanes of GWS_{init} requires the solution of a series of QP's according to Algorithm 6. The resulting computation times

Table 5.1: *ICR Evaluation Results - Parallel Shifting:* Quantities are represented as mean and 1-STD values

l	μ	t^{EWS} [s]	t^{PW} [s]	t^{CC} [s]	$\sum \lvert\mathcal{R}_i\rvert^{PW}$	$\sum \lvert\mathcal{R}_i\rvert^{CC}$
7	0.2	0.0017 ± 0.0002	0.0006 ± 0.0001	0.04 ± 0.03	14.55 ± 12.16	16.49 ± 14.11
	0.5	0.0018 ± 0.0003	0.0008 ± 0.0003	0.12 ± 0.08	52.42 ± 41.56	65.29 ± 48.95
	0.8	0.0018 ± 0.0003	0.0010 ± 0.0005	0.19 ± 0.12	86.71 ± 66.52	114.20 ± 80.66
9	0.2	0.0031 ± 0.0003	0.0007 ± 0.0002	0.06 ± 0.05	16.20 ± 13.52	17.71 ± 14.92
	0.5	0.0031 ± 0.0003	0.0011 ± 0.0006	0.21 ± 0.14	59.68 ± 45.81	69.50 ± 51.40
	0.8	0.0032 ± 0.0003	0.0016 ± 0.0009	0.35 ± 0.22	105.12 ± 76.77	126.30 ± 86.73
11	0.2	0.0050 ± 0.0004	0.0008 ± 0.0004	0.10 ± 0.08	17.12 ± 14.50	18.44 ± 15.81
	0.5	0.0050 ± 0.0004	0.0016 ± 0.0009	0.33 ± 0.23	65.22 ± 49.32	73.23 ± 53.87
	0.8	0.0050 ± 0.0004	0.0024 ± 0.0015	0.55 ± 0.34	113.60 ± 80.69	130.97 ± 89.11

Table 5.2: *ICR Evaluation Results - Prioritizing Fingers:* Quantities are represented as mean and 1-STD values.

l	μ	t^{EWS} [s]	t^{PW} [s]	t^{CC} [s]	$\sum \lvert\mathcal{R}_i\rvert^{PW}$	$\sum \lvert\mathcal{R}_i\rvert^{CC}$
9	0.5	0.1505 ± 0.0091	0.0006 ± 0.0003	0.11 ± 0.10	11.22 ± 17.72	25.54 ± 27.52

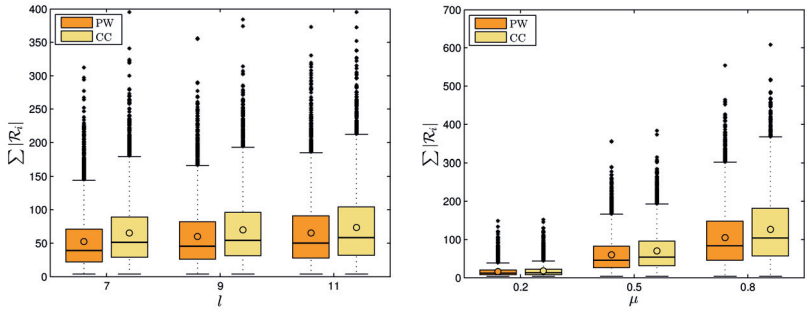


Figure 5.4: *ICR Evaluation - Region Size*: Boxplots comparing the cumulative number of ICR members $\sum \mathcal{R}_i$, $i = 1, \ldots, f$ for the convex combination-based inclusion test according to Algorithm 3 and the primitive wrench-based inclusion test according to Algorithm 4. (a) Constant friction coefficient $\mu = 0.5$ and varying friction cone discretization l. (b) Constant friction cone discretization $l = 9$ and varying friction coefficient μ.

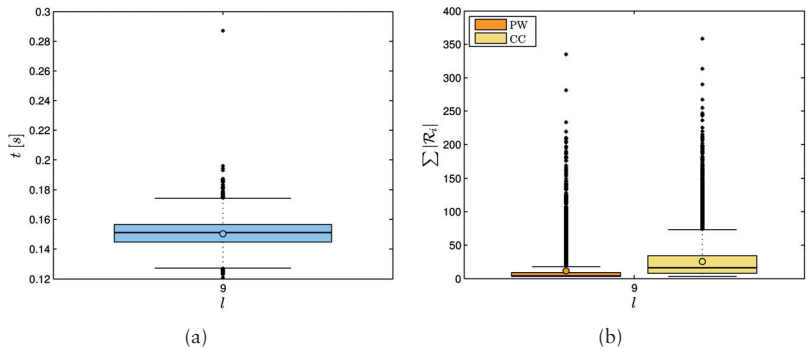


Figure 5.5: *ICR Evaluation - Prioritizing Fingers*: Boxplots showing the evaluation results for regions based on an EWS approximation according to Algorithm 6. (a) Run times for Algorithm 6 for $l = 9$ and $\mu = 0.5$. (b) Comparison of the cumulative number of ICR members $\sum \mathcal{R}_i$, $i = 1, \ldots, f$ for $l = 9$ and $\mu = 0.5$.

are well below 1s, but they are two orders of magnitude higher then in the previously investigated approximation according to Algorithm 5. Here, the influence of the chosen inclusion test variant is of higher importance. On average, regions computed with the convex combination-based inclusion test according to Algorithm 3 contain more than twice the number of elements than regions constructed with the primitive wrench-based test in Algorithm 4. All results are summarized in Table 5.2.

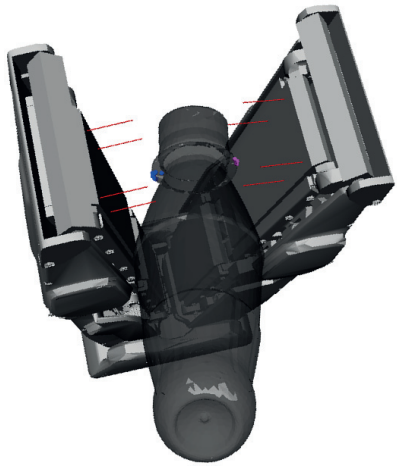


Figure 5.6: *ICR Applications - Grasp Scoring Measure*: Shown is a 2-fingered grasp on the model of the Coke bottle in Fig. 5.1, utilizing frictional soft finger contacts (see Section 4.3). Red lines indicate predefined virtual contacts/contact normals. Prototype grasp $\mathcal{G}_{\text{init}}$ contains the projections of the average virtual contact location for each finger onto the object. ICR are computed via Algorithms 2, 3 and 5. Regions $\mathcal{R}_i$ are indicated by the magenta and blue dots respectively. The TWS if formed by the zero-wrench. Utilized parameters: $\mu = 1$, $\rho = 1$, $l = 10$, $g = 1$.

Currently, the EWS approximation via prioritizing contacts according to Algorithm 7 in Section 4.6.2 is only available as a proof-of-concept Matlab implementation. For a preliminary bench-mark we used the model of a parallelepiped in Fig 4.11, which is sampled with a number of $o = 1714$ contact points and meshed by 3424 triangles. For evaluation purposes, 100 7-fingered prototype form-closure grasps were randomly created for the test object. A TWS represented by an origin-centered sphere with half the radius of the largest insphere of the according GWS_{init} was associated to each of these grasps. ICR were generated based on the EWS approximation according to Algorithm 7, by sequentially adding points to regions $\mathcal{R}_i$ as presented in Section 4.6.2. The flexible orientation of hyperplanes which approximate the EWS allows for a more than three times larger average number of contact points in regions $\mathcal{R}_i = 186 \pm 83$, compared to regions $\mathcal{R}_i = 60 \pm 39$ computed on the basis of a parallel shifting approximation according to Algorithm 5.

5.3 Applications

The implementations of the following practical application examples of the ICR paradigm are built upon the previously discussed icrcpp library.

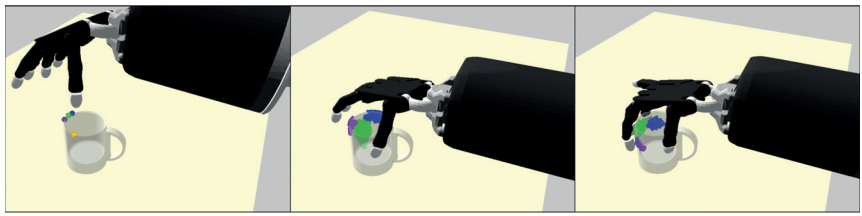
5.3.1 Grasp Quality and Scoring Metric

Employing ICR to quantify the quality of a prototype grasp $\mathcal{G}_{\text{init}}$ is straightforward: by construction, as argued in the previous chapter, every grasp resulting from each finger contacting the object anywhere within its respective region is guaranteed to withstand expected disturbances. Therefore, the number and distribution of contacts in the regions $\mathcal{R}_i$ allows to classify grasps according to the requirements of the considered task. Increasing the number of contacts in regions $\mathcal{R}_i$ can significantly increase the flexibility of $\mathcal{G}_{\text{init}}$ [39]. In this work, we utilize ICR as a support tool in the grasping pipeline which will be detailed in Section 6.2.2. Here, the overall number of ICR elements $\sum \mathcal{R}_i$, $i = 1, \ldots, f$ serves as an argument to the scoring function in (6.2) which ranks the synthesized grasps. An example of ICR computed for the Velvet Gripper is shown in Fig. 5.6. Here, a joint configuration and palm pose according to (2.5) is generated by the grasp synthesis scheme presented in Section 6.2. Subsequently, one predefined contact location for each fingertip is projected onto the surface of the target object (according to the minimal euclidean distance). These projected points then form the contacts of the prototype contact-level grasp $\mathcal{G}_{\text{init}}$ in (2.6) for the ICR computation.

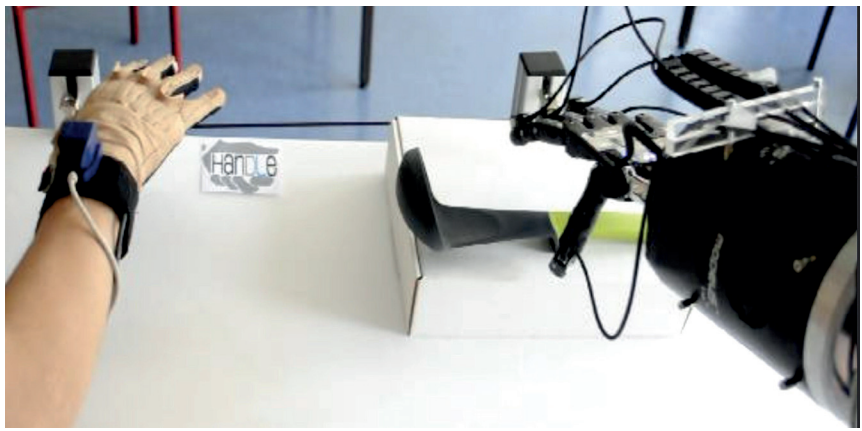
5.3.2 Visual Guidance for Teleoperation

Empirical data is acquired in a TbD setting via human demonstrations in the scope of this thesis. In this context, one arising question is how to solve the correspondence problem (i. e., the physical differences between the demonstrators hand and the targeted robotic platform). As a solution, Babič, Hale and Oztop [101] use teleoperation to make the human part of the control loop and utilize her sensorimotor learning ability where the brain acts as an adaptive controller. The underlying concept is to consider the target robot platform as a tool, and to provide the demonstrator with an intuitive interface to control this tool.

In this section, we show how to incorporate ICR in such an interface in order to visually guide the operator during the demonstration process. Following the idea in [92], we teleoperate the Shadow Robot platform by means of a sensorized glove and a 6D pose tracker (see Section 2.4) in a virtual environment. During demonstration, regions $\mathcal{R}_i$ are computed repeatedly by forming prototype grasps via projecting desired contacts on the fingertips onto the target object as in the previous section. Subsequently, ICR are computed from an EWS approximation based on parallel shifting according to Algorithm 5, utilizing the primitive wrench-based inclusion test in Algorithm 4. In this configuration, as shown in Section 5.2, the overall computation times are in the millisecond range on a standard PC and therefore short enough for real-time applications. Intuitively, grasps comprising large contact regions are advantageous for the robot since they indicate robustness to positioning errors. Thus,



(a)



(b)

Figure 5.7: *ICR Applications - Visual Guidance*: (a) Sequence showing the visual feedback in form of regions $\mathcal{R}_i$, which guides the operator during the demonstration procedure. The prototype grasp contact points $\mathbf{p}_{g_i}$ are signified by large dots, the TWS consists of the zero-wrench. Utilized Parameters: $\mu = 0.8$, $l = 9$, $g = 1$. (b) Grasping a ladle via teleoperating the actual Shadow Robot platform.

while demonstrating, the operator can adjust the grasp configuration based on the visual feedback in order to achieve a suitable final grasp pose as shown in Fig. 5.7(a).

It is difficult to replicate the dynamic properties of a robot precisely in a virtual environment. Therefore, we also tested the proposed interface by teleoperating the actual Shadow Robot platform as shown in Fig. 5.7(b). Here, the visual feedback is presented to the operator on an adjacent computer screen in form of regions on a corresponding model of the target object. In order to find the corresponding reference transform, the pose of the physical object was estimated via the feature matching procedure in [102]. The employed method uses RGB images acquired by the Kinect sensor mounted on the robot's shoulder.

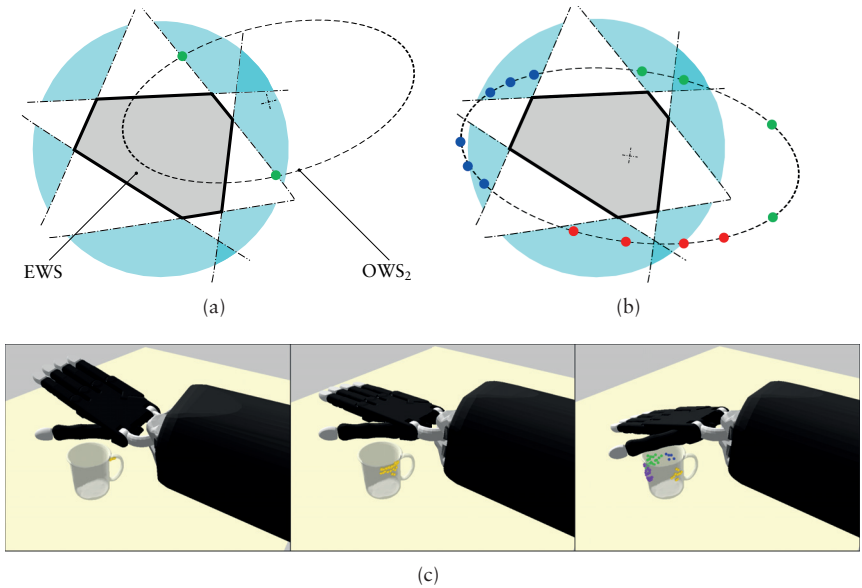


Figure 5.8: *ICR Applications - Interactive Grasp Transfer*: Shown is a sequential illustration of the grasp transfer procedure. (a) The cyan shaded regions indicate search zones $\mathcal{S}_{i,j}$ constructed from GWS_{init}, which corresponds to a prototype grasp performed on object 1. The dashed ellipse represents OWS_2. Some of its wrenches fulfill the inclusion condition which results in non-empty regions $\mathcal{R}_i$. (b) Once the operator brings the palm in an appropriate pose, OWS_2 contributes primitive wrenches (or convex combinations thereof) to all search zones and the transfer is complete. (c) The demonstrator only needs to command the palm pose. During the process, non-empty regions $\mathcal{R}_i$ serve as a visual feedback. Utilized Parameters: $\mu = 0.8$, $l = 9$, $g = 1$.

5.3.3 Interactive Grasp Transfer

Teleoperating a system with many DoF is a strenuous task which requires high concentration. Therefore, we propose an interactive method which eases the burden on the demonstrator by allowing to transfer grasps of high quality from the demonstration object to similar novel objects. The idea is to use an available grasp $\mathcal{G}_{init}$ on object 1 to construct search zones $\mathcal{S}_{i,j}$ according to (4.11) from an EWS approximation based on $GWS_{init} \subseteq OWS_1$. Here, OWS_1 denotes the object wrench space in (4.5) corresponding to object 1, i. e., the set of all wrenches that can be generated by grasp forces acting anywhere on the object's surface. During the transfer step, a family of equivalent grasps is synthesized on object 2 by "filling" the previously computed search zones with primitive wrenches (or convex combinations thereof) from OWS_2 using Algorithm 2 and any of

the two inclusion tests according to Algorithms 3 or 4. The key point here is the choice of reference frame in which the objects are represented, since this determines the geometry of the associated object wrench spaces. Therefore, we suggest to represent objects (and thus their corresponding OWS) in the hand palm frame. In the transfer stage, the operator only needs to teleoperate the palm. At each time step during this process, an affine transformation of OWS_2 into the hand palm frame is performed, i. e., the forces and torques forming the elements of OWS_2 are expressed in the palm frame. Since the two considered objects are geometrically similar, OWS_2 resembles OWS_1 if the two palm poses are comparable and the search zones $\mathcal{S}_{i,j}$ are non-empty. Figures 5.8(a) and (b) illustrate the concept. Figure 5.8(c) exemplary shows the interactive transfer of the demonstrated grasp in Fig 5.7(a) to a novel object.

5.3.4 Finger Gait Planning

Finger gaiting constitutes an interesting aspect of dexterous manipulation. It is defined as relocating the contacts of a given grasp on the target object's surface, such that grasp stability is preserved during the process. In this section, we suggest to use ICR computed while prioritizing single fingers as discussed in Section 4.6.2 as a support tool for finger gait planning. The core idea is to place a redundant finger within a region $\mathcal{R}_i$ associated with another finger, which allows that finger to break contact and be relocated. During this operation, stability is ensured by definition of the ICR paradigm. Here, we assume that a suitable relocation strategy is available which, e. g., can be derived by observing a human demonstrator. An illustrating example is provided in Fig. 5.9 which shows how fingers can be sequentially replaced to be relocated on the object's surface in a clockwise fashion.

5.4 Discussion

In this chapter, we presented and evaluated the icrcpp `C++` library for EWS approximation and ICR computation. Furthermore, we discussed applications of the ICR paradigm related to grasp ranking, teleoperation, grasp transfer and finger gait planning. For evaluation purposes, we used object models acquired with a high-accuracy laser scanner [99] and CAD models. Computing ICR is sensitive to noise in the contact normals which can pose problems when only low-quality object models reconstructed from range data are available. This issue can be addressed by a conservative choice of friction coefficient [103] or by appropriate mesh smoothing techniques [14]. Furthermore, the utilized point contact models often poorly reflect the interaction between finger and object and it has been shown in [15] how to incorporate a more realistic patch contact model in the ICR computation.

Apart from allowing to transfer a contact-level grasp with a human in the loop, the method introduced in Section 5.3.3 also constitutes a solution to the

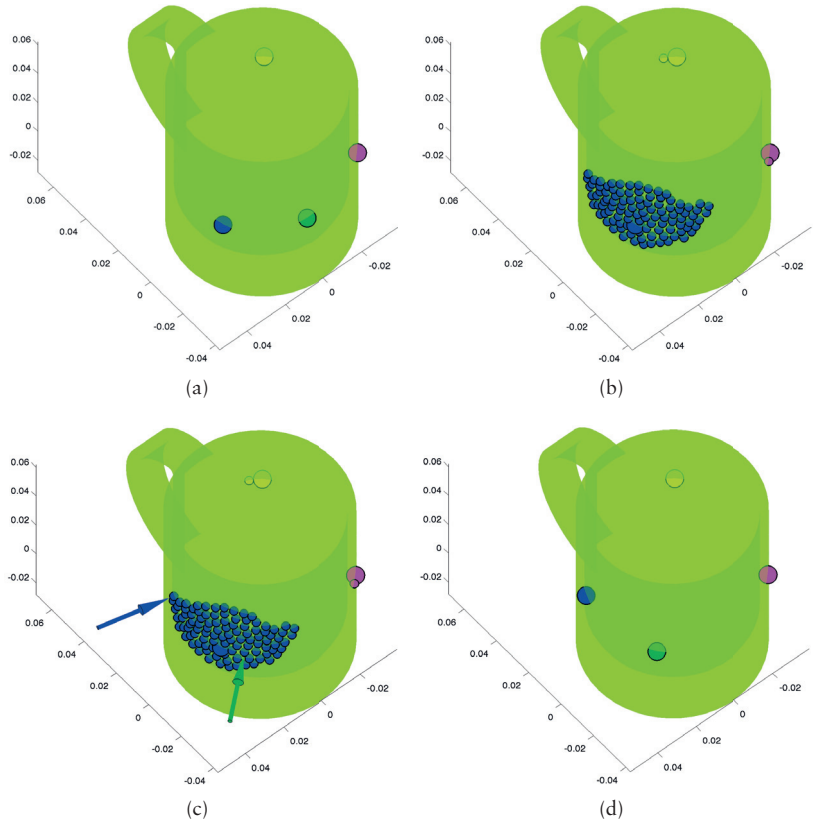


Figure 5.9: *ICR Applications - Finger Gait Planning*: ICR are computed based on the EWS approximation with prioritized fingers according to Algorithm 6, utilizing frictional hard-finger contacts. Algorithm 3 is used for the inclusion test, the TWS consists of the zero-wrench. Utilized parameters: $\mu = 0.5$, $l = 9$, $g = 1$. (a) Initial four-fingered grasp configuration. The predefined goal is to relocate the green and the blue contact. (b) Shown are the ICR resulting from removing the green contact and prioritizing the blue contact. (c) After placing the green contact at the indicated location, the blue contact can be replaced. (d) Final grasp configuration after the gaiting procedure.

palm pose synthesis problem. It would be interesting to investigate ways to automatize the solution, e. g., by treating ICR as features and employing point cloud registration techniques to fit them to a novel object.

Chapter 6
Grasp Synthesis via Constrained Optimization

Grasp contact point synthesis
Palm pose synthesis
Hand joint configuration synthesis
Hand motion planning and control

In this chapter, we incorporate the problem of synthesizing appropriate hand joint configurations in the grasp synthesis process. We contribute an optimization-based grasp synthesis framework which is tailored to underactuated gripping devices operating in cluttered scenes, in order to reflect the requirements of the RobLog project. The presented method incorporates grasp strategies observed in humans by imposing corresponding constraints on the underlying optimization problem. Test runs to evaluate the approach have been conducted with the Velvet Gripper fitted on the two platforms in Fig. 2.2(c) and Fig. 2.2(d) respectively.

6.1 Introduction

Research in the field of robotic grasping has produced a multitude of different gripper designs in an attempt to achieve reliable grasps of various target objects. An overview of existing concepts is provided by Bicchi [104]. One line of research has focused on creating devices mimicking the mechanical structure of the human hand in order to allow grasping/manipulation of objects with a wide range of shapes and sizes [6]. This, however, results in complex designs and control schemes. In the scope of the RobLog project an alternative approach was pursued. Here, the goal is to simplify the design and control by

Figure 6.1: *Gripper and Test Objects:* Shown is the Velvet Fingers gripper mounted on a 7-DoF KUKA lightweight arm. Perception is done with an ASUS Xtion structured light camera mounted on the gripper. Also shown are the five test objects used in the evaluation in Section 6.3 – two different boxes, a beer barrel, a ball and a drum.

employing an underactuated grasping device which preserves desired grasping and manipulation features.

A recently investigated way to achieve more dexterous grippers is the addition of active surfaces to otherwise simple mechanical structures. Active surfaces have been used to regulate the adhesion between fingers and target object [105], or to augment the mechanical structure with conveyor belts to control the tangential push exerted on the target object. In the scope of RobLog, we utilize an implementation of such a device in form of the Velvet Fingers gripper [26, 27] which is depicted in Fig. 6.1. This gripper combines underactuation and active surfaces in the form of conveyor belts on the finger phalanges. The mechanical design (discussed in more detail in Section 6.3.1) features one actuated degree of freedom for opening and closing and two for the belt movements. If, during grasping, the proximal phalanges are blocked by the object, the gripper's distal phalanges continue to "wrap around" the object and envelope it in a firm grasp.

Of particular interest in this context are two questions which arise when using such a device in a grasping scenario where a cluttered scene containing multiple objects is to be cleared by a robot:

- How to account for an underactuated gripper structure during grasp synthesis?
- How can active surfaces contribute to the grasp execution process?

In this chapter, we investigate a data-driven approach to grasp synthesis, where a knowledge base is populated offline with target object models which are asso-

ciated to pre-planned grasps. During execution, the database is used to retrieve grasps and rank them according to the expected quality in the current scene.

The contributions in this chapter are two-fold: first, we address the grasp synthesis problem by adapting the well known optimization-based scheme introduced by Ciocarlie and Allen [54] to the specifics of underactuated gripper designs. Second, we investigate a method to improve the success of the grasp execution process by using the gripper's active surfaces. We employ a simple "pull-in" strategy where the target object is simultaneously manipulated and grasped by using the belts to pull the object towards the palm of the gripper while closing the fingers. The use of active surfaces regulating adhesion for gripping devices has so far been only explored on the micro/nano scale [105]. Some designs of grippers utilizing belts have been employed in industrial settings but, to the best of our knowledge, no attempt has been made so far to develop a programmatic strategy of utilizing active surfaces during grasp execution.

This chapter is organized as follows: below, we introduce our grasp synthesis and execution scheme before describing the experimental setup, the target scenarios and the obtained results in Section 6.3. The final conclusions are discussed in Section 6.4.

Nomenclature

j	Index used for virtual contacts, $j = 1, \ldots, v$
$\mathbf{n}_j$	Surface normal at the gripper's j-th virtual contact
q	Gripper opening angle $q \in \mathbb{R}$
ϕ	Palm roll angle, $\phi \in \mathbb{R}$
ψ	Palm pitch angle, $\psi \in \mathbb{R}$
d	Approach distance, $d \in \mathbb{R}_+$
$\mathbf{P}$	Gripper palm pose, $\mathbf{P} = \mathbf{P}(d, \phi, \psi) \in \mathbb{R}^6$
$\mathbf{p}_c$	Contact point on a discrete object's surface, $\mathbf{p}_c \in \mathbb{R}^3$
$\mathbf{n}_c$	Contact normal at $\mathbf{p}_c$, $\mathbf{n}_c \in \mathbb{R}^3$
G	Grasp, $G = (\mathbf{P}, q)$
$E(\cdot)$	Grasp energy function, $E : \mathbb{R}^6 \times \mathbb{R} \rightarrow \mathbb{R}$
s	Scaling factor, $s \in \mathbb{R}_+$
t	Desired number of synthesized grasps, $t \in \mathbb{Z}_+$
h	Number of grasp synthesis attempts per approach direction, $h \in \mathbb{Z}_+$
f	Number of fingers of the grasping device, $f \in \mathbb{Z}_+$
$S(\cdot)$	Grasp scoring function, $S : \mathbb{R}^4 \rightarrow \mathbb{R}$
s_E	Grasp energy score, $s_E \in \mathbb{R} : 0 \leqslant s_E \leqslant 1$
s_I	Grasp ICR score, $s_I \in \mathbb{R} : 0 \leqslant s_I \leqslant 1$
s_O	Gripper-rel. orientation score, $s_O \in \mathbb{R} : 0 \leqslant s_O \leqslant 1$

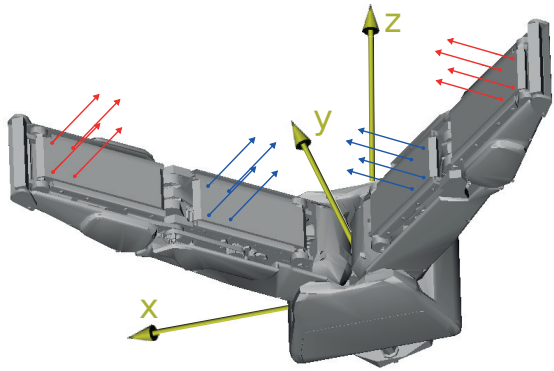


Figure 6.2: *Predefined Virtual Contact Locations:* Shown are the preset virtual contact locations and corresponding surface normals $\mathbf{n}_j$ on the Velvet Fingers gripper. Contacts depicted in blue result in enveloping grasps, for the synthesis of fingertip grasps the set of contacts shown in red were used.

6.2 Grasp Synthesis and Execution

Underactuated grippers can only control their active joints and rely on physical interaction with the environment for reconfiguring the passive ones. Thus, accurately accounting for the interaction between such a gripper and a target object in the grasp synthesis process would require a detailed dynamic simulation of the grasp procedure. Even then, grasps pre-planned in this fashion might fail when applied in cluttered environments, since a physical simulation of the current scene would be necessary to guarantee successful target object retrieval.

Here, we chose a different strategy aimed at exploiting the underactuated structure and the active surfaces of the Velvet Fingers gripper to simplify grasp synthesis. Therefore, for synthesis purposes, we assume extended distal links and a fully actuated mechanical structure comprising only one proximal joint connecting the fingers to the palm (see Section 6.3.1 for an overview of the actual kinematic structure). These assumptions conform to the natural configurations of the gripper during approach (see Fig. 6.1 for an exemplary approach configuration) when no mechanical interaction with the environment occurs. During grasp execution, we then rely on the specific underactuated kinematic and transmission design to envelope the object in a stable grasp or use the belts to pull the object into such an enveloping grasp as described in Section 6.3.3. The goal of the presented grasp synthesis methodology is to find grasps $G = (\mathbf{P}, q)$, which are defined as pairs of a palm pose $\mathbf{P} \in \mathbb{R}^6$ and a joint configuration as discussed in Section 2.2. Compared to the general grasp definition in (2.5), a joint configuration of the Velvet Gripper is described by a single angle $q \in \mathbb{R}$ only.

Our grasp synthesis methodology is built upon the work by Ciocarlie and Allen [54], who generate pre-grasps in low-dimensional hand joint subspaces by minimizing an energy function based on distance/alignment between predefined virtual contact locations on the hand and the target object. The corresponding non-convex optimization problem is solved via the derivative-free simulated annealing algorithm, which has been shown to work well for synthesizing enveloping grasps for multi-fingered hands [53, 45]. The Velvet Gripper's relatively large size, and the fact that target objects can only be approached with extended distal links, makes many grasps with inappropriate palm poses infeasible since they would result in collisions with the environment. Consequently, we additionally incorporate the following two heuristics which correspond to observations of human grasp strategies:

i) It has been shown, that most successful grasps usually approach along a surface normal of the object [56, 106].

ii) In many successful grasps, the grasping device's lateral axis (the x-axis in Fig. 6.2) is normal to one of the principal component directions of the object.

In accordance with the first observation, the approach direction (the y-axis in Fig. 6.2) is constrained to be in an environment around a surface normal $\mathbf{n}_c \in \mathbb{R}^3$ of the object. We plan palm poses $\mathbf{P} = \mathbf{P}(d, \phi, \psi)$ over the approach distance $d \in \mathbb{R}_+$ and palm roll and pitch angles $\phi \in \mathbb{R}$ and $\psi \in \mathbb{R}$. To this end, we minimize the following energy function $E : \mathbb{R}^6 \times \mathbb{R} \to \mathbb{R}$ which was introduced in [54]

$$E(\mathbf{P}, q) = \sum_{j=1}^{v} \left(1 - \frac{\mathbf{n}_j^T \mathbf{o}_j}{\|\mathbf{o}_j\|_2} + \frac{\|\mathbf{o}_j\|_2}{s} \right), \tag{6.1}$$

where $v \in \mathbb{Z}_+$ indicates the number of preset virtual target contact locations on the phalanges of the gripping device (see Fig. 6.2), $\mathbf{n}_j \in \mathbb{R}^3$ denotes the outward-pointing surface unit normal at the j-th virtual contact, $\mathbf{o}_j \in \mathbb{R}^3$ is the vector from virtual contact j on the hand to the closest point $\mathbf{p}_c \in \mathbb{R}^3$ on the object and $s \in \mathbb{R}_+$ is a scaling parameter (for clarity, the implicit dependence of $\mathbf{o}_j$ and $\mathbf{n}_j$ on $\mathbf{P}$ and q is omitted in the notation). The first term in Equation (6.1) captures the alignments between the object and target contact locations on the hand, the second term indicates the distances of the target contacts to the object. As in [54], the grasp energy in (6.1) is minimized using the simulated annealing algorithm which is able to escape local minima through possible "uphill moves" during optimization via generating random neighbors.

Given a vertex $\mathbf{p}_c$ with associated vertex normal $\mathbf{n}_c$ on the target object's discretized surface, we account for the second of the aforementioned observations by providing the solver with appropriate initial conditions. To this end, the gripper is positioned at a fixed offset distance from $\mathbf{p}_c$ along the vertex

Algorithm 8: *Grasp Synthesis*

Input: Target object $\{\mathbf{p}_c, \mathbf{n}_c\}$, synthesis attempts per approach direction h, desired number of grasps t, grasp energy threshold E_{max}
Output: Set of synthesized grasps $\{G\}$

```
g ← 0
while g < t do
    /* Sample a vertex and associated vertex normal without replacement */
    (p_c, n_c) ← randomSample({p_c, n_c})
    for q ← 1 to h do                    /* Initialize the solver from h initial states */
        φ_0 ← (q − 1)2π/h                /* Compute the initial roll angle φ_0 */
        initializeGripperPose(p_c, n_c, φ_0)
        minimize_{P,q} E(P, q) in (6.1)
        subject to box constraints on all variables
        /* Add the solution to the output set if it satisfies the energy threshold */
        if E ⩽ E_max then
            add (P*, q*) to {G}
            g ← g + 1
return {G}
```

normal $\mathbf{n}_c$ in negative direction. The initial palm rotation is determined such that the lateral axis of the gripper is normal to the object's principal component direction with the largest eigenvalue which is not parallel to $\mathbf{n}_c$. This rotation is then offset by an initial roll angle ϕ_0. During optimization, box constraints are enforced to ensure that the variables do not diverge too far from the initial condition. The proposed method allows to put bounds on the resulting grasp poses while, opposed to approaches solely relying on heuristic sampling [56, 52], still retains the flexibility to adjust them to the specific object geometry.

Our grasp synthesis methodology is summarized in Algorithm 8 and was integrated in the GraspIt! [53] simulator, using GraspIt!'s simulated annealing solver. We iteratively sample vertex/vertex normal pairs from a discretized target object representation. Subsequently, each time a new grasp is found, the solver is restarted with initial states as described above. In Algorithm 8, h synthesis attempts are made for each approach direction. To obtain grasps conforming to other principal directions and to obtain symmetrical grasp hypotheses, $h = 4$ is typically chosen since this advances the initial roll angle ϕ_0 in steps of size $\pi/2$. Our optimization scheme only operates on four decision variables which results in fast synthesis times. Grasps comprising energies in (6.1) above a given threshold E_{max} are discarded.

In Section 6.3 we evaluate our synthesis scheme with two different sets of contact references as illustrated in Fig. 6.2. The choice of reference locations on the gripper provides an easy way to control the resulting grasp configura-

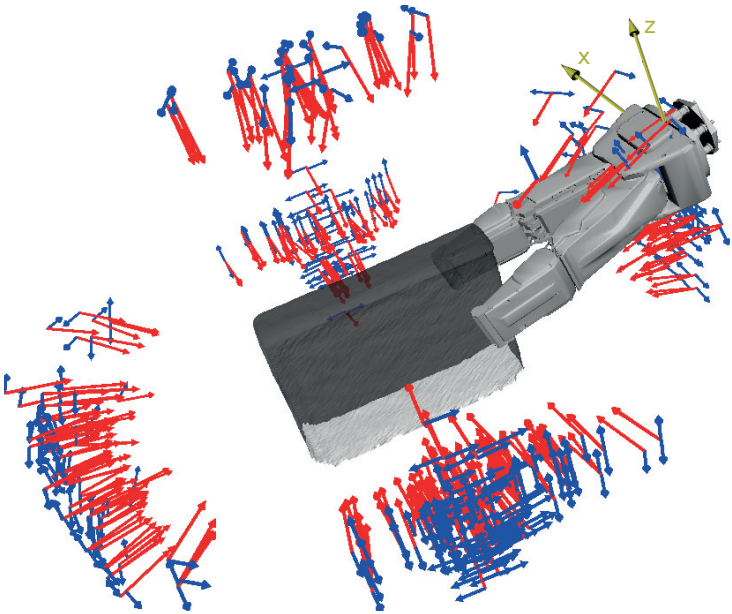


Figure 6.3: *Grasp Synthesis - Example:* Shown is a set of fingertip grasps as computed by the planner. Palm poses are indicated by the approach direction (y-axis) in red and the x-axis in blue. As intended, the resulting palm orientations are roughly normal to principal component directions of the target object.

Figure 6.4: *Pull-in Grasping Strategy:* Depicted is a sequence of intermediate grasp states where the belts of the gripper are used to pull the object towards the palm which results in a transition from a fingertip to an enveloping grasp.

tions. Locations on the proximal phalanges result in enveloping grasps, which are potentially robust. However, in cluttered scenes many enveloping grasps are not achievable without collisions and fingertip grasps are preferable, especially when active surfaces are available to aid the grasp execution process as discussed below. An example for the obtained output is depicted in Fig. 6.3.

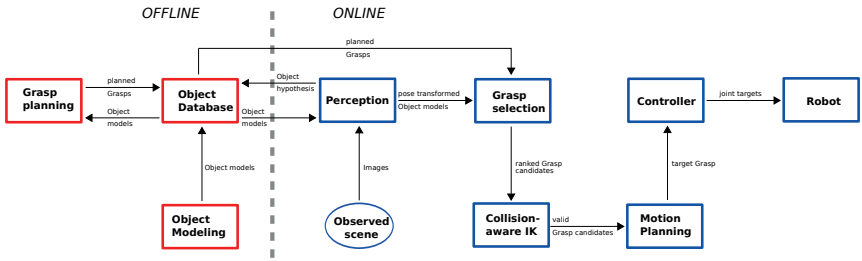


Figure 6.5: *Grasp Execution - Pipeline:* The pipeline uses an offline created database which stores object models together with the synthesized grasps. The perception module creates object hypotheses from the observed scene and matches them to the models in the database. Found models are forwarded to the grasp selection module which ranks the associated grasps according to their score in the current scene. Subsequently, inverse kinematics and goal pose collision checks are performed on a chosen grasp before a motion plan is generated which is then executed by the controller.

6.2.1 Simultaneous Manipulation and Grasping

An interesting possibility offered by the active surfaces of the considered gripper is to manipulate the object while grasping. One idea to improve the grasp execution success in the presence of object pose and gripper positioning uncertainties is to employ a "pull-in" strategy. Here, the belts move the object towards the gripper's palm while the phalanges squeeze the object as illustrated in Fig. 6.4. In section 6.3.3, we provide a verification of this strategy in combination with fingertip grasps in cluttered scenes. Here, enveloping grasps are often infeasible and the pull-in strategy aids in obtaining firm grasps which would not be achievable without the active surfaces.

6.2.2 Grasping Pipeline

To carry out the experiments in Section 6.3 we employ the grasping pipeline illustrated in Fig. 6.5. Our approach employs an offline and an online stage. In the offline stage, we acquire 3D models of the target objects, train a perception module and compute grasps. The first step in the offline stage is to acquire accurate models of the objects of interest. The model acquisition approach follows the work of Mihalyi et al. [107]. First, a number of augmented reality markers are placed in the scene and a set of training RGB-D images is collected. The training set is used to estimate a graph of marker positions and orientations. Next, we successively place each target object in the scene and acquire a set of RGB-D images, which are subsequently registered in a common reference frame using the marker graph. We use the registered depth images to reconstruct a Truncated Signed Distance Field (TSDF) representation of the object from which triangular meshes are extracted using the marching tetrahe-

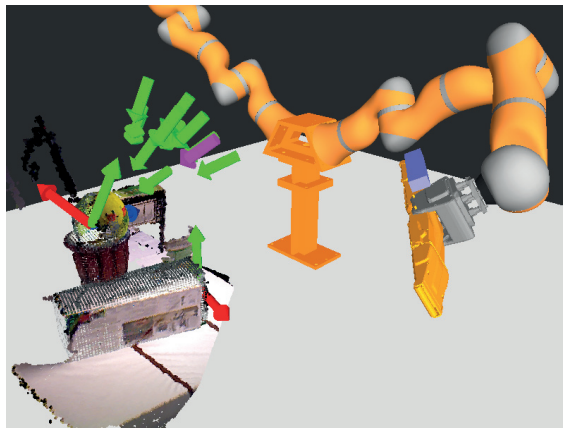


Figure 6.6: *Grasp Synthesis - Retrieval:* Shown is a scene as observed by the robot with two detected objects (ball and box). Also depicted are the feasible grasps $\overline{G}$ for the ball which are indicated by their approach vectors in green, the purple vector signifies the grasp chosen by the grasp selection module.

drons algorithm. The TSDF tracker algorithm in [108] is employed to correct for local inaccuracies in the RGB-D data registration. By fusing consecutive views of the object in the TSDF representation we effectively eliminate a large portion of the sensor noise, leveraging on the changing sensor viewpoints. The smoothed triangle meshes are then stored in the database and used by the previously described grasp synthesis scheme to generate grasps. Thus, contrary to prior approaches that use ground truth geometric models, we train grasps on models reconstructed from noisy sensor observations. The final component of the offline stage uses the RGB-D images of each object to train the recognition modules of the perception system. The Perception module in the pipeline is based on previous work by Vaskevicius et al. [109]. During the offline stage, this system extracts local visual features from the RGB data component, references them based on the depth component and stores the resulting feature graph in the database.

The online stage of the proposed pipeline starts with the acquisition of an RGB-D image of the target scene. Following the work in [109], the image is then over-segmented in patches. Local visual features from each patch are then extracted and compared against the feature graphs stored in the database. If a candidate match to an object is detected, additional checks for consistency are performed by back-projecting the database object to the scene. Once the perception system obtains a list of detected objects, the grasp selection module is used to associate a set of grasps to each of the pose-transformed object candidates. For evaluation purposes, as described in Section 6.3.2, this module tests

all grasps from the database for feasibility, using a fast RRT planner. An example is shown in Fig. 6.6. Given a high number of objects and possible grasps, this strategy may not be feasible for online operation. To this end, we employ a grasp ranking procedure and a first-feasible execution strategy. We follow an approach similar to the one outlined by Berenson et al. [56] and evaluate, for the given environment, a grasp scoring function $S : \mathbb{R}^3 \rightarrow \mathbb{R}$ which is the weighted sum of three sub-scores

$$S(s_E, s_I, s_O; \mathbf{a}) = a_1 s_E + a_2 s_I + a_3 s_O, \tag{6.2}$$

where the vector $\mathbf{a} \in \mathbb{R}^3$, $\mathbf{a} \geqslant \mathbf{0}$ contains positive weights. In this work, we use uniform weights $\mathbf{a} = \mathbf{1}$. The grasp energy score $s_E \in \mathbb{R}$, $0 \leqslant s_E \leqslant 1$ is the value of the energy function in (6.1) divided by the maximum energy of all grasps for an object. To capture the robustness of a grasp to modeling and positioning uncertainties, we employ ICR computed from an EWS approximation based on parallel shifting according to Algorithms 2, 3 and 5 as explained in Section 5.3.1. The corresponding ICR score $s_I \in \mathbb{R}$, $0 \leqslant s_E \leqslant 1$ in (6.2) is composed of one minus the normalized sum of ICR contacts $\sum \mathcal{R}_i, i = 1, \ldots, f$. For the considered Velvet Fingers gripper, the number of fingers $f = 2$. The gripper-relative orientation score $s_O \in \mathbb{R}$, $0 \leqslant s_E \leqslant 1$ in (6.2) captures the similarity of the grasp pose to the current gripper pose and is expressed as one minus the cosine of half the angle between the current and target palm orientation. All grasps are ranked by increasing scores in (6.2) and successively tested for feasibility as described above. Once a valid candidate is found, joint motion trajectories are generated and passed on to the controller to execute the movement. In all experiments, inverse kinematics, collision checking and motion planning were carried out with the MoveIt! framework [110].

6.3 Evaluation and Results

In this section, we outline the hardware setup and target scenarios, before proceeding with a discussion of the obtained results.

6.3.1 System Configuration and Target Scenarios

For the experimental evaluation we used the Velvet Fingers Gripper whose kinematics and control scheme is depicted in Fig. 6.7. All actuators are controlled with simple PID control loops which are closed on the angle rotation of the motor shafts through magnetic encoders. Current sensors on the electronic boards allow to set thresholds on the current absorptions. This ensures a robust grasping behavior and, at the same time, enables safeguarding the entirety of the gripper (see [27] for more details). For all object recognition and collision detection tasks an ASUS Xtion structured light camera, which is mounted on the gripper, was used.

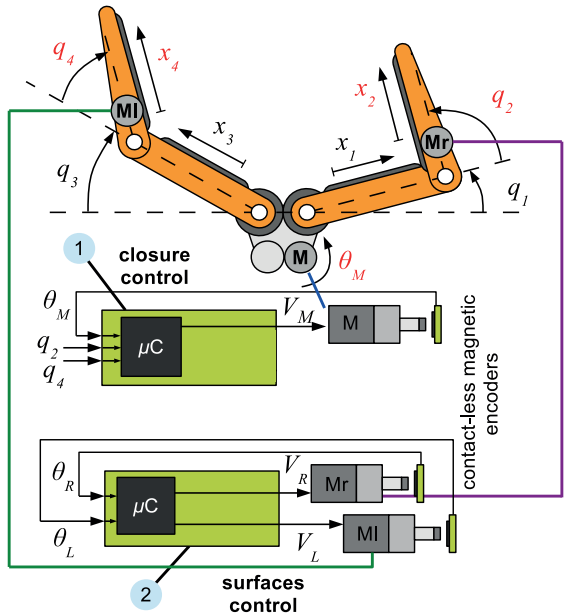


Figure 6.7: *Velvet Fingers Gripper Architecture:* One electronic board controls the motor M for the opening and closing and reads the angular position of the joints q2 and q4. The second board controls the motors M_r and M_l driving the belts. The current assumption of both boards is monitored by corresponding on-board sensors.

In the experiments, we used a database containing the five target objects shown in Fig. 6.1. We opted for a two-staged evaluation. In the first set of experiments we investigate the quality of the proposed grasp synthesis approach under different environment configurations. We then proceed to analyze the performance of the pipeline, including a preliminary performance analysis of the proposed pull-in strategy, in a tabletop grasping scenario utilizing the platforms in Fig. 6.1 and Fig. 2.2(d).

6.3.2 Grasp Synthesis Evaluation

To evaluate the synthesis, we computed three sets of 400 grasps for each object, containing pinch grasps G_P, enveloping grasps G_E and an equal mixture of both G_M respectively. Next, the platform in Fig. 6.1 was used to collect two data sets of depth and color images. The first scene data set $\mathcal{S}_I$ contains three observations of each target object in isolation (i. e., only one object in the scene). The second data set $\mathcal{S}_C$ contains observations of scenes with various amount of clutter (i. e., multiple objects in the scene). Here, a total of fifteen

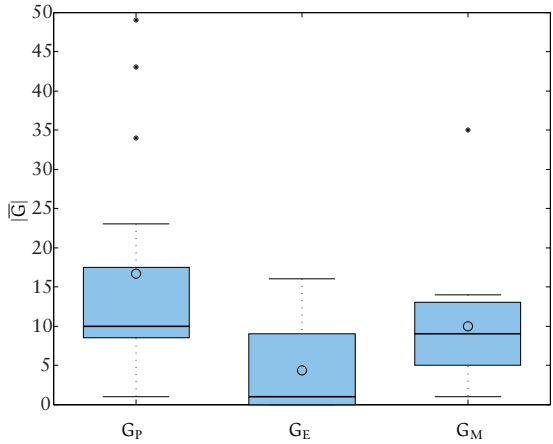


Figure 6.8: *Grasp Synthesis - Results Isolated Objects:* Boxplot showing the number of feasible grasps $|\overline{G}|$ per object (out of a total number of 400 pre-planned grasps) as a function of the chosen synthesis flavor (pinch, envelope or mixed grasps). The grasps were extracted from data set $\mathcal{S}_I$ which contains observations of isolated target objects only.

Table 6.1: *Grasp Synthesis - Parameters:* Predefined parameters for Algorithm 8

$\Delta\phi$ [rad]	$\Delta\psi$ [rad]	Δd [m]	h	E_{max}
$\pm\,\pi/10$	$\pm\,\pi/10$	$\pm\,0.4$	4	20

different scenes with all five objects, which are incrementally cleared by a human removing one object at a time, were recorded resulting in a total of sixty observed scenes which contained at least two objects each. For each observed scene we executed our grasping pipeline using the differently synthesized grasp sets. In these sets of experiments we evaluate all grasps up to the motion planning module, but do not select or execute a grasp since we aim to evaluate how many of the synthesized grasps are feasible (i. e., reachable by a collision-free path) under different conditions. The predefined parameters for the box constraints, palm rotation discretization and energy threshold used in Algorithm 8 are summarized in Table 6.1.

The outcome of the first experiment is visualized in Fig. 6.8. It is clear that pinch grasps are much more likely to be feasible, even if only a single target object is in the robot's workspace. Many enveloping grasps are rejected because they necessitate large opening angles resulting in bulky gripper silhouettes for which no collision free approach trajectories can be found.

Table 6.2: *Grasp Synthesis - Results Cluttered Scenes:* Quantities are represented as mean and 1-STD values

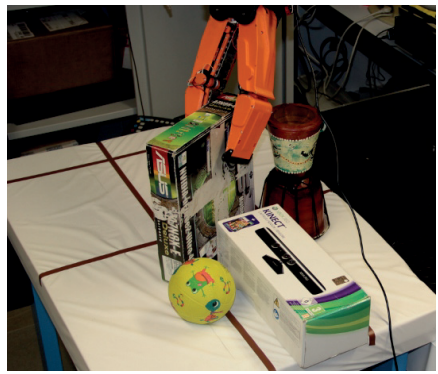
	$\lvert\overline{G}_P\rvert$/obj	$\lvert\overline{G}_E\rvert$/obj	$\lvert\overline{G}_M\rvert$/obj
5 obj	7.5 ± 12.3	1.3 ± 3.4	3.3 ± 8.0
4 obj	4.7 ± 6.2	1.4 ± 3.2	2.1 ± 3.5
3 obj	5.1 ± 8.9	1.5 ± 2.5	2.1 ± 4.2
2 obj	8.4 ± 9.3	4.0 ± 5.7	4.6 ± 8.0

Table 6.2 shows the feasible grasps per object depending on the number of objects in the cluttered scene and the chosen synthesis flavor (pinch, envelope or mixed grasps). We note that two out of the 15 scenes containing all five objects yielded an exceptionally high number of feasible pinch grasps, which biased the according entry in Table 6.2. This is due to the fact, that the number of feasible grasps found also significantly depends on the location of the objects in the robot's workspace which favored pinch grasps in these cases. Nevertheless, it is evident that the difficulty of finding feasible enveloping grasps increases with the amount of clutter, whereas it still possible to obtain pinch grasps with multiple objects in the scene. Also, the set of mixed grasps does not perform significantly better than enveloping grasps. These results provide a strong motivation to exclusively use dexterous initial pinch grasps, coupled with a strategy for subsequent improvement of robustness such as using active surfaces to pull the object into a firm enveloping grasp.

6.3.3 Grasp Execution using Active Surfaces

For a proof of concept evaluation of the suggested methods, we used the platform in Fig. 6.1 to incrementally clear one of the cluttered scenes using the described grasping pipeline and utilizing the set of synthesized pinch grasps G_P. Initial grasps were performed by thresholding the current absorption of the gripper's closing actuator. No current feedback was available for the belts on the fingers on this platform. Therefore, after an initial pinch grasp was acquired, the belt movements responsible for pulling the object into an enveloping grasp were triggered by an operator. In this fashion, the robot was able to retrieve all five objects needing six attempts, one object was dropped during the lift phase. An example of the robot using the pull-in strategy in a tabletop scenario is depicted in Fig. 6.9(a).

Additionally, we conducted a preliminary evaluation on the Parcelrobot platform which is described in Section 2.4. Here, we used a simple interaction control routine which was triggered after the platform reached the preplanned grasp configuration. At the time of conducting these experiments, it



(a)

(b)

Figure 6.9: *Grasp Execution - Test Runs:* (a) Shown is an example of the gripper using the active surfaces in a tabletop scenario to retrieve an object via pulling it towards its palm. (b) The gripper on the Parcelrobot platform retrieves a box autonomously.

was only possible to threshold the gripper actuators current consumptions and no direct current control was available. Therefore, we experimentally verified appropriate thresholds and used position control with predefined setpoints in the aforementioned grasp routine. An example of the Parcelrobot platform retrieving an object is illustrated in Fig. 6.9(b).

6.4 Discussion

In this chapter, we contribute a data-driven solution to the problem of finding palm poses and joint configurations which are appropriate to grasp objects with an underactuated gripper. The main concept is to simplify grasp synthesis and to entrust the specific kinematic/transmission structure of such an underactuated device with robust grasp execution. To this end, we adapt a well-established optimization based synthesis scheme [53, 54] to the specifics of such grasping devices. Heuristics describing grasp strategies observed in humans are incorporated via constraints which are imposed to the underlying optimization

problem. Furthermore, we investigate how active surfaces in form of belts on the fingers can aid in the grasp execution process, by manipulating an object in order to pull it into a robust enveloping grasp starting from an initial dexterous pinch grasp. We provide a numerical evaluation of the proposed grasp synthesis scheme in a real-world scenario and conduct proof-of-concept test runs on robot platforms featuring an underactuated gripper.

Depending on the number and configuration of objects in an observed scene, generating motion plans for many pre-planned grasps in the employed feasible-first manner can result in significant time delays. It would be interesting to cast the problem of finding a motion plan for the full arm-gripper chain as an optimization problem to be solved online. To that end, one could employ trajectory optimization techniques [111, 112] and describe simple target object geometries via constraints in order to solve the grasp synthesis and motion planning problems simultaneously.

Chapter 7
Reactive Hand Motion Planning and Control

Grasp contact point synthesis
Palm pose synthesis
Hand joint configuration synthesis
Hand motion planning and control

So far, we addressed the question of *where* to move a grasping device to allow for a successful grasp. For complex articulated hands, such as the anthropomorphic Shadow Robot platform utilized in the Handle project, also the issue of *how* to move the grasping device poses a challenge, since it necessitates to coordinate the motion of many DoF. To this end, we propose a reactive real-time motion generation framework which is based on DS whose parameters are learned from human demonstrations. In an extension, we augment the concept with a MPC scheme which generates locally optimal motions and allows to incorporate auxiliary tasks such as obstacle avoidance.

7.1 Introduction

On the frontier between motion planning and control DS have emerged as a popular way to encode desired movement behaviors in form of state transition policies [75, 113]. Here, opposed to strictly following pre-planned paths or using spline-based methods [114, 115], motions are generated reactively which provides robustness to perturbations occurring during execution.

In order to generate appropriate motion patterns for a targeted robotic system, the underlying DS parameter estimation problem[1] is commonly solved by

[1] Also referred to as parameter identification, nonlinear regression or data fitting [116].

providing data examples specifying desired transitions from given initial to final states. One way to provide experimental data is to record movements of a human expert in a TbD setting [117]. Another possibility is to create data artificially, *e. g.*, in form of smooth minimum-jerk trajectories [118] or as the pre-computed solutions of optimal control problems [119].

The choice of an appropriate DS for motion generation is typically guided by the ability of the underlying model to generalize over the provided examples while guaranteeing certain structural properties, and their potential to express coupling between the dynamics of different subsystems. Also, in order to facilitate the parameter estimation problem, simple models are often preferred. Especially in an imitation learning setting where the provided demonstrations are usually relatively sparse, it might happen that the behavior of the DS in unexplored parts of the state space is unexpected/undesirable. A classical approach for dealing with this problem is to enforce certain structural properties of the DS such as Global Asymptotic Stability (GAS), ensuring that the state is guaranteed to (at least) converge to the global equilibrium point. One shortcoming of such an approach is that it does not state any preference about the behavior of the system in relation to the demonstrations.

Since the considered DS constitute policies over the state space whose state evolution is guaranteed to converge, they can be seen as global planners which always reach their goal in the absence of obstacles [78]. In the context of reactive planning schemes, obstacles are typically dealt with locally – often by modeling them with repelling potential fields as suggested by Khatib [75].

This work originates from efforts related to modeling and generation of grasping movements, based on demonstrations of taxonomic grasps [120], for the anthropomorphic Shadow Hand platform which is shown in Fig. 2.2(b). Including the two wrist joints, the hand comprises 20 controlled DoF. Even under consideration of possible dimensionality reduction techniques [54, 121], this requires a model capable of dealing with a substantial number of DoF. Another desideratum is the ability to incorporate multiple demonstrations since, even for the same grasp type, grasping motions can exhibit fundamentally different dynamics (*e. g.*, when starting the movement from an open and closed hand configuration). In this work we suggest an approach using a dynamical system described by Ordinary Differential Equations (ODE) to encode demonstrations provided by a user. The method incorporates the DMP concept proposed by Ijspeert et al. [24, 122]. The contributions in this chapter are the following:

(i) We extend the DMP concept to learning of separate DS corresponding to multiple demonstrations which allows to better capture a motion's actual underlying dynamics. The corresponding parameter estimation is carried out using nonlinear optimization (instead of the usually used linear approximation) which reduces the number of parameters necessary to achieve a good fit to the provided demonstrations.

(ii) For real-time motion generation and control, we employ online optimization and introduce a linear receding horizon MPC scheme, which is based

on a convex combination of the learned DS ensuring predictable behavior over the state space. Opposed to the usage of explicit DS as in related works [122, 70, 74, 71], our formulation is able to account for spatial and temporal constraints to incorporate additional considerations such as obstacle avoidance.

This chapter is structured as follows: below, we formalize the tackled problem before we introduce our DMP formulation in Section 7.3. In Section 7.4 we suggest a method to combine multiple DS online in order to generalize over multiple demonstrations and introduce our MPC scheme for obstacle avoidance. Next, we use simulations and test runs with a robotic hand to evaluate the proposed approach in Section 7.5 before we discuss the results in Section 7.6.

7.2 Problem Description and Assumptions

Nomenclature

	Indices
i	Trajectory point index, $i \in \{1, \dots, m\}$
j	Demonstration index, $j \in \{1, \dots, d\}$
b	Gaussian basis function index, $b \in \{1, \dots, n\}$
a	DoF index, $a \in \{1, \dots, f\}$
r	Preview window index, $r \in \{1, \dots, p\}$
k	Discrete time index, $k \in \mathbb{Z}_+$
h	Hyperplane index, $h \in \mathbb{Z}_+$
	General
$\mathbf{q}$	Joint configuration, $\mathbf{q} = [q_1, \dots, q_f]^\mathsf{T}$
$\boldsymbol{x}$	State vector, $\boldsymbol{x} = [q,\ \dot{q}]^\mathsf{T}$
$\bar{\mathbf{q}}$	Discretized demonstration, $\bar{\mathbf{q}} = [\bar{q}_1, \dots, \bar{q}_m]^\mathsf{T}$
$\bar{t}$	Dilated time, $\bar{t} \in [0,\ 1]$
$\boldsymbol{\Phi}(\cdot)$	Dynamical Movement Primitive, $\boldsymbol{\Phi} : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^2$
s	Phase variable, $s \in \mathbb{R}$
$u(\cdot)$	Forcing function, $u : \mathbb{R} \to \mathbb{R}$
$\Psi_b(\cdot)$	b-th GBF, $\Psi_b : \mathbb{R} \to \mathbb{R}$
$\boldsymbol{w}$	GBF weights, $\boldsymbol{w} = [w_1, \dots, w_n]^\mathsf{T}$
$\mathbf{p}$	GBF centers and widths, $\mathbf{p} = [c_1, \sigma_1, \dots, c_n, \sigma_n]^\mathsf{T}$
ϵ	Basis function limit at $s = 1$, $\epsilon \in \mathbb{R}_+$
τ	Motion duration, $\tau \in \mathbb{R}_+$
$\mathbf{A}$	Continuous system matrix, $\mathbf{A} \in \mathbb{R}^{2\times2}$
$\mathbf{B}$	Continuous input matrix, $\mathbf{B} \in \mathbb{R}^2$
$\bar{\mathbf{A}}$	Discrete state transition matrix, $\bar{\mathbf{A}} \in \mathbb{R}^{2\times2}$
$\bar{\mathbf{B}}$	Discrete control matrix, $\bar{\mathbf{B}} \in \mathbb{R}^2$
$\kappa,\ \nu$	Penalty coefficients, $\kappa \in \mathbb{R}_+$, $\nu \in \mathbb{R}_+$
$\mathbf{C}$	Selection matrix, $\mathbf{C} \in \mathbb{R}^2$
$(\mathbf{H},\ \boldsymbol{e})$	State constraints, $\mathbf{H} \in \mathbb{R}^{c\times f}$, $\boldsymbol{e} \in \mathbb{R}^c$
o	Number of hyperplanes defining an obstacle, $o \in \mathbb{Z}_+$

Our goal is to develop a reactive motion generation system whose output trajectories resemble given demonstrations and which allows to incorporate state constraints for auxiliary targets such as obstacle avoidance. To this end, we learn movement primitives by fitting the parameters of dynamical systems, described as a set of ODE with a single global attractor point, to experimental data provided in form of multiple point-to-point trajectories in either joint or task-space. The state evolution of these dynamical systems, obtained by integrating from a given initial state, describes motion profiles which then can be converted to motor commands for the targeted platform by a low-level tracking controller. Important requirements are the ability to account for inherently different dynamics in the demonstrations and ensuring predictable behavior in regions of the state space which were not covered by the demonstrations. Also, a model structure not suffering from the curse of dimensionality is necessary, since we aim at platforms with a substantial number of DoF.

For convenience, and without loss of generality, all definitions regarding dynamical systems and their respective states are stated under the assumption of an implicit change of variable, such that the equilibrium point of the considered system is at the origin [123]. A demonstrated point-to-point trajectory is given as position, velocity and acceleration vectors $\bar{q}$, $\dot{\bar{q}}$, $\ddot{\bar{q}} \in \mathbb{R}^m$ sampled at m discrete points in time. The trajectory is rescaled on a time interval between zero and one, i. e., $\bar{t}_i \in [0, 1]$, $i = 1, \ldots, m$, in order to make different trajectories comparable. In accordance with the above assumption regarding the change of variable, the trajectory is shifted to converge at the origin, i. e., $\bar{q}_m = 0$. For simplicity of notation we assume that each trajectory is sampled with the same number m of points and that the same number d of demonstrations is provided for each DoF, although these are not explicit requirements of the proposed methods. Although we present our approach for motion generation in configuration space, it is equally applicable in operational space.

7.3 Learning Dynamical Movement Primitives

In this Section we first show, for one DoF, how to learn a motion primitive from a single demonstration by solving a NLP. Subsequently, we extend the formulation to account for multiple demonstrations which allows to encode fundamentally different dynamics for the same DoF.

7.3.1 Encoding a Single Demonstration

The motion of one DoF, corresponding to a given demonstration, is encoded in a DS $\mathbf{\Phi} : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^2$ formulated as the ODE

$$\dot{\boldsymbol{x}}(t) = \mathbf{\Phi}\left(\boldsymbol{x}(t), s(t); \boldsymbol{w}, \mathbf{p}\right),$$

depending on parameters $\boldsymbol{w}$ and $\mathbf{p}$, the state $\boldsymbol{x}(t) \in \mathbb{R}^2$, and a phase variable $s(t) \in \mathbb{R}$. The phase variable provides a convenient way to scale time in order

to modify the duration of the resulting motion. Its evolution is governed by the following simple dynamics

$$\frac{ds}{dt} = \dot{s} = 1/\tau, \tag{7.1}$$

where the scalar constant τ determines the movement's duration. The DS, together with the phase variable driving it constitutes a DMP. Synchronized motions across multiple DoF, each of which is associated with a separate DS, are achieved by using a common phase variable $s(t)$. A DS consists of a linear mass-spring-damper excited by a nonlinear input $u(s) : \mathbb{R} \to \mathbb{R}$, which is often referred to as a forcing function. As in [24], we choose to represent the forcing function as a weighted sum of n Gaussian Basis Functions (GBF) with weights $\boldsymbol{w} = [w_1, \ldots, w_n]^T$, respective centers $c_b \in [0,\ 1]$ and widths $\sigma_b \geqslant 0$ which are collected in the vector $\mathbf{p} = [c_1, \sigma_1, \ldots, c_n, \sigma_n]^T$. The system $\boldsymbol{\Phi}(\boldsymbol{x}(t), s(t); \boldsymbol{w}, \mathbf{p})$ is given by

$$\underbrace{\begin{bmatrix}\dot{q}\\ \ddot{q}\end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix}0 & 1\\ \alpha/\tau^2 & \beta/\tau\end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix}q\\ \dot{q}\end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix}0\\ 1/\tau^2\end{bmatrix}}_{\mathbf{B}} u(s) \tag{7.2}$$

$$u(s) = \sum_{b=1}^{n} \Psi_b(s; c_b, \sigma_b) w_b, \tag{7.3}$$

where $\alpha \in \mathbb{R}_+$ and $\beta \in \mathbb{R}_+$ are predefined such that critical damping is enforced and $\Psi_b = \exp\left(-0.5(s - c_b)^2/\sigma_b^2\right)$. In the original DMP framework [24], the phase variable s is governed by converging dynamics and used to scale the inputs u in order to guarantee GAS. In our formulation this is not required since we compute the parameters of the DS by solving an optimization problem in which we enforce appropriate constraints to ensure GAS as shown in Section 7.3.2.

To generate a motion, s is reset to zero and the DS in (7.2) is integrated from a given initial state. When s reaches one, the forcing terms $u(s)$ become negligible. The time evolution of the phase variable, and thus the movement duration, is governed by τ. Our choice of the system in (7.1) governing the evolution of the phase variable was made for simplicity. The use of alternative canonical systems is possible but would not qualitatively change the results.

7.3.2 Parameter Estimation via Nonlinear Programming

Learning a DMP amounts to estimating the GBF parameters $\boldsymbol{w}$ and $\mathbf{p}$ of the forcing function $u(s)$ in (7.3). This is a nonlinear problem which is usually tackled by fixing the nonlinear parameters in $\mathbf{p}$ according to some heuristics (*e. g.*, uniform Gaussian widths σ_b and equidistantly spaced centers c_b). Here,

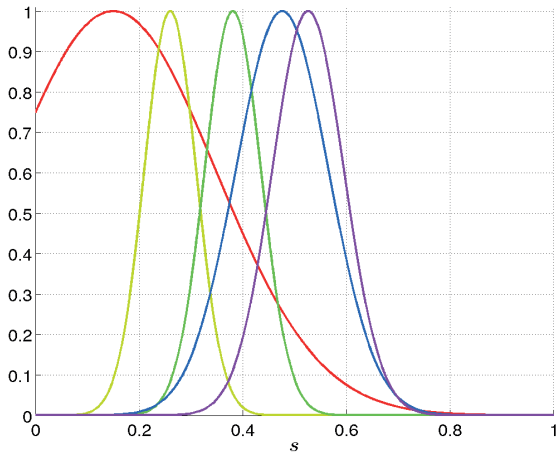


Figure 7.1: *DMP - Gaussian Basis Functions:* Shown are $n = 5$ GBF Ψ_b obtained via solving (7.4) for the demonstration in Fig. 7.2. The widths decrease with increasing s according to the constraint $\sigma_b \leqslant \hat{\sigma}(1 - c_b)$ in (7.5), ensuring negligible magnitudes of $u(s)$ for $s > 1$.

in a first step, we formulate a NLP in order to fit the parameters for a single system $\boldsymbol{\Phi}(\mathbf{x}, s; \boldsymbol{w}, \mathbf{p})$ to a provided demonstration. The goal is to learn forcing terms u such that the system resembles the dynamics of the demonstration. This is achieved by minimizing the squared L_2 norm of the acceleration residual between the demonstrated data and the output generated by the model. The corresponding constrained nonlinear least squares problem is given below[2]

$$\underset{\boldsymbol{w}, \mathbf{p}}{\text{minimize}} \quad \frac{1}{2} \sum_{i=1}^{m} \left(\mathbf{C}\boldsymbol{\Phi}\left(\bar{\mathbf{x}}_i, \bar{s}_i; \boldsymbol{w}, \mathbf{p}\right) - \ddot{\bar{q}}_i \right)^2 \tag{7.4}$$

subject to (7.5)

$$\begin{aligned} \sigma_b &\leqslant \hat{\sigma}(1 - c_b), & b &= 1, \ldots, n \\ 0 &\leqslant c_b \leqslant 1, & b &= 1, \ldots, n \\ \Delta c &\leqslant c_b - c_{b-1}, & b &= 2, \ldots, n, \end{aligned}$$

where $\bar{\mathbf{x}}_i = [\bar{q}_i, \dot{\bar{q}}_i]^T$ and $\bar{s}_i = \bar{t}_i$ due to the time scaling of the demonstrations as stated in Section 7.2. $\mathbf{C} = [0, 1]$ is a selection matrix and $\Delta c \in \mathbb{R}$, $0 \leqslant \Delta c \leqslant 1/n$ is a constant limiting the minimum distance between the centers of basis functions in order to prevent overlapping. The scalar $\epsilon \in \mathbb{R}$, $0 < \epsilon \ll 1$ can be used to arbitrary limit the value of the basis functions at the end of the interval $s \in [0, 1]$, i. e., $\Psi_b(1) \leqslant \epsilon$, $\forall b$, which ensures GAS. To this end,

[2] This problem is not convex and thus, in general, only a local minimizer will be found.

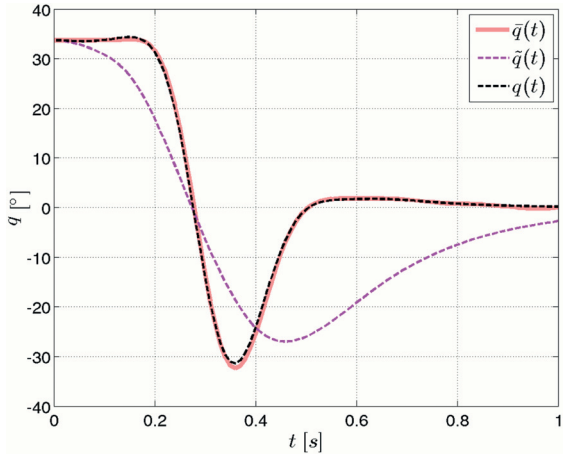


Figure 7.2: *DMP - Comparison:* Shown is the reproduction ability of the DS in (7.2) whose parameters are fitted by solving (7.4), compared to a DS using equidistantly spaced GBF and uniform basis function widths. The result was generated by integrating the respective systems from the initial state $\tilde{\mathbf{x}}(0)$ of the demonstration. The demonstration $\bar{\mathrm{q}}(\mathrm{t})$ is denoted in pink, the dashed black line represents the position curve $\mathrm{q}(\mathrm{t})$ yielded by our DS, the dashed magenta line shows the result obtained from the DS with predefined nonlinear parameters $\mathbf{p}$ ($\tilde{\mathrm{q}}(\mathrm{t})$ was generated with the code accompanying [122]). In both cases, $\mathrm{n} = 5$ basis functions were used.

$\hat{\sigma} = \sqrt{-0.5/\log(\epsilon)}$ corresponds to the width of a basis function centered at zero. To provide the solver with a feasible initial guess, the problem above is solved with fixed basis functions centers and widths which reduces (7.4) to a Quadratic Programming (QP) problem. Here, the n initial centers $\tilde{\mathrm{c}}_\mathrm{b}$ are equidistantly spaced on the interval $s \in [0, 1]$ and the associated widths are located on the corresponding constraint in (7.5) such that $\tilde{\sigma}_\mathrm{b} = \hat{\sigma}(1 - \tilde{\mathrm{c}}_\mathrm{b}), \forall \mathrm{b}$.

An example of the parameters $\mathbf{p}$ obtained by solving (7.4) is shown in Fig. 7.1. The corresponding demonstration, along with a comparison to a solution generated with heuristically fixed nonlinear parameters is depicted in Fig. 7.2. Evidently, by including the nonlinear parameters $\mathbf{p}$ in the decision variables, a better fit can be obtained as shown in Section 7.5.1.

7.3.3 Encoding Multiple Demonstrations

In the next step, the goal is to fit (for one DoF) the forcing terms of d dynamical systems to d provided demonstrations such that the j-th DS encodes the dynamics in the vicinity of the j-th demonstration. One could simply use the NLP in (7.4) to identify $\boldsymbol{w} \in \mathbb{R}^{\mathrm{n}}$ and $\mathbf{p} \in \mathbb{R}^{2\mathrm{n}}$ separately for each DS which would amount to estimate $3\mathrm{dn}$ parameters. Instead, we reformulate (7.4) such

that the nonlinear basis function parameters $\mathbf{p}$ are shared among the d dynamical systems while the j-th DS has associated linear parameters $\boldsymbol{w}_j$. The objective function becomes

$$\underset{\boldsymbol{w}_1,\ldots,\boldsymbol{w}_d,\mathbf{p}}{\text{minimize}} \; \frac{1}{2}\sum_{j=1}^{d}\left\{\sum_{i=1}^{m}\left(\mathbf{C}\boldsymbol{\Phi}_j\left(\bar{\mathbf{x}}_{j,i},\bar{s}_i;\boldsymbol{w}_j,\mathbf{p}\right)-\ddot{\bar{q}}_{j,i}\right)^2\right\} \tag{7.6}$$

and the problem is subjected to the constraints in (7.5). The above formulation allows a fit with $n(d+2)$ parameters and was used for the evaluation in Section 7.5. The concept of sharing basis functions between motion generators is similar as the one used by Rückert and d'Avella in [124], where it is put in the context of muscular synergies.

7.4 Real-time Control with Movement Primitives

In this section we first discuss how to form a new implicit DS based on a locally optimal combination of the previously learned systems (each of which corresponds to a demonstration). Then, we proceed to derive our MPC scheme with state constraints.

7.4.1 Generating Locally Optimal Motions

Let $\mathbf{x}_j[k]$ denote the state at time t_k obtained by integrating $\boldsymbol{\Phi}_j(\mathbf{x}_j, s)$ from $t = t_1$ to $t = t_k$ starting from $\bar{\mathbf{x}}_j(0)$ (i. e., from the initial state of the j-th demonstration). Our approach makes dual use of the dynamical systems. First, the set of *reference states* collected in the columns of the matrix

$$\mathbf{R}[k] = [\mathbf{x}_1[k],\ldots,\mathbf{x}_d[k]] \in \mathbb{R}^{2\times d} \tag{7.7}$$

provides, at each time t_k, a representation of the corresponding demonstration encoded in $\boldsymbol{\Phi}_j(\mathbf{x}_j, s)$. Second, we formulate a movement primitive comprising a new DS where the forcing term is formed as a convex combination of individual inputs $u_j[k]$ associated with the systems $\boldsymbol{\Phi}_j(\mathbf{x}_j, s)$

$$\dot{\mathbf{x}}[k] = \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k]^T\boldsymbol{\lambda}[k], \tag{7.8}$$

where $\mathbf{u}[k] = [u_1[k],\ldots,u_d[k]]^T$ and $\boldsymbol{\lambda}[k] = [\lambda_1[k],\ldots,\lambda_d[k]]^T$. Here, $\mathbf{A}$ and $\mathbf{B}$ are the same as in (7.2). Equation (7.8) describes an implicit DS, where by implicit we imply that the system is not given in closed form. Rather, its definition relies on an online solution of an optimization problem. Here, the coefficients $\lambda_j[k]$ are recomputed at every time step t_k by minimizing the residual

$$\Delta\mathbf{x}[k] = \mathbf{x}[k] - \mathbf{R}[k]\boldsymbol{\lambda}[k] \tag{7.9}$$

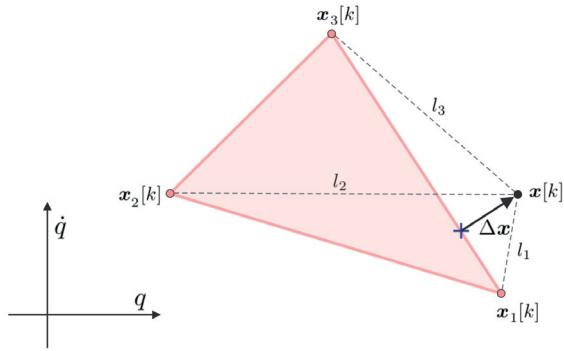


Figure 7.3: *DMP - Convex Combination*: The shaded area represents the convex hull over the reference states in $\mathbf{R}[k]$, the projection $\mathbf{R}[k]\boldsymbol{\lambda}[k]$ of the current state $\mathbf{x}[k]$ onto this convex hull is indicated by the cross, $\Delta\boldsymbol{x}$ signifies the projection residual.

of the projection of the current state $\mathbf{x}[k]$ onto the convex hull over the current reference states in the columns of $\mathbf{R}[k]$ in (7.7) as shown in Fig. 7.3. The associated minimization problem is formalized as the QP below

$$\begin{aligned} \underset{\boldsymbol{\lambda}[k]}{\text{minimize}} \quad & \|\Delta\mathbf{x}[k]\|_H^2 + \kappa\mathbf{l}[k]^T\boldsymbol{\lambda}[k] \\ \text{subject to} \quad & \mathbb{1}^T\boldsymbol{\lambda}[k] = 1, \\ & \boldsymbol{\lambda}[k] \geqslant 0, \end{aligned} \tag{7.10}$$

where the elements $l_j[k] = \|\mathbf{x}[k] - \mathbf{x}_j[k]\|_2$ of the vector $\mathbf{l}[k] = [l_1[k], \ldots, l_d[k]]^T$ describe the euclidean distances of the reference states to the current states, $\kappa \geqslant 0$ is a (small) scalar and $\mathbb{1}$ is an appropriately dimensioned column vector of ones. The second term in the objective function in (7.10) is added in order to resolve the redundancy between multiple equivalent solutions for $\boldsymbol{\lambda}[k]$ which can occur if the residual $\Delta\mathbf{x}$ is zero. We define $\|\boldsymbol{z}\|_H^2 = \boldsymbol{z}^T\mathbf{H}\boldsymbol{z}$ for some $\boldsymbol{z} \in \mathbb{R}^z$ and a positive semi-definite and symmetric matrix $\mathbf{H} \in \mathbb{R}^{z\times z}$. Let the vector $\boldsymbol{\lambda}^\star = [\lambda_1^\star, \ldots, \lambda_d^\star]^T$ denote a solution of (7.10), i. e., $\boldsymbol{\lambda}[k] = \boldsymbol{\lambda}^\star$. The coefficients $\lambda_j^\star$ are recomputed only at discrete steps k according to (7.10) and are assumed to be constant within the time window $[t_k, t_{k+1}]$. In order to characterize the behavior of the newly formed DS in (7.8) we formulate the following proposition.

Proposition 7.1 (System behavior).
The projection residual $\Delta\mathbf{x}[k]$ *converges onto the convex hull over the reference states in* $\mathbf{R}[k]$ *with dynamics governed by the matrix* $\mathbf{A}$

$$\Delta\dot{\mathbf{x}}[k] = \mathbf{A}\Delta\mathbf{x}[k], \quad t \in [t_k, t_{k+1}].$$

If the convex hull over the states in $\mathbf{R}[k]$ *contains the current state* $\mathbf{x}[k]$*, the projection residual* $\Delta\mathbf{x}[k]$ *is zero and the next state* $\mathbf{x}[k+1]$ *will be a convex combination of the reference states in* $\mathbf{R}[k+1]$*,* i. e.*,*

$$\mathbf{x}[k+1] = \mathbf{R}[k+1]\boldsymbol{\lambda}^{\star}.$$

A proof of the above proposition is given in Appendix B. Proposition 7.1 summarizes a key concept in this work. The DS in (7.8) accounts for different dynamics encoded from multiple demonstrations while exhibiting a predictable behavior over the whole state space. This is achieved by encoding a representation of the underlying demonstrations by means of the DS itself. States inside the convex hull of the reference states evolve according to a convex combination of the references. The matrix $\mathbf{A}$ in (7.8) governs the evolution for states outside the convex hull over the references and can be tuned according to the application. As in the original DMP framework [122], arbitrary many DoF can be synchronized via a common phase variable s.

7.4.2 DMP-based Model Predictive Control

A remaining question is how appropriate the trajectories generated by the policy in (7.8) are in the presence of obstacles which are not known a priori. One could imagine an example were the combination of the reference dynamics leads to collisions with unforeseen obstacles. Opposed to existing approaches [68, 70, 71, 72] which use statistical learning techniques to combine pre-learned DMP in order to generalize to novel situations, the suggested method provides a straightforward way to incorporate state constraints. Since the approach allows to modify the motion generating system in (7.8) at each time step, we suggest an alternative way of handling obstacles using model predictive control under a set of spatial and temporal polyhedral constraints which are designed to lead the system around a given (potentially moving) obstacle.

To start, let us note that the matrix formed by the product $\mathbf{B}\mathbf{u}[k]^{\mathsf{T}} \in \mathbb{R}^{2\times d}$ in (7.8) can loose rank (e. g., towards the end of a motion when the elements of $\mathbf{u}[k]$ vanish) and that the vector $\boldsymbol{\lambda}[k]$ is bound by the convex constraints in (7.10). Therefore, to ensure the ability to satisfy additional state constraints, we augment the system in (7.8) with an auxiliary control input $\tilde{\lambda}$ and discretize to obtain

$$\mathbf{x}[k+1] = \bar{\mathbf{A}}\mathbf{x}[k] + \bar{\mathbf{B}}\boldsymbol{\mu}[k]^{\mathsf{T}}\boldsymbol{\gamma}[k], \tag{7.11}$$

where $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ are state transition matrix and control matrix of the discrete system, $\boldsymbol{\mu}[k] = [\mathbf{u}[k], 1]^{\mathsf{T}} \in \mathbb{R}^{d+1}$ and $\boldsymbol{\gamma}[k] = \left[\boldsymbol{\lambda}[k], \tilde{\lambda}[k]\right]^{\mathsf{T}} \in \mathbb{R}^{d+1}$ denotes the augmented control vector. Next, we want to predict the residual of the projection of the current state on the reference states p step forwards in time.

Inserting the augmented system in (7.11) in the residual formulation in (7.9) and performing recursion yields

$$\underbrace{\begin{bmatrix} \Delta x[k] \\ \Delta x[k+1] \\ \vdots \\ \Delta x[k+p] \end{bmatrix}}_{\Delta X \in \mathbb{R}^{2(p+1)}} = \begin{bmatrix} \bar{A}^0 \\ \bar{A}^1 \\ \vdots \\ \bar{A}^p \end{bmatrix} x[k] + Z \underbrace{\begin{bmatrix} \gamma[k] \\ \gamma[k+1] \\ \vdots \\ \gamma[k+p] \end{bmatrix}}_{\Gamma \in \mathbb{R}^{(d+1)(p+1)}}, \tag{7.12}$$

where the matrix $Z \in \mathbb{R}^{2(p+1)\times(d+1)(p+1)}$ is given as

$$Z = \begin{bmatrix} -R[k] & 0 & \dots & \dots & 0 \\ \bar{B}\mu[k] & -R[k+1] & 0 & \dots & 0 \\ \bar{A}\bar{B}\mu[k] & \bar{B}\mu[k+1] & -R[k+2] & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ \bar{A}^{p-1}\bar{B}\mu[k] & \bar{A}^{p-2}\bar{B}\mu[k+1] & \dots & \bar{B}\mu[k+p-1] & -R[k+p] \end{bmatrix}.$$

Without consideration of additional state constraints, we can now formulate a receding horizon MPC scheme as the following optimization problem which needs to be solved at every time step t_k

$$\begin{aligned} \underset{\Gamma}{\text{minimize}} \quad & \|\Delta X[k]\|_H^2 + \kappa l[k]^T \lambda[k] + \nu \sum_{r=0}^{p} \tilde{\lambda}[k+p]^2 \\ \text{subject to} \quad & \mathbb{1}^T \lambda[k+r] = 1, \quad r = 0, \dots, p, \\ & \lambda[k+r] \geqslant 0, \quad r = 0, \dots, p. \end{aligned} \tag{7.13}$$

Here, compared to the previous formulation in (7.10) where only the current projection residual at time t_k is optimized, the minimization is carried out over a temporal preview window of p steps according to (7.12). The penalty factor ν in (7.13) is chosen to be large in order to suppress the auxiliary control inputs $\tilde{\lambda}[k+r]$ since their role is to deviate the system only if additional constraints need to be obeyed as discussed below.

7.4.3 State Constraints for Obstacle Avoidance

Here, the goal is to avoid obstacles in state space. For simplicity, we only consider constraints on the positions $q = [q_1, \dots, q_f]^T$ of the f state vectors x_α according to (7.13), although velocity constraints on $\dot{q} = [\dot{q}_1, \dots, \dot{q}_f]^T$ can be handled in the same fashion. To ensure convexity, we only consider linear state constraints of the form $h^T q + e \leqslant 0$ which facilitates the solution of the underlying optimization problem. Here, $h \in \mathbb{R}^f$ is a unit normal vector and e is a scalar offset.

In our framework, one DS in (7.11) is learned to guide each DoF. At this point, the state evolutions of these DS are independent, there is only a potential temporal coupling via shared phase variables s driving the inputs $\mathbf{u}[k]$ in (7.11). Here, we couple f systems in (7.11) via the state constraints which requires extending the MPC scheme in (7.13) as shown below

$$\underset{\boldsymbol{\Gamma}_1,\ldots,\boldsymbol{\Gamma}_f}{\text{minimize}} \quad \sum_{a=1}^{f} \left(\|\Delta \mathbf{X}_a[k]\|_H^2 + \kappa \mathbf{l}_a[k]^T \boldsymbol{\lambda}_a[k] + \nu \sum_{r=0}^{p} \tilde{\lambda}_a[k+r]^2 \right) \tag{7.14}$$

subject to

$$\begin{aligned} \mathbb{1}^T \boldsymbol{\lambda}_a[k+r] &= 1, \quad r = 0,\ldots,p, \quad a = 1,\ldots,f \\ \boldsymbol{\lambda}_a[k+r] &\geqslant \mathbf{0}, \quad r = 0,\ldots,p, \quad a = 1,\ldots,f \\ \mathbf{H}[k+r]\mathbf{q}[k+r] + \boldsymbol{e}[k+r] &\leqslant 0, \quad r = 1,\ldots,p. \end{aligned} \tag{7.15}$$

Here, c constraints are considered at a given time step in the preview window. The matrix $\mathbf{H}[k+r] = [\mathbf{h}_1[k+r],\ldots,\mathbf{h}_c[k+r]]^T \in \mathbb{R}^{c\times f}$ collects the constraint normals, vector $\boldsymbol{e}[k+r] = [e_1[k],\ldots,e_c[k]]^T$ holds the corresponding offsets. To account for the coupling introduced by the predicted configurations $\mathbf{q}[k+r]$ in (7.15), we have to consider the evolutions of each of the f states $\boldsymbol{x}_a[k+r]$

$$\begin{bmatrix} \boldsymbol{x}_a[k+1] \\ \boldsymbol{x}_a[k+2] \\ \vdots \\ \boldsymbol{x}_a[k+p] \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{A}}^1 \\ \bar{\mathbf{A}}^2 \\ \vdots \\ \bar{\mathbf{A}}^p \end{bmatrix} \boldsymbol{x}[k] + \boldsymbol{\Upsilon}\boldsymbol{\Gamma}_a,$$

where the matrix $\boldsymbol{\Upsilon} \in \mathbb{R}^{2p\times(d+1)(p+1)}$ is given as

$$\boldsymbol{\Upsilon} = \begin{bmatrix} \bar{\mathbf{B}}\boldsymbol{\mu}[k] & 0 & \ldots & \ldots & 0 \\ \bar{\mathbf{A}}\bar{\mathbf{B}}\boldsymbol{\mu}[k] & \bar{\mathbf{B}}\boldsymbol{\mu}[k+1] & 0 & \ldots & \vdots \\ \vdots & \vdots & \vdots & 0 & \vdots \\ \bar{\mathbf{A}}^{p-1}\bar{\mathbf{B}}\boldsymbol{\mu}[k] & \bar{\mathbf{A}}^{p-2}\bar{\mathbf{B}}\boldsymbol{\mu}[k+1] & \ldots & \bar{\mathbf{B}}\boldsymbol{\mu}[k+p-1] & 0 \end{bmatrix}.$$

Note that the state coupling is only introduced in the constraints of (7.14), not the objective function which is simply a sum over the objectives in (7.13). Thus, if no constraint in (7.15) is active at a given time step, the resulting behavior is identical to the one produced by the uncoupled scheme in (7.13) and resembles the learned trajectories. Only if constraints in (7.15) are active, the auxiliary controls $\tilde{\lambda}_a[k+r]$ cause deviations in order to satisfy these constraints. Considering the choice of objective function in (7.14) and assuming a long enough preview horizon, the stability of the proposed controller can be guaranteed [125]. In the tests reported in Section 7.5.3, we experimented with horizon lengths of $p = 5$ and $p = 10$ time steps which led to stable behavior.

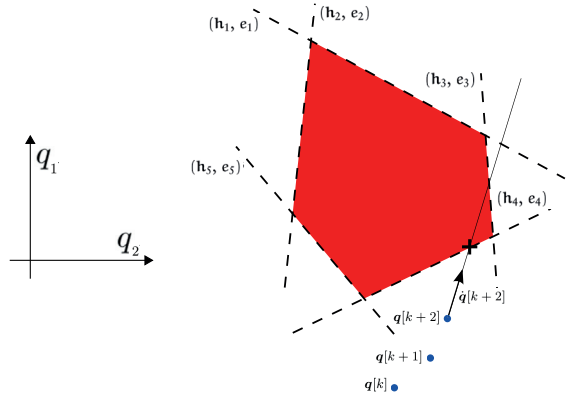


Figure 7.4: *Obstacle Avoidance - Constraints:* Shown is an example in a 2-dimensional configuration space, (i. e., $f = 2$) with a preview window size of $p = 2$. The velocity ray at $\mathbf{q}[k+2]$ intersects hyperplane $(\mathbf{h}_4, e_4)$ at the point $\mathbf{q}[k+2] + \xi\dot{\mathbf{q}}[k+2]$, which is indicated with a cross. Thus, $(\mathbf{h}_4, e_4)$ is added to the constraints in (7.15).

A remaining issue is how to extract appropriate spatio-temporal constraints for obstacle and self-collision avoidance from the robot's environment. This is an open research question and is out of the scope of this work. Previous works suggest heuristics based on simplified pre-planned paths [82, 83]. Here, we only consider a point-robot model and introduce a simple heuristics in order to be able to verify our MPC scheme in Section 7.5.3. We assume that an obstacle is represented as a convex polytope in $\mathcal{H}$-representation (see Section 2.1), given as a set of bounding hyperplanes $\{(\mathbf{h}_h, e_h)\}$, $h = 1, \ldots, o$. At each time step t_k we want to determine whether to augment $(\mathbf{H}[k+p], \boldsymbol{e}[k+p])$ in the optimization problem in (7.14) with a new constraint at the end of the preview horizon, i. e., at time t_{k+p}. To this end, we formulate the following LP

$$\begin{aligned} &\underset{\xi \in \mathbb{R}}{\text{minimize}} \quad \xi && (7.16)\\ &\text{subject to} \\ &\mathbf{h}_h^T \left(\mathbf{q}[k+p] + \xi\dot{\mathbf{q}}[k+p]\right) + e_h \geqslant 0, \quad h = 1, \ldots, o \\ &\xi \geqslant 0, \end{aligned}$$

which projects the state $\mathbf{q}[k+p]$ along the ray corresponding to the velocity $\dot{\mathbf{q}}_{k+p}$ onto the obstacle as illustrated in Fig. 7.4. If the above LP is feasible, i. e., the state evolution "heads towards" the obstacle, the hyperplane containing the projection forms a new constraint in (7.15).

The computational load of the presented MPC scheme at each time step k consists of integrating the canonical system in (7.1) and the fd dynamical systems in (7.3), where f is the number of DoF and d denotes the number of DS

Figure 7.5: *Motion Generation - Data Acquisition:* An Immersion Cyberglove-18 was used to record joint angles during grasp motions at a sample rate of 30 Hz. Starting from open and closed initial hand configurations, grasps according to the taxonomy in [120] were performed on cylindrical objects.

(each corresponding to a demonstration) per DoF. Furthermore, the solution of f QP's according to (7.14) and an LP according to (7.16) is required.

7.5 Evaluation

In this section we evaluate, by means of simulations and test runs on the Shadow Robot platform, the application of the suggested methods to offline learning of motion primitives from experimental data and the usage of these primitives for real-time motion control. To this end we used a sensorized glove to record taxonomic grasps on two cylindrical objects with different diameters as illustrated in Fig. 7.5. We chose to evaluate the approach on the following nine grasp types according to [120]: Tripod, Parallel Extension, Palmar Pinch, Large Diameter, Small Diameter, Lateral, Precision Sphere, Power Sphere and Inferior Pincer. The recordings were made while starting from open and closed initial hand configurations respectively. The Shadow hand's joint angles were obtained via a linear regression mapping from the glove's sensor space to the robot's joint angle space. As the goal is to model grasp joint motions using DMP driven by a common phase variable s, the corresponding demonstrations have to live on a common time interval. Thus, all trajectories were segmented from the time a non-zero velocity was detected at a joint, until all joints stopped moving. Furthermore, the demonstrated trajectories were smoothed by means of a linear least squares regression and numerically differentiated to obtain velocities and accelerations. After rescaling and shifting, as described in Section 7.2, the trajectories were re-sampled with a number of $\mathrm{m} = 100$ points each. A standard PC equipped with 6 GB memory and a 3.40 GHz Intel i7-2600 CPU was used to generate the presented results.

Table 7.1: *Motion Generation - Parameters:* Predefined parameters used for the evaluation presented in Section 7.5.1

n	α	β	ϵ	Δc	$\mathbf{H}$	κ
5	-132.5	-23	10^{-4}	0.05	diag(100, 1)	0

Table 7.2: *Motion Generation - Reproduction Results:* Quatities are represented as mean and 1-STD values. Δq and $\Delta \dot{q}$ denote the position/velocity errors between the demonstrated data and the trajectories reproduced by the locally optimal DMP combination in (7.8), t denotes the computation time needed to solve (7.6). Superscripts $(\cdot)^F$ and $(\cdot)^O$ denote whether the results were generated with fixed or optimized basis function parameters.

N	$\|\|\Delta q^F\|\|_2^2$	$\|\|\Delta q^O\|\|_2^2$	$\|\|\Delta \dot{q}^F\|\|_2^2$	$\|\|\Delta \dot{q}^O\|\|_2^2$	t^F [s]	t^O [s]
3	2770.6±6802.7	454.7±1552.1	$3.8{\cdot}10^5 \pm 9.0{\cdot}10^5$	$5.3{\cdot}10^4 \pm 1.6{\cdot}10^5$	0.005±0.026	0.5±0.4
5	415.4 ±1094.6	78.1 ±170.5	$9.6{\cdot}10^4 \pm 2.5{\cdot}10^5$	4694.3±1.1·10^4	0.006±0.002	2.1±4.0
7	98.4 ±217.4	70.1 ±150.7	$2.0{\cdot}10^4 \pm 4.8{\cdot}10^4$	2772.6±5470.8	0.006±0.002	6.2±9.4
9	74.4 ±160.8	69.7 ±150.5	7841.9±1.8·10^4	2399.9±4931.8	0.006±0.002	30.9±63.9
1	70.4 ±152.4	69.3 ±149.4	4089.4±9328.8	2358.6±4798.7	0.006±0.002	58.8±92.13
13	69.1 ±149.9	68.8 ±149.1	2751.2±6087.1	2355.0±4927.6	0.007±0.004	26.9±50.0
15	68.8 ±149.3	68.9 ±149.1	2411.8±5184.8	2350.8±4786.2	0.007±0.003	62.2±79.4

Table 7.3: *Obstacle Avoidance - Parameters:* Predefined parameters used for the examples in Section 7.5.3

n	α	β	ϵ	Δc	$\mathbf{H}$	κ	ν	d
5	-132.5	-23	10^{-4}	0.05	diag(1, 1)	10^{-6}	10^4	6

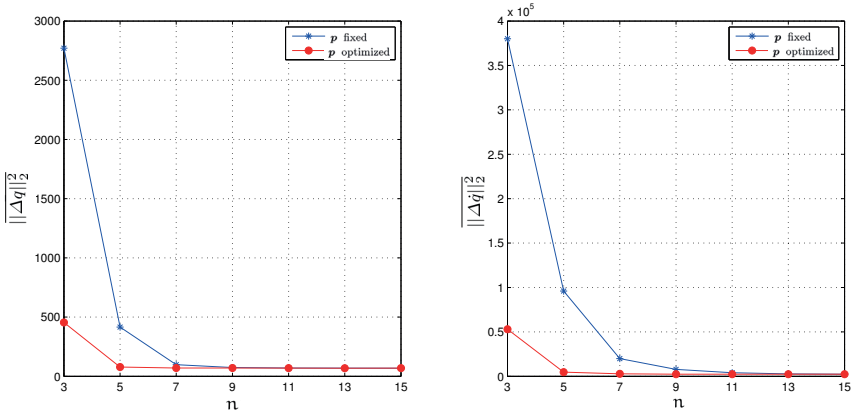
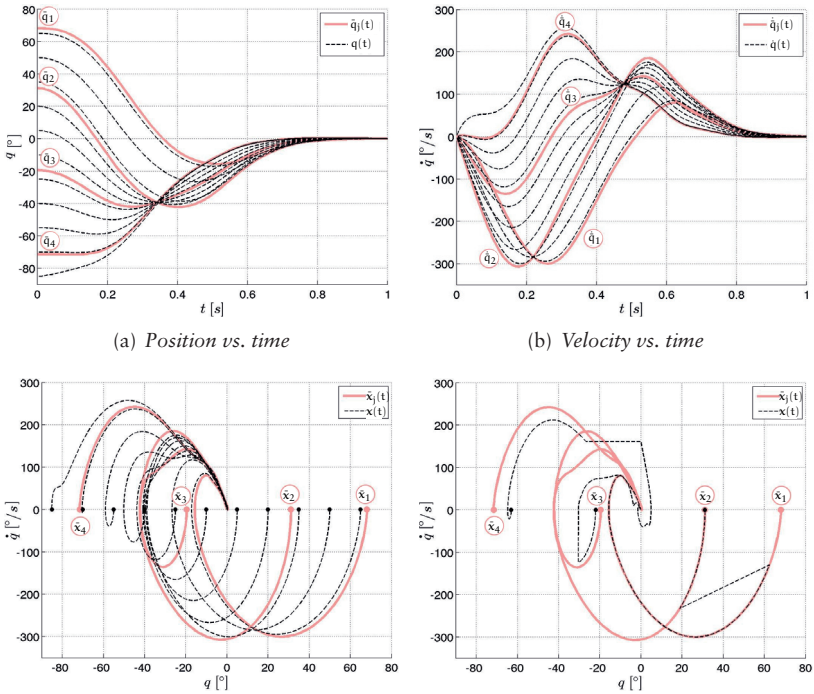


Figure 7.6: *Motion Generation - Reproduction Quality:* Illustrated are the MSE which describe the deviation from the trajectories produced by the systems in (7.8) from the learning data. Shown are the MSE for position (left) and velocity (right) for different numbers n of basis functions and for the GBF parameters $\mathbf{p}$ in (7.4) fixed/optimized.

7.5.1 Reproduction and Generalization Capabilities

Here, the aim is to assess the introduced offline DMP learning scheme in (7.6). For the $f = 20$ DoF of the Shadow hand we used, for each of the aforementioned nine grasp types, demonstrated trajectories to estimate the free parameters of 20 motion primitives in (7.8) as described in Section 7.3.3. Thus, a total of 180 trajectories were used for the evaluation, the utilized fixed parameters are summarized in Table 7.1. The constrained nonlinear least squares problems in (7.6) were solved with a Sequential Quadratic Programming (SQP) algorithm, utilizing the ACADO Toolkit [126].

In order to quantify the reproduction capabilities of the learned DMP, we reproduced the demonstrated trajectories by integrating (7.8) starting from the same initial values as the corresponding demonstrations. We experimented with different numbers n of basis functions in (7.3) and compared to results generated with DMP learned with fixed basis function parameters as in [122]. The resulting position and velocity Mean Square Errors (MSE), as well as the computation times for solving (7.6) for different numbers n of basis functions are summarized in Table 7.2. Additionally, the position/velocity MSE are also depicted in Fig. 7.6. It is evident that, for small numbers of employed basis functions, the nonlinear learning scheme vastly outperforms linear learning with fixed basis function parameters. Also, the mean computation times for solving (7.6) while including the basis function parameters in the decision variables are within reasonable bounds (e. g. for $n = 7$ basis functions, the mean computation time is 6.2s).



(a) *Position vs. time* (b) *Velocity vs. time*

(c) *Phase plane generalization* (d) *Phase plane - disturbance compensation*

Figure 7.7: *Motion Generation - Generalization:* Dashed lines represent the trajectories obtained by simulating the dynamical system in (7.8), describing the tripod grasp motion primitive for the MCP joint, starting from different initial conditions. The system was parametrized via the demonstrated trajectories denoted in pink. Demonstrations $j = 1$ and $j = 3$ are associated with grasps made on a cylindrical object with diameter 65 mm starting from closed and open initial hand configurations respectively, $j = 2$ and $j = 4$ correspond to grasps on an object with diameter 33 mm. (a) and (b) depict the curves for position and velocity, the corresponding phase diagram is shown in (c). The behavior of the system in the presence of disturbances is depicted in (d). After evolving unperturbed initially, the system was subjected to disturbances in position, velocity and a combined disturbance respectively.

To gauge the generalization capabilities of the learned models for the considered point-to-point movements, we performed simulations by initializing our combined motion primitive formulation in (7.8) from different initial states. Exemplary, the results for the dynamical system describing the flexion/extension motions of the middle fingers Metacarpophalangeal (MCP) joint (the MCP joints connect the proximal phalanges of the fingers to the palm) during a tri-

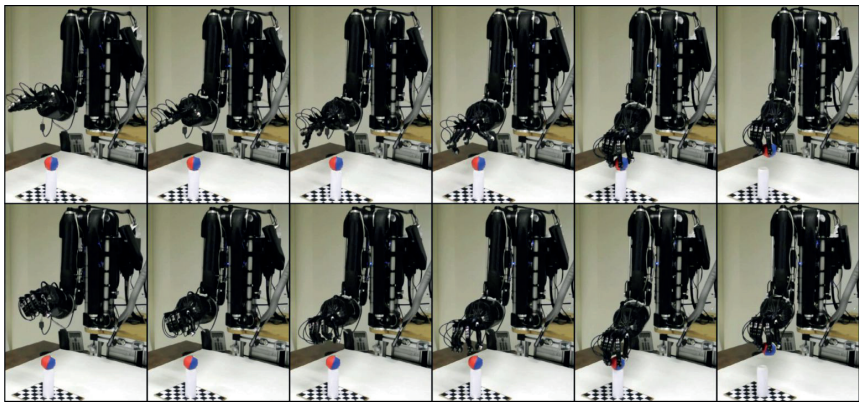


Figure 7.8: *Motion Generation - Test Runs:* Shown is the Shadow Robot hand/arm platform executing tripod grasp primitives triggered from different initial configurations. Synchronized finger joint movements are generated by means of integrating motion primitives corresponding to (7.8) which are driven by a common phase variable. Top row: Starting from an open hand configuration; bottom row: starting from a closed hand configuration.

pod grasp are shown in Fig. 7.7. Depicted are the obtained position, velocity and phase plane curves. As argued in Section 7.4, for states evolving inside the convex hull over the reference states the distance ratio to the references is governed by the convex combination coefficients computed as a solution of (7.10). States outside the convex hull over the references are attracted towards this convex hull according to dynamics governed by the matrix $\mathbf{A}$ in (7.8). It can be seen that the model can reproduce the demonstrated trajectories with high fidelity while exhibiting a deterministic behavior in regions of the state space not covered by the demonstrations.

Furthermore, we analyzed the behavior of the model in the presence of state disturbances. We investigated separate position and velocity disturbances as well as a combined disturbance. When, at time t_k, the system is perturbed inside the convex hull of the reference states, the update of the convex combination coefficients according to (7.10) at time t_{k+1} adjusts the future evolution of the system according to the reference states at time t_{k+1}. An example is shown in Fig. 7.7(d), where a trajectory was started at the initial state $\bar{\mathbf{x}}_2(0)$ corresponding to the second demonstration and is pushed onto the reference trajectory associated with the first demonstration. After adjusting the combination coefficients in the next time step, the system continues to evolve according to $\bar{\mathbf{x}}_1$. Disturbances with states resulting outside the convex hull of the references again cause the system to converge towards the projection onto this convex hull with dynamics as specified in (7.8).

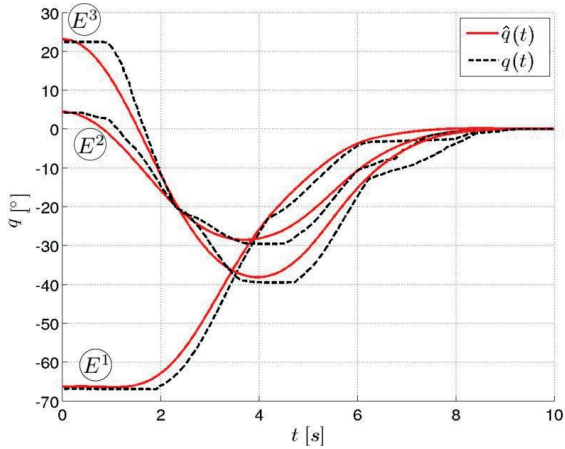
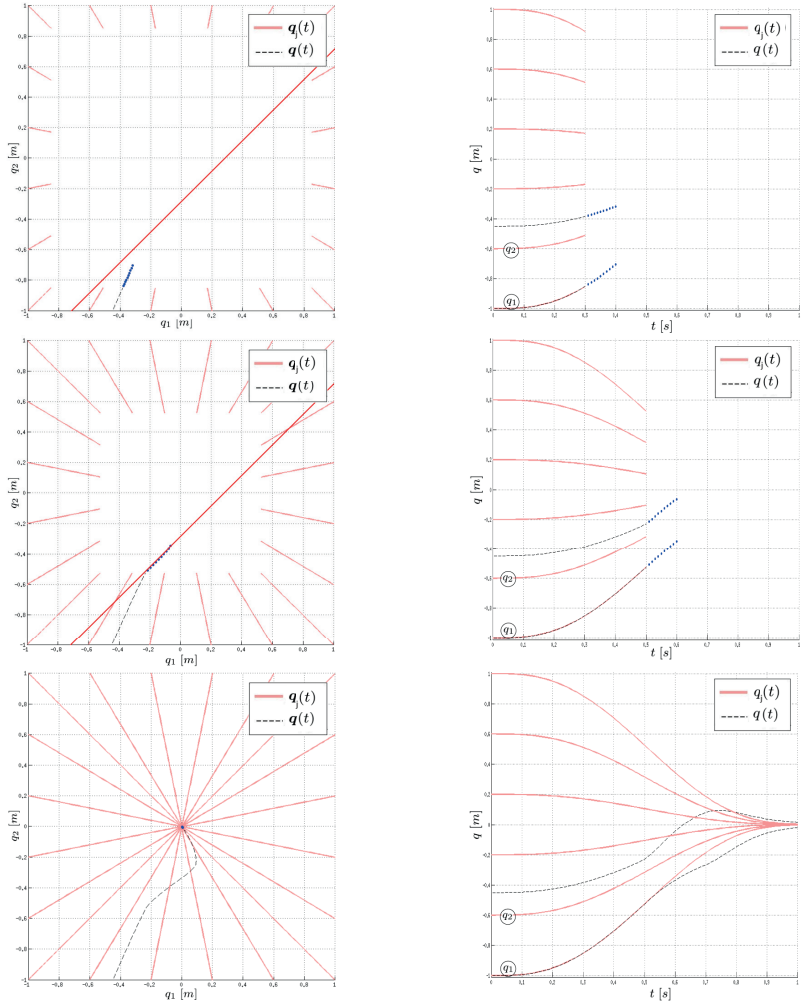


Figure 7.9: *Motion Generation - Tracking Control:* Tracking controller set-points $\hat{q}(t)$ and position curves $q(t)$ for the MCP joint starting from open (experiment E^1), pronated (E^2) and closed (E^3) initial hand configuration. Motion duration $\tau = 10s$.

7.5.2 Verification on the Shadow Robot Platform

Here, the goal is to demonstrate the feasibility of the developed motion primitives for real-time motion generation and control. A standard laptop was used to control the Shadow Robot platform via the Robot Operating System (ROS) framework at 100 Hz. We use the aforementioned nine grasp types which were considered feasible for the specific mechanical structure of the Shadow hand. The learned motion primitives were used to generate motion profiles for the 20 DoF of the Shadow hand. Appropriate motion profiles for the 4 DoF of the arm were generated with the ROS joint spline trajectory controllers, such that hand and arm motion comprised the same duration. Desired final hand/arm configurations were obtained via kinesthetic teaching and subsequently adding an empiric small increment to the joint values in order to ensure sufficient squeezing of the object. Then, the motion primitives for the hand joints were triggered from initial conditions corresponding to open, pronated and closed hand configurations respectively which allowed to successfully execute synchronized grasp and subsequent lifting motions as shown in Fig. 7.8. Here, the arm joints were moved between predefined start and final positions. One encountered problem was that the ROS messaging system introduced unacceptable feedback delays and that the available low-level position PID tracking control was of limited quality. Thus, the test runs were carried out in an open-loop fashion, i. e., the primitives were only used for instantaneous planning of reference profiles between the given start and end positions without considering state feedback. Figure 7.9 shows, again exemplary for the MCP joint, the controller set points



(a) *Configuration space* (b) *Position vs. time*

Figure 7.10: *Obstacle Avoidance - Constraint Satisfaction:* Shown is the evolution of a 2-dimensional system driven by two primitives in (7.11) controlled by the MPC scheme in (7.14) with a preview window size of p = 10. The positions $\mathbf{q}$ computed by the controller are depicted with dashed black lines, the pink lines indicate the evolution of the encoded demonstrated position curves $\mathbf{q}_j$. The constraint vanishes after t = 0.7s which allows the system to converge to its equilibrium. (a) shows the behavior in configuration space, (b) depicts the according position curves.

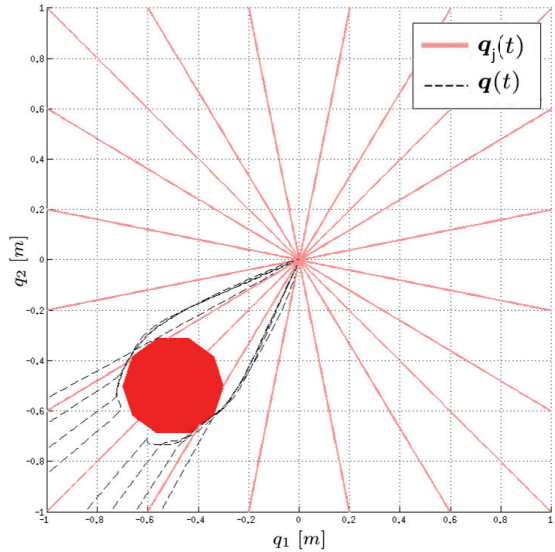


Figure 7.11: *Obstacle Avoidance - Behavior*: Shown is the obstacle avoidance behavior of the system controlled by (7.14) with a preview window size of $p = 5$, while using the heuristic according to (7.16) in order to extract constraints. The system was initialized with different start states, positions $\mathbf{q}$ computed by the controller are depicted with dashed black lines, the pink lines indicate the evolution of the encoded demonstrated position curves $\mathbf{q}_j$.

obtained from integrating the DS and the resulting position curves generated by the tracking controller. Despite the obvious limitations in the low-level control, the grasping tasks were conducted successfully.

7.5.3 Obstacle Avoidance

Here, we want to discuss the behavior of the MPC scheme formalized in (7.14) under the influence of state constraints. To this, end we give two illustrating examples in a 2-dimensional (i. e., $f = 2$) configuration space. The fixed parameters which were used in (7.14) are summarized in Table 7.3. We use two primitives in (7.11), each of which was learned from the same $d = 6$ synthetically generated examples of minimum-jerk trajectories. Figure 7.10 shows the evolution of the system at different points in time under influence of a single spatial constraint which is active during part of the motion. The controller in (7.14) computes auxiliary control inputs $\tilde{\lambda}[k + r]$ such as to obey the constraint.

A second example is depicted in Figure 7.11. Here, we employ the heuristic for the automatic extraction of appropriate constraint hyperplanes, as it was presented in Section 7.4.3, to avoid a convex obstacle. Shown are the trajectories generated when starting from different points in the obstacle space.

7.6 Discussion

In this chapter, we presented a solution to the hand motion control problem for articulated hands. Our approach uses empirical data in form of demonstrated motions in order to parametrize dynamical systems for movement generation via nonlinear optimization. For real-time control, we introduce a MPC scheme based on a locally optimal combination of the previously learned DS. This results in a deterministic behavior in state regions which were not explored during the demonstrations. Furthermore, the demonstrations can be reproduced with high fidelity while relying on a comparatively small number of parameters. We assessed the introduced method by means of parametrizing the proposed model from demonstrations of grasp movements and subsequent simulations and test runs with the Shadow Robot platform. Our approach affords the flexibility to modify the control inputs of the implicit system used for motion generation at each time step, which allows to incorporate state constraints to account for auxiliary tasks such as obstacle avoidance. The use of embedded optimization for addressing the online obstacle avoidance problem is a promising approach already heavily utilized in other scientific fields. This work is a first step towards a reactive online planning/control scheme.

Chapter 8
Conclusion

This dissertation discussed some of the challenges which are encountered when developing systems for autonomous robot grasp synthesis and execution. Starting from contact-level grasp synthesis and reaching reactive hand motion planning and control, this thesis has attempted to investigate the main ingredients which are necessary to enable a robotic platform to robustly grasp specified target objects in a consistent manner. The work was carried out in the scope of two EU-FP7 projects whose settings were fundamentally different: one employed an underactuated gripper in a logistics scenario, the other aimed at endowing a fully anthropomorphic platform with skills resembling those exhibited by humans. Nevertheless, it showed that the overarching idea of this thesis – combining empirical with analytical approaches based on optimality criteria – is applicable to common sub-problems arising in both of these projects. In this concluding chapter, we will summarize the most important high-level contributions, discuss the limitations of the proposed methods and chart possible improvements and future work.

8.1 Contributions

A full list of contributions made in this thesis is presented in Section 1.3. Here, the focus is on highlighting the three most important achievements of this work.

The first notable contribution in this dissertation is the extension of the grasp wrench space, a fundamental concept in contact-level grasp analysis, to the exertable wrench space which characterizes how well a family of grasps can resist disturbances. We utilize this concept for the computation of Independent Contact Regions (ICR), which represent a grasp family, and propose construction algorithms which are largely based on the solution of convex optimization problems and can be efficiently parallelized as argued in Chapter 4. Empirical user-input can be incorporated in form of demonstrated prototype grasps and the synthesis process considers tasks with a clear physical meaning. Also provided is an open-source `C++` implementation of the introduced algorithms

which is evaluated numerically, as well as in several application examples ranging from grasp ranking over visually guided teleoperation and interactive grasp transfer to a support role in finger gait planning (see Chapter 5).

A second major contribution is the data-driven solution to the grasp synthesis problem on a palm pose and hand joint configuration level, which is described in Chapter 6. Here, we adapt a leading optimization-based scheme and incorporate heuristics describing grasp strategies observed in humans. This is achieved by imposing appropriate constraints on the underlying optimization problem. We provide a numerical evaluation of the proposed grasp synthesis approach in a real-world scenario and conduct proof-of-concept test runs on robotic platforms featuring an underactuated grasping device with active surfaces.

Last but not least, a solution to the grasp motion control problem for articulated hands is given in Chapter 7. Our approach uses empirical data in form of demonstrated motions in order to parametrize dynamical systems for movement generation via nonlinear optimization. For real-time control, we form locally optimal combinations of the previously learned dynamical systems, which results in a deterministic behavior in state regions which were not explored during the demonstrations. The method is evaluated in test runs on an anthropomorphic hand/arm platform. Furthermore, we suggest an extension to a Model Predictive Control (MPC) scheme which relies on embedded optimization to achieve auxiliary tasks such as obstacle avoidance.

8.2 Limitations

It needs to be noted that some of the developed methods have limitations which need to be considered when integrating them in applied systems.

Computing ICR as discussed in Chapters 4 and 5 is sensitive to noise in the contact normals which can pose problems if only imperfect object models reconstructed from range data are available. Possible remedies are a conservative choice of friction coefficient [103], or appropriate mesh smoothing techniques [14]. However, both of these methods might yield to overly restrictive results.

The grasping pipeline detailed in Section 6.2.2 suffers from the inherent problems of approaches which separate offline grasp synthesis from online motion planning: checking many pre-computed grasps for feasibility can incur significant time delays during execution and, because only a finite number of grasps can be pre-planned and stored for an object, no solutions might be found for solvable cases. Increasing the number of synthesized grasps increases the success probability but, in turn, increases the computational load during the online stage.

The primitive-based joint-level motion controllers presented in Chapter 7 do not explicitly account for potential self-collisions during movement. Applied to grasp motion control of anthropomorphic robot hands, the underlying as-

sumption is that the generated motions resemble demonstrated movements and therefore this does not pose a problem. Also, the dynamical movement primitive learning scheme in Section 7.3 has several free parameters which need to be tuned carefully to obtain good performance. A simple way to address this issue could be to employ a greedy parameter selection algorithm which stops when the fitting error is low enough. Also, there is room for improvement in the obstacle avoidance scheme in Section 7.5.3 which currently only works for dimensionless point-robots in simulation.

8.3 Future Research Directions

To keep the problems tractable, this thesis follows the common practice of seeing grasp synthesis and hand/manipulator motion planning as independent sub-problems. Considering the increasing level of computational power available on embedded systems and the significant improvement of numerical optimization techniques in recent years, it would be interesting to cast the problem of finding a motion plan for the full arm-gripper chain as an optimization problem to be solved online. To that end, one could employ contact-invariant trajectory optimization techniques [111, 112] and describe simple target object geometries via constraints in order to solve the grasp synthesis and motion planning problems simultaneously. Here, the idea would be to encode a high-level grasping/manipulation task as a cost function and utilize an optimization algorithm which uses models of the environment and the robot to plan a behavior. Solving the underlying optimization problem is computationally expensive and currently not in the realm of real-time. Building on the idea in Chapter 7, a possibility could be to optimize over a limited time preview window only via a MPC scheme [127] which sidesteps the curse of dimensionality since a policy is only generated at the last moment for states that are actually visited. Additionally, reoptimizing at each time step allows to incorporate state feedback which enables reactive behavior.

In the literature, the MPC-based trajectory optimization paradigm has only been employed in simulation so far. Application in a real-world grasping scenario with imperfect knowledge about the scene will incur additional challenges, since the precise target object pose and geometry is unknown in such a setting. We envision to augment the underlying optimization with local knowledge of the object geometry, which is acquired by tactile sensors after grasp contact is made to in order to adjust the initial grasp [128]. An early version of this concept was developed in Section 6.3.3, where active surfaces on the fingers of the gripper are used to improve the robustness of an acquired grasp by in-hand manipulation.

Appendix A
EWS Approximation via Prioritizing Contacts

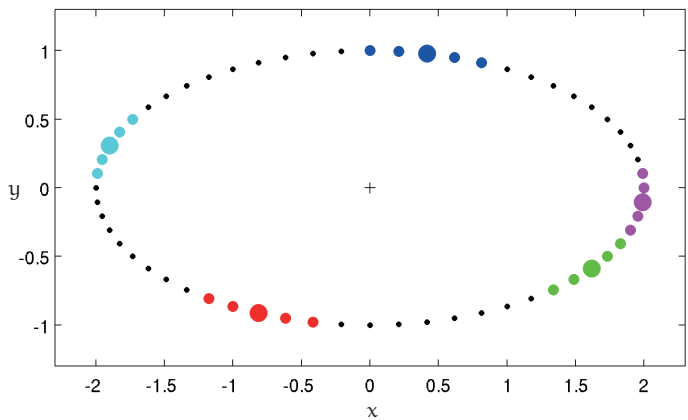


Figure A.1: *Predefined ICR*: Uniform regions $\mathcal{R}_i$ for a five-fingered grasp with frictionless contact constraints. Big dots characterize the prototype grasp's contact points $\mathbf{p}_{g_i}$. The depicted regions are used as an input for the EWS approximation example illustrated in Fig. A.2.

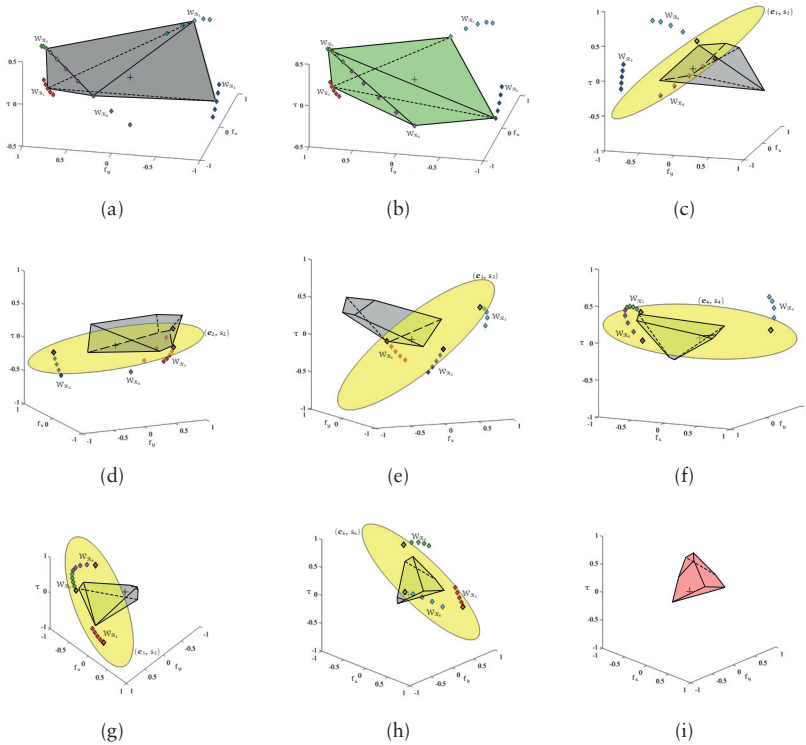


Figure A.2: *Sequential EWS Approximation*: Exemplary computations of hyperplanes according to Algorithm 7 for the ellipse shown in Fig. A.1, the considered TWS is represented by the zero-wrench. (a) Shown is the prototype grasp's wrench space $\mathrm{GWS}_{\mathrm{init}}$ and the wrench sets $\mathcal{W}_{\mathcal{R}_i}$ corresponding to the predefined regions $\mathcal{R}_i$. (b) A viable grasp whose corresponding GWS comprises a different face lattice than $\mathrm{GWS}_{\mathrm{init}}$, i.e., the polytope in (a) comprises a facet spanned by one wrench each from the sets $\mathcal{W}_{\mathcal{R}_3}$, $\mathcal{W}_{\mathcal{R}_4}$ and $\mathcal{W}_{\mathcal{R}_5}$ whereas the GWS in (b) does not. Thus, the grasps corresponding to the GWS in (a) and (b) are not similar according to Definition 4.6.1. (c)–(h) The EWS is approximated by gradually limiting $\mathrm{GWS}_{\mathrm{init}}$, the wrenches supporting $(\boldsymbol{e}_h, s_h)$ are depicted in bold frames. Hyperplane $(\boldsymbol{e}_1, s_1)$ in (c) does not fulfill Proposition 4.3, thus it is unnecessarily constrictive. The gray shaded polytope in (h) represents the final approximation of the EWS. (i) The actual EWS, yielded by a brute force computation according to (4.8).

Appendix B
Proof of Proposition 7.1

To prove Proposition 7.1 in Section 7.4.1 we consider for simplicity zero-order hold discretized systems, although the proof can be trivially extended to handle the continuous time case. The respective discretizations of the systems in (7.2) and (7.8) are

$$\begin{aligned} \boldsymbol{x}_j[k+1] &= \bar{\mathbf{A}}\boldsymbol{x}_j[k] + \bar{\mathbf{B}}\mathbf{u}_j[k] && \text{(B.1)} \\ \boldsymbol{x}[k+1] &= \bar{\mathbf{A}}\boldsymbol{x}[k] + \bar{\mathbf{B}}\mathbf{u}[k]^T\boldsymbol{\lambda}^*, && \text{(B.2)} \end{aligned}$$

where $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ are the respective state transition matrix and control matrix of the discrete system. Substituting (B.1) and (B.2) in (7.9) for time t_{k+1} results in

$$\Delta\boldsymbol{x}[k+1] = \bar{\mathbf{A}}\underbrace{(\boldsymbol{x}[k] - \mathbf{R}[k]\boldsymbol{\lambda}^*)}_{\Delta\boldsymbol{x}[k]}, \qquad \text{(B.3)}$$

which confirms the first part of Proposition 7.1.

Furthermore, we note that if the projection residual $\Delta\boldsymbol{x}[k]$ in (B.3) is zero, the state $\mathbf{x}[k]$ can be expressed as a convex combination of the reference states in the columns of $\mathbf{R}[k]$. Thus, for $\Delta\boldsymbol{x}[k] = 0$, we can rewrite (B.2) as

$$\begin{aligned} \mathbf{x}[k+1] &= \bar{\mathbf{A}}\underbrace{\mathbf{R}[k]\boldsymbol{\lambda}^*}_{\mathbf{x}[k]} + \bar{\mathbf{B}}\mathbf{u}[k]^T\boldsymbol{\lambda}^* \\ &= \mathbf{R}[k+1]\boldsymbol{\lambda}^* \end{aligned}$$

which concludes the proof.

References

[1] Sony Corporation. AIBO. http://www.sony.net/SonyInfo/News/Press_Archive/199905/99-046/index.html, 1999. [Accessed: 2014-04-08]. (Cited on page 1.)

[2] iRobot Corporation. iRobot Roomba® Vacuum Cleaning Robot. http://www.irobot.com/us/learn/home/roomba.aspx, 2014. [Accessed: 2014-04-08]. (Cited on page 1.)

[3] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994. (Cited on pages 1, 3, and 26.)

[4] M. T. Mason and J. K. Salisbury. *Robot hands and the mechanics of manipulation*. MIT Press, 1985. (Cited on page 1.)

[5] S. C. Jacobsen, J. E. Wood, D. F. Knutti, and K. B. Biggers. The UTAH/MIT dextrous hand: Work in progress. *International Journal of Robotics Research*, 3(4):21–50, 1984. (Cited on page 1.)

[6] M. Grebenstein, A. Albu-Schäffer, T. Bahls, M. Chalon, O. Eiberger, W. Friedl, R. Gruber, S. Haddadin, U. Hagn, H. Haslinger, R. Hoppner, S. Jorg, M. Nickl, A. Nothhelfer, F. Petit, J. Reill, N. Seitz, T. Wimbock, S. Wolf, T. Wusthoff, and G. Hirzinger. The DLR hand arm system. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 3175–3182, 2011. (Cited on pages 1 and 59.)

[7] A. Namiki, Y. Imai, M. Ishikawa, and M. Kaneko. Development of a high-speed multifingered hand system and its application to catching. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2666–2671, 2003. (Cited on page 2.)

[8] Barrett Technology Inc. The Barrett Hand. http://www.barrett.com/robot/products-hand.htm, 2014. [Accessed: 2014-04-08]. (Cited on page 2.)

[9] A. M. Dollar and R. D. Howe. The highly adaptive SDM hand: Design and performance evaluation. *International Journal of Robotics Research*, 29(5):585–597, 2010. (Cited on page 2.)

[10] R. Krug, T. Stoyanov, M. Bonilla, V. Tincani, N. Vaskevicius, G. Fantoni, A. Birk, A. J. Lilienthal, and A. Bicchi. Velvet fingers: Grasp planning and execution for an underactuated gripper with active surfaces. In *Proc. of the IEEE International Conference on Robotics and Automation (to appear)*, 2014. (Cited on pages 6 and 7.)

[11] R. Krug and D. N. Dimitrov. Representing movement primitives as implicit dynamical systems learned from multiple demonstrations. In *Proc. of the International Conference on Advanced Robotics*, pages 1–8, 2013. (Cited on page 6.)

[12] R. Krug, D. N. Dimitrov, K. Charusta, and B. Iliev. Prioritized independent contact regions for form closure grasps. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1797–1803, 2011. (Cited on page 6.)

[13] R. Krug, D. N. Dimitrov, K. Charusta, and B. Iliev. On the efficient computation of independent contact regions for force closure grasps. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 586–591, 2010. (Cited on page 6.)

[14] K. Charusta, R. Krug, T. Stoyanov, D. N. Dimitrov, and B. Iliev. Generation of independent contact regions on objects reconstructed from noisy real-world range data. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 1338–1344, 2012. (Cited on pages 6, 7, 57, and 98.)

[15] K. Charusta, R. Krug, D. N. Dimitrov, and B. Iliev. Independent contact regions based on a patch contact model. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 4162–4169, 2012. (Cited on pages 6, 7, 17, 30, and 57.)

[16] E. Berglund, B. Iliev, R. Palm, R. Krug, K. Charusta, and D. N. Dimitrov. Mapping between different kinematic structures without absolute positioning during operation. *Electronics Letters*, 48(18):1110–1112, 2012. (Cited on page 7.)

[17] E. Berglund, B. Iliev, R. Palm, R. Krug, K. Charusta, and D. N. Dimitrov. Mapping between different kinematic structures without absolute positioning. In *IEEE International Conference on Robotics and Automation, Workshop on Autonomous Grasping*, 2011. (Cited on page 7.)

[18] C. Borst, M. Fischer, and G. Hirzinger. Calculating hand configurations for precision and pinch grasps. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1553–1559, 2002. (Cited on page 9.)

[19] L. Han, J. C. Trinkle, and Z. X. Li. Grasp analysis as linear matrix inequality problems. *IEEE Transactions on Robotics*, 16(6):663–674, 2000. (Cited on page 10.)

[20] S. Chiaverini, B. Siciliano, and L. Villani. A survey of robot interaction control schemes with experimental comparison. *IEEE/ASME Transactions on Mechatronics*, 4(3):273–285, 1999. (Cited on page 10.)

[21] V. Klee. Convex polytopes and linear programming. In *Proc. of the IBM Scientific Computing Symposium: Combinatorial Problems*, pages 123–158, 1966. (Cited on pages 10, 36, and 42.)

[22] M. G. Ziegler. *Lectures on Polytopes*. Springer, 1995. (Cited on pages 10 and 35.)

[23] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer, 3rd edition, 2008. (Cited on pages 10 and 30.)

[24] A. J. Ijspeert, J. Nakanishi, and S. Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. In *Proc. of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1398–1403, 2002. (Cited on pages 12, 19, 76, and 79.)

[25] Shadow Robot Company. The Shadow Dextrous Hand. `http://www.shadowrobot.com/products/dexterous-hand/`, 2014. [Accessed: 2014-04-05]. (Cited on page 12.)

[26] V. Tincani, M. G. Catalano, E. Farnioli, M. Garabini, G. Grioli, G. Fantoni, and A. Bicchi. Velvet fingers: A dexterous gripper with active surfaces. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1257–1263, 2012. (Cited on pages 12 and 60.)

[27] V. Tincani, G. Grioli, M. G. Catalano, M. Garabini, S. Grechi, G. Fantoni, and A. Bicchi. Implementation and control of the velvet fingers: a dexterous gripper with active surfaces. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 2744–2750, 2013. (Cited on pages 12, 60, and 68.)

[28] A. Sahbani, S. El-Khoury, and P. Bidaud. An overview of 3d object grasp synthesis algorithms. *Robotics and Autonomous Systems*, 60(3):326–336, 2012. (Cited on page 17.)

[29] C. Borst, M. Fischer, and G. Hirzinger. Grasping the dice by dicing the grasp. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 4, pages 3692–3697, 2003. (Cited on page 17.)

[30] C. Borst, M. Fischer, and G. Hirzinger. Grasp planning: how to choose a suitable task wrench space. In *Proc. of the IEEE International Conference on Robotics and Automation*, volume 1, pages 319–325, 2004. (Cited on pages 17, 23, 26, and 28.)

[31] C. Ferrari and J. Canny. Planning optimal grasps. In *Proc. of the IEEE International Conference on Robotics and Automation*, volume 3, pages 2290–2295, 1992. (Cited on pages 17, 18, and 27.)

[32] C. Borst, M. Fischer, and G. Hirzinger. A fast and robust grasp planner for arbitrary 3d objects. In *Proc. of the IEEE International Conference on Robotics and Automation*, volume 3, pages 1890–1896, 1999. (Cited on page 17.)

[33] X. Zhu and J. Wang. Synthesis of force-closure grasps on 3-d objects based on the q distance. *IEEE Transactions on Robotics and Automation*, 19(4):669–679, 2003. (Cited on page 17.)

[34] A. Bicchi and V. Kumar. Robotic grasping and contact: a review. In *Proc. of the IEEE International Conference on Robotics and Automation*, volume 1, pages 348–353, 2000. (Cited on page 17.)

[35] V. D. Nguyen. Constructing force-closure grasps. *International Journal of Robotics Research*, 7(3):3–16, 1988. (Cited on page 17.)

[36] J. Ponce and B. Faverjon. On computing three-finger force-closure grasps of polygonal objects. *IEEE Transactions on Robotics*, 11(6):868–881, 1995. (Cited on page 17.)

[37] J. Ponce, S. Sullivan, A. Sudsang, J. D. Boissonnat, and J. P. Merlet. On computing four-finger equilibrium and force-closure grasps of polyhedral objects. *International Journal of Robotics Research*, 16(1):11–35, 1997. (Cited on pages 17 and 46.)

[38] N. S. Pollard. *Parallel methods for synthesizing whole-hand grasps from generalized prototypes*. PhD thesis, MIT, Department of Electrical Engineering and Computer Science, 1994. (Cited on pages 17, 23, 25, 27, 28, 30, 35, and 44.)

[39] N. S. Pollard. Closure and quality equivalence for efficient synthesis of grasps from examples. *International Journal of Robotics Research*, 23(6):595–614, 2004. (Cited on pages 17, 30, 33, 34, 35, 44, 48, and 54.)

[40] D. Ding, Y. H. Lee, and S. Wang. Computation of 3-d form-closure grasps. *IEEE Transactions on Robotics*, 17(4):515–522, 2001. (Cited on page 18.)

[41] M. A. Roa and R. Suárez. Computation of independent contact regions for grasping 3-d objects. *IEEE Transactions on Robotics*, 25(4):839–850, 2009. (Cited on pages 18, 33, 34, 43, 44, 45, and 46.)

[42] B. A. Dang-Vu, M. A. Roa, and C. Borst. Extended independent contact regions for grasping applications. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3527–3534, 2013. (Cited on pages 18, 44, 45, and 46.)

[43] Y. Li, J. L. Fu, and N. S. Pollard. Data-driven grasp synthesis using shape matching and task-based pruning. *IEEE Transactions on Visualization and Computer Graphics*, 13(4):732–747, 2007. (Cited on page 18.)

[44] C. Goldfeder, M. T. Ciocarlie, J. Peretzman, H. Dang, and P. K. Allen. Data-driven grasping with partial sensor data. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1278–1283, 2009. (Cited on page 18.)

[45] C. Goldfeder and P. K. Allen. Data-driven grasping. *Autonomous Robots*, 31(1):1–20, 2011. (Cited on pages 18 and 63.)

[46] S. Ekvall and D. Kragic. Learning and evaluation of the approach vector for automatic grasp generation and planning. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 4715–4720, 2007. (Cited on page 18.)

[47] J. Bohg and D. Kragic. Learning grasping points with shape context. *Robotics and Autonomous Systems*, 58(4):362–377, 2010. (Cited on page 18.)

[48] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen. Automatic grasp planning using shape primitives. In *Proc. of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1824–1829, 2003. (Cited on page 18.)

[49] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof. Grasp planning via decomposition trees. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 4679–4684, 2007. (Cited on page 18.)

[50] K. Huebner and D. Kragic. Selection of robot pre-grasps using box-based shape approximation. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1765–1770, 2008. (Cited on page 18.)

[51] D. Berenson and S. Srinivasa. Grasp synthesis in cluttered environments for dexterous hands. In *Proc. of the IEEE/RAS International Conference on Humanoid Robots*, pages 189–196, 2008. (Cited on page 18.)

[52] R. Diankov. *Automated Construction of Robotic Manipulation Programs*. PhD thesis, Carnegie Mellon University, Robotics Institute, 2010. (Cited on pages 18 and 64.)

[53] A. T. Miller and P. K. Allen. Graspit! a versatile simulator for robotic grasping. *Robotics & Automation Magazine, IEEE*, 11(4):110–122, 2004. (Cited on pages 18, 63, 64, and 72.)

[54] M. T. Ciocarlie and P. K. Allen. Hand posture subspaces for dexterous robotic grasping. *International Journal of Robotics Research*, 28(7):851–867, 2009. (Cited on pages 18, 61, 63, 72, and 76.)

[55] J. Bohg, A. Morales, T. Asfour, and D. Kragic. Data-driven grasp synthesis—a survey. *IEEE Transactions on Robotics*, 30(2):289–309, 2014. (Cited on page 18.)

[56] D. Berenson, R. Diankov, K. Nishiwaki, S. Kagami, and J. Kuffner. Grasp planning in complex scenes. In *Proc. of the IEEE/RAS International Conference on Humanoid Robots*, pages 42–48, 2007. (Cited on pages 18, 63, 64, and 68.)

[57] M. R. Dogar and S. S. Srinivasa. Push-grasping with dexterous hands: Mechanics and a method. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2123–2130, 2010. (Cited on page 18.)

[58] M. R. Dogar, K. Hsiao, M. T. Ciocarlie, and S. S. Srinivasa. Physics-based grasp planning through clutter. In *Robotics: Science and Systems VIII*, July 2012. (Cited on page 18.)

[59] A. Saxena, L. Wong, M. Quigley, and A. Y. Ng. A vision-based system for grasping novel objects in cluttered environments. In *Robotics Research*, pages 337–348. Springer, 2011. (Cited on page 18.)

[60] S. Schaal and C. G. Atkeson. Constructive incremental learning from only local information. *Neural Computation*, 10:2047–2084, 1997. (Cited on page 19.)

[61] A. J. Ijspeert, J. Nakanishi, P Pastor, H. Hoffmann, and S. Schaal. Dynamical movement primitives: Learning attractor models for motor behaviors. *Neural Computation*, 25:328–373, 2013. (Cited on page 19.)

[62] P. Kormushev, S. Calinon, and D. G. Caldwell. Robot motor skill coordination with EM-based reinforcement learning. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 3232–3237, 2010. (Cited on page 19.)

[63] F. Stulp and S. Schaal. Hierarchical reinforcement learning with movement primitives. In *Proc. of the IEEE/RAS International Conference on Humanoid Robots*, pages 231–238, 2011. (Cited on page 19.)

[64] F. Stulp, E. Theodorou, J. Buchli, and S. Schaal. Learning to grasp under uncertainty. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 5703–5708, 2011. (Cited on page 19.)

[65] J. Kober and J. Peters. Policy search for motor primitives in robotics. *Machine Learning*, 84(1-2):171–203, 2011. (Cited on page 19.)

[66] B. Nemec, R. Vuga, and A. Ude. Efficient sensorimotor learning from multiple demonstrations. *Advanced Robotics*, 27(13):1023–1031, 2013. (Cited on page 19.)

[67] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal. Learning and generalization of motor skills by learning from demonstration. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 763–768, 2009. (Cited on page 19.)

[68] T. Matsubara, S. Hyon, and J. Morimoto. Learning stylistic dynamic movement primitives from multiple demonstrations. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1277–1283, 2010. (Cited on pages 19 and 84.)

[69] S. Calinon, Z. Li, T. Alizadeh, N. G. Tsagarakis, and D. G. Caldwell. Statistical dynamical systems for skills acquisition in humanoids. In *Proc. of the IEEE/RAS International Conference on Humanoid Robots*, pages 323–329, 2012. (Cited on page 19.)

[70] A. Ude, A. Gams, T. Asfour, and J. Morimoto. Task-specific generalization of discrete and periodic dynamic movement primitives. *IEEE Transactions on Robotics*, 26(5):800–815, 2010. (Cited on pages 19, 77, and 84.)

[71] D. Forte, A. Gams, J. Morimoto, and A. Ude. On-line motion synthesis and adaptation using a trajectory database. *Robotics and Autonomous Systems*, 60(10):1327–1339, 2012. (Cited on pages 19, 77, and 84.)

[72] K. Muelling, J. Kober, O. Kroemer, and J. Peters. Learning to select and generalize striking movements in robot table tennis. *International Journal of Robotics Research*, 32(3):263–279, 2013. (Cited on pages 19 and 84.)

[73] E. Gribovskaya, S. M. Khansari-Zadeh, and A. Billard. Learning nonlinear multivariate dynamics of motion in robotic manipulators. *International Journal of Robotics Research*, 30(1):80–117, 2011. (Cited on pages 19 and 20.)

[74] S. M. Khansari-Zadeh and A. Billard. Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Transactions on Robotics*, 27(5):943–957, 2011. (Cited on pages 19, 20, and 77.)

[75] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98, 1986. (Cited on pages 20, 75, and 76.)

[76] H. Hoffmann, P. Pastor, D. H. Park, and S. Schaal. Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 2587–2592, 2009. (Cited on page 20.)

[77] A. Gams, B. Nemec, L. Zlajpah, M. Wachter, A. J. Ijspeert, T. Asfour, and A. Ude. Modulation of motor primitives using force feedback: Interaction with the environment and bimanual tasks. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5629–5635, 2013. (Cited on page 20.)

[78] S. M. Khansari-Zadeh and A. Billard. A dynamical system approach to realtime obstacle avoidance. *Autonomous Robots*, 32(4):433–454, 2012. (Cited on pages 20 and 76.)

[79] M. Saveriano and D. Lee. Point cloud based dynamical system modulation for reactive avoidance of convex and concave obstacles. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5380–5387, 2013. (Cited on page 20.)

[80] S. V. Raković and D. Q. Mayne. Robust model predictive control for obstacle avoidance: discrete time case. In *Assessment and Future Directions of Nonlinear Model Predictive Control*, pages 617–627. Springer, 2007. (Cited on page 20.)

[81] Z. Shiller, S. Sharma, I. Stern, and A. Stern. Online obstacle avoidance at high speeds. *International Journal of Robotics Research*, 32(9-10):1030–1047, 2013. (Cited on page 20.)

[82] F. Pecora, M. Cirillo, and D. N. Dimitrov. On mission-dependent coordination of multiple vehicles under spatial and temporal constraints. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5262–5269, 2012. (Cited on pages 20 and 87.)

[83] S. J. Anderson, S. B. Karumanchi, and K. Iagnemma. Constraint-based planning and control for safe, semi-autonomous operation of vehicles. In *Proc. of the IEEE Intelligent Vehicles Symposium*, pages 383–388, 2012. (Cited on pages 20 and 87.)

[84] R. Suárez, M. A. Roa, and J. Cornellà. Grasp quality measures. Technical report, Institute of Industrial and Control Engineering, Technical University of Catalonia, 2006. (Cited on pages 21 and 47.)

[85] A. Bicchi. On the closure properties of robotic grasping. *International Journal of Robotics Research*, 14(4):319–334, 1995. (Cited on page 22.)

[86] F. Reuleaux. *The Kinematics of Machinery*. Macmillan, translated by A. B. W. Kennedy, London, 1876. (Cited on page 22.)

[87] B. Mishra, J. T. Schwartz, and M. Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica*, 2:541–558, 1987. (Cited on page 22.)

[88] Z. Li and S. Sastry. Task-oriented optimal grasping by multifingered robot hands. *IEEE Journal of Robotics and Automation*, 4(1):32–44, 1988. (Cited on page 22.)

[89] D. G. Kirkpatrick, B. Mishra, and C. K. Yap. Quantitative steinitz's theorems with applications to multifingered grasping. In *Proc. of the twenty-second annual ACM symposium on Theory of computing*, pages 341–351, 1990. (Cited on pages 23 and 28.)

[90] M. Strandberg. A grasp evaluation procedure based on disturbance forces. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1699–1704, 2002. (Cited on page 23.)

[91] C. Rosales, L. Ros, J. M. Porta, and R. Suárez. Synthesizing grasp configurations with specified contact regions. *International Journal of Robotics Research*, 30(4):431–443, 2011. (Cited on page 23.)

[92] M. A. Roa, K. Hertkorn, C. Borst, and G. Hirzinger. Reachable independent contact regions for precision grasps. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 5337–5343, 2011. (Cited on pages 23, 48, and 54.)

[93] X. Markenscoff, L. Ni, and C. H. Papadimitriou. The geometry of grasping. *International Journal of Robotics Research*, 9(1):61–74, 1990. (Cited on page 26.)

[94] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, 1996. (Cited on pages 36 and 42.)

[95] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001. (Cited on page 37.)

[96] G. I. Boutselis, C. P. Bechlioulis, M. V. Liarokapis, and K. J. Kyriakopoulos. An integrated approach towards robust grasping with tactile sensing. In *Proc. of the IEEE International Conference on Robotics and Automation (to appear)*, 2014. (Cited on page 48.)

[97] J. Aleotti and S. Caselli. Interactive teaching of task-oriented robot grasps. *Robotics and Autonomous Systems*, 58(5):539–550, 2010. (Cited on page 48.)

[98] R. Krug and K. Charusta. icrcpp. `https://github.com/rtkg/icrcpp`, 2014. [Accessed: 2014-04-02]. (Cited on page 49.)

[99] A. Kasper, Z. Xue, and R. Dillmann. The KIT object models database: An object model database for object recognition, localization and manipulation in service robotics. *International Journal of Robotics Research*, 31(8):927–934, 2012. (Cited on pages 49 and 57.)

[100] Inc. Gurobi Optimization. Gurobi optimizer reference manual. `http://www.gurobi.com`, 2014. [Accessed: 2014-04-07]. (Cited on page 49.)

[101] J. Babič, J. G. Hale, and E. Oztop. Human sensorimotor learning for humanoid robot skill synthesis. *Adaptive Behavior*, 19(4):250–263, 2011. (Cited on page 54.)

[102] J. Bimbo, S. Rodriguez-Jimenez, H. Liu, X. Song, N. Burrus, L. D. Senerivatne, M. Abderrahim, and K. Althoefer. Object pose estimation and tracking by fusing visual and tactile information. In *Proc. of the IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 65–70, 2012. (Cited on page 55.)

[103] M. A. Roa and R. Suárez. Influence of contact types and uncertainties in the computation of independent contact regions. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 3317–3323, 2011. (Cited on pages 57 and 98.)

[104] A. Bicchi. Hands for dexterous manipulation and robust grasping: a difficult road toward simplicity. *IEEE Transactions on Robotics and Automation*, 16(6):652–662, 2000. (Cited on page 59.)

[105] J. Dejeu, M. Bechelany, P. Rougeot, L. Philippe, and M. Gauthier. Adhesion control for micro- and nanomanipulation. *ACS Nano*, 5(6):4648–4657, 2011. (Cited on pages 60 and 61.)

[106] E. Rombokas, P. Brook, J.R. Smith, and Y. Matsuoka. Biologically inspired grasp planning using only orthogonal approach angles. In *Proc. of the IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 1656–1661, 2012. (Cited on page 63.)

[107] R. G. Mihalyi, K. Pathak, N. Vaskevicius, and A. Birk. Uncertainty estimation of AR-marker poses for graph-SLAM optimization in 3D object model generation with RGBD data. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1807–1813, 2013. (Cited on page 66.)

[108] D. R. Canelhas, T. Stoyanov, and A. J. Lilienthal. SDF tracker: A parallel algorithm for on-line pose estimation and scene reconstruction from depth images. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3671–3676, 2013. (Cited on page 67.)

[109] N. Vaskevicius, K. Pathak, A. Ichim, and A. Birk. The Jacobs Robotics approach to object recognition and localization in the context of the ICRA'11 solutions in perception challenge. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 3475–3481, 2012. (Cited on page 67.)

[110] I. A. Sucan and S. Chitta. "Moveit!". `http://moveit.ros.org/`, 2013. [Accessed: 2013-09-11]. (Cited on page 68.)

[111] T. Erez and E. Todorov. Trajectory optimization for domains with contacts using inverse dynamics. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4914–4919, 2012. (Cited on pages 73 and 99.)

[112] I. Mordatch, Z. Popović, and E. Todorov. Contact-invariant optimization for hand manipulation. In *Proc. of the ACM SIGGRAPH/Eurographics symposium on computer animation*, pages 137–144, 2012. (Cited on pages 73 and 99.)

[113] O. Brock and O. Khatib. Elastic strips: A framework for motion generation in human environments. *International Journal of Robotics Research*, 21(12):1031–1052, 2002. (Cited on page 75.)

[114] J. H. Hwang, R. C. Arkin, and D. S. Kwon. Mobile robots at your fingertip: Bezier curve on-line trajectory generation for supervisory control. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1444–1449, 2003. (Cited on page 75.)

[115] J. Aleotti and S. Caselli. Robust trajectory learning and approximation for robot programming by demonstration. *Robotics and Autonomous Systems*, 54(5):409–413, 2006. (Cited on page 75.)

[116] K. Schittkowski. *Numerical Data Fitting in Dynamical Systems*. Kluwer Academic Publishers, 2002. (Cited on page 75.)

[117] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Robot programming by demonstration. In B. Siciliano and O. Khatib, editors, *Handbook of Robotics*, pages 1371–1394. Springer, 2008. (Cited on page 76.)

[118] T. Flash and N. Hogan. The coordination of arm movements: An experimentally confirmed mathematical model. *Journal of neuroscience*, 5:1688–1703, 1985. (Cited on page 76.)

[119] R. Weitschat, S. Haddadin, F. Huber, and A. Albu-Schäffer. Dynamic optimality in real-time: A learning framework for near-optimal robot motions. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5636–5643, 2013. (Cited on page 76.)

[120] T. Feix, R. Pawlik, H. Schmiedmayer, J. Romero, and D. Kragic. A comprehensive grasp taxonomy. In *RSS: Workshop on Understanding the Human Hand for Advancing Robotic Manipulation*, 2009. (Cited on pages 76 and 88.)

[121] A. Bernardino, M. Henriques, N. Hendrich, and J. Zhang. Precision grasp synergies for dexterous robotic hands. In *Proc. of the IEEE International Conference on Robotics and Biomimetics*, pages 62–67, 2013. (Cited on page 76.)

[122] A. J. Ijspeert, J. Nakanishi, and S. Schaal. Learning attractor landscapes for learning motor primitives. In *Advances in Neural Information Processing Systems*, pages 1523–1530. MIT Press, 2003. (Cited on pages 76, 77, 81, 84, and 90.)

[123] H. Khalil. *Nonlinear Systems*. Prentice Hall, 2002. (Cited on page 78.)

[124] E. Rückert and A. d'Avella. Learned parametrized dynamic movement primitives with shared synergies for controlling robotic and musculoskeletal systems. *Frontiers in computational neuroscience*, 7, 2013. (Cited on page 82.)

[125] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000. (Cited on page 86.)

[126] B. Houska, H. J. Ferreau, and M. Diehl. ACADO Toolkit – an open source framework for automatic control and dynamic optimization. *Optimal Control Applications and Methods*, 32(3):298–312, 2011. (Cited on page 90.)

[127] Y. Tassa, T. Erez, and E. Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4906–4913, 2012. (Cited on page 99.)

[128] R. Platt, A. H. Fagg, and R. A. Grupen. Null-space grasp control: theory and experiments. *IEEE Transactions on Robotics*, 26(2):282–295, 2010. (Cited on page 99.)

Publications *in the series*
Örebro Studies in Technology

1. Bergsten, Pontus (2001) *Observers and Controllers for Takagi – Sugeno Fuzzy Systems*. Doctoral Dissertation.
2. Iliev, Boyko (2002) *Minimum-time Sliding Mode Control of Robot Manipulators*. Licentiate Thesis.
3. Spännar, Jan (2002) *Grey box modelling for temperature estimation*. Licentiate Thesis.
4. Persson, Martin (2002) *A simulation environment for visual servoing*. Licentiate Thesis.
5. Boustedt, Katarina (2002) *Flip Chip for High Volume and Low Cost – Materials and Production Technology*. Licentiate Thesis.
6. Biel, Lena (2002) *Modeling of Perceptual Systems – A Sensor Fusion Model with Active Perception*. Licentiate Thesis.
7. Otterskog, Magnus (2002) *Produktionstest av mobiltelefonantenner i mod-växlande kammare*. Licentiate Thesis.
8. Tolt, Gustav (2003) *Fuzzy-Similarity-Based Low-level Image Processing*. Licentiate Thesis.
9. Loutfi, Amy (2003) *Communicating Perceptions: Grounding Symbols to Artificial Olfactory Signals*. Licentiate Thesis.
10. Iliev, Boyko (2004) *Minimum-time Sliding Mode Control of Robot Manipulators*. Doctoral Dissertation.
11. Pettersson, Ola (2004) *Model-Free Execution Monitoring in Behavior-Based Mobile Robotics*. Doctoral Dissertation.
12. Överstam, Henrik (2004) *The Interdependence of Plastic Behaviour and Final Properties of Steel Wire, Analysed by the Finite Element Metod*. Doctoral Dissertation.
13. Jennergren, Lars (2004) *Flexible Assembly of Ready-to-eat Meals*. Licentiate Thesis.
14. Jun, Li (2004) *Towards Online Learning of Reactive Behaviors in Mobile Robotics*. Licentiate Thesis.
15. Lindquist, Malin (2004) *Electronic Tongue for Water Quality Assessment*. Licentiate Thesis.
16. Wasik, Zbigniew (2005) *A Behavior-Based Control System for Mobile Manipulation*. Doctoral Dissertation.

17. Berntsson, Tomas (2005) *Replacement of Lead Baths with Environment Friendly Alternative Heat Treatment Processes in Steel Wire Production*. Licentiate Thesis.

18. Tolt, Gustav (2005) *Fuzzy Similarity-based Image Processing*. Doctoral Dissertation.

19. Munkevik, Per (2005) *"Artificial sensory evaluation – appearance-based analysis of ready meals"*. Licentiate Thesis.

20. Buschka, Pär (2005) *An Investigation of Hybrid Maps for Mobile Robots*. Doctoral Dissertation.

21. Loutfi, Amy (2006) *Odour Recognition using Electronic Noses in Robotic and Intelligent Systems*. Doctoral Dissertation.

22. Gillström, Peter (2006) *Alternatives to Pickling; Preparation of Carbon and Low Alloyed Steel Wire Rod*. Doctoral Dissertation.

23. Li, Jun (2006) *Learning Reactive Behaviors with Constructive Neural Networks in Mobile Robotics*. Doctoral Dissertation.

24. Otterskog, Magnus (2006) *Propagation Environment Modeling Using Scattered Field Chamber.* Doctoral Dissertation.

25. Lindquist, Malin (2007) *Electronic Tongue for Water Quality Assessment*. Doctoral Dissertation.

26. Cielniak, Grzegorz (2007) *People Tracking by Mobile Robots using Thermal and Colour Vision*. Doctoral Dissertation.

27. Boustedt, Katarina (2007) *Flip Chip for High Frequency Applications – Materials Aspects*. Doctoral Dissertation.

28. Soron, Mikael (2007) *Robot System for Flexible 3D Friction Stir Welding*. Doctoral Dissertation.

29. Larsson, Sören (2008) *An industrial robot as carrier of a laser profile scanner. – Motion control, data capturing and path planning.* Doctoral Dissertation.

30. Persson, Martin (2008) *Semantic Mapping Using Virtual Sensors and Fusion of Aerial Images with Sensor Data from a Ground Vehicle*. Doctoral Dissertation.

31. Andreasson, Henrik (2008) *Local Visual Feature based Localisation and Mapping by Mobile Robots*. Doctoral Dissertation.

32. Bouguerra, Abdelbaki (2008) *Robust Execution of Robot Task-Plans: A Knowledge-based Approach*. Doctoral Dissertation.

33. Lundh, Robert (2009) *Robots that Help Each Other: Self-Configuration of Distributed Robot Systems*. Doctoral Dissertation.

34. Skoglund, Alexander (2009) *Programming by Demonstration of Robot Manipulators*. Doctoral Dissertation.

35. Ranjbar, Parivash (2009) *Sensing the Environment: Development of Monitoring Aids for Persons with Profound Deafness or Deafblindness*. Doctoral Dissertation.

36. Magnusson, Martin (2009) *The Three-Dimensional Normal-Distributions Transform – an Efficient Representation for Registration, Surface Analysis, and Loop Detection*. Doctoral Dissertation.

37. Rahayem, Mohamed (2010) *Segmentation and fitting for Geometric Reverse Engineering. Processing data captured by a laser profile scanner mounted on an industrial robot*. Doctoral Dissertation.

38. Karlsson, Alexander (2010) *Evaluating Credal Set Theory as a Belief Framework in High-Level Information Fusion for Automated Decision-Making*. Doctoral Dissertation.

39. LeBlanc, Kevin (2010) *Cooperative Anchoring – Sharing Information About Objects in Multi-Robot Systems*. Doctoral Dissertation.

40. Johansson, Fredrik (2010) *Evaluating the Performance of TEWA Systems*. Doctoral Dissertation.

41. Trincavelli, Marco (2010) *Gas Discrimination for Mobile Robots*. Doctoral Dissertation.

42. Cirillo, Marcello (2010) *Planning in Inhabited Environments: Human-Aware Task Planning and Activity Recognition*. Doctoral Dissertation.

43. Nilsson, Maria (2010) *Capturing Semi-Automated Decision Making: The Methodology of CASADEMA*. Doctoral Dissertation.

44. Dahlbom, Anders (2011) *Petri nets for Situation Recognition*. Doctoral Dissertation.

45. Ahmed, Muhammad Rehan (2011) *Compliance Control of Robot Manipulator for Safe Physical Human Robot Interaction*. Doctoral Dissertation.

46. Riveiro, Maria (2011) *Visual Analytics for Maritime Anomaly Detection*. Doctoral Dissertation.

47. Rashid, Md. Jayedur (2011) *Extending a Networked Robot System to Include Humans, Tiny Devices, and Everyday Objects*. Doctoral Dissertation.

48. Zain-ul-Abdin (2011) *Programming of Coarse-Grained Reconfigurable Architectures*. Doctoral Dissertation.

49. Wang, Yan (2011) *A Domain-Specific Language for Protocol Stack Implementation in Embedded Systems*. Doctoral Dissertation.

50. Brax, Christoffer (2011) *Anomaly Detection in the Surveillance Domain*. Doctoral Dissertation.

51. Larsson, Johan (2011) *Unmanned Operation of Load-Haul-Dump Vehicles in Mining Environments*. Doctoral Dissertation.

52. Lidström, Kristoffer (2012) *Situation-Aware Vehicles: Supporting the Next Generation of Cooperative Traffic Systems*. Doctoral Dissertation.

53. Johansson, Daniel (2012) *Convergence in Mixed Reality-Virtuality Environments. Facilitating Natural User Behavior.* Doctoral Dissertation.

54. Stoyanov, Todor Dimitrov (2012) *Reliable Autonomous Navigation in Semi-Structured Environments using the Three-Dimensional Normal Distributions Transform (3D-NDT).* Doctoral Dissertation.

55. Daoutis, Marios (2013) *Knowledge Based Perceptual Anchoring: Grounding percepts to concepts in cognitive robots*. Doctoral Dissertation.

56. Kristoffersson, Annica (2013) *Measuring the Quality of Interaction in Mobile Robotic Telepresence Systems using Presence, Spatial Formations and Sociometry.* Doctoral Dissertation.

57. Memedi, Mevludin (2014) *Mobile systems for monitoring Parkinson's disease*. Doctoral Dissertation.

58. König, Rikard (2014) *Enhancing Genetic Programming for Predictive Modeling*. Doctoral Dissertation.

59. Erlandsson, Tina (2014) *A Combat Survivability Model for Evaluating Air Mission Routes in Future Decision Support Systems*. Doctoral Dissertation.

60. Helldin, Tove (2014) *Transparency for Future Semi-Automated Systems. Effects of transparency on operator performance, workload and trust.* Doctoral Dissertation.

61. Krug, Robert (2014) *Optimization-based Robot Grasp Synthesis and Motion Control*. Doctoral Dissertation.