

**Low-Dimensional Robotic Grasping:
Eigengrasp Subspaces
and Optimized Underactuation**

Matei Th. Ciocarlie

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2010

©2010

Matei Th. Ciocarlie

All Rights Reserved

ABSTRACT

Low-Dimensional Robotic Grasping: Eigengrasp Subspaces and Optimized Underactuation

Matei Th. Ciocarlie

This thesis introduces new methods for enabling the effective use of highly dexterous robotic hands, interfacing with the upcoming generation of neurally controlled hand prostheses, and designing a new class of simple yet effective grasping devices based on underactuation and mechanical adaptation. These methods share a common goal: reducing the complexity that has traditionally been associated, at both computational and mechanical levels, with robotic grasping in unstructured environments.

A key prerequisite for robot operation in human settings is versatility, which, in terms of autonomous grasping, translates into the ability to reliably acquire and interact with a wide range of objects. In an attempt to match the abilities of the most versatile end-effector known, the human hand, many anthropomorphic robotic models have been proposed, with the number of degrees of freedom starting to approach that of their human counterpart. However, these models have proven difficult to use in practice, as the high dimensionality of the posture space means that finding adequate grasps for a target object is often an intractable problem.

In this thesis, we propose using low-dimensional posture subspaces for dexterous or anthropomorphic hands. Human user studies have shown that most of the variance in hand posture for a wide range of grasping tasks is contained in relatively few dimensions. We extend these results to a range of robotic designs, and introduce

the concept of *eigengrasps* as the bases of a low-dimensional, linear hand posture subspace. We then show that a grasp synthesis algorithm that optimizes hand posture in eigengrasp space is both computationally efficient and likely to yield stable grasps.

The emerging field of neurally controlled hand prosthetics faces a similar challenge when using dexterous hand models: bridging the gap between incomplete or noisy neural recordings and the complete set of variables needed to execute a grasping task. We propose using an automated grasp planning component as an interface, accepting real-time operator input and using it to assist in the synthesis of stable grasps. Computational rates needed for direct interaction can be achieved by combining operation in eigengrasp space with on-line operator input. Furthermore, the eigengrasp planning space can also act as an interaction space, allowing the operator to provide meaningful input for the hand posture using few channels of communication.

Algorithmic approaches to low-dimensional grasping can enable computationally effective algorithms and interaction models. Hardware implementations have the potential to reduce the mechanical complexity and construction costs of a hand design, using concepts such as underactuation and passive mechanical adaptation. Instead of complex run-time algorithms, hand models in this class use design-time analysis to improve performance over a spectrum of tasks. Along these directions, we present a set of analysis and optimization tools for the design of low-dimensional, underactuated hands. We focus on tendon-based mechanisms featuring adaptive joints and compliant fingertips, and show how a number of design parameters, such as tendon routes or joint stiffnesses, can be optimized to enable a wide range of stable grasps.

The ability to effect change on the environment through object acquisition (grasping) and manipulation has the potential to enable many robotic applications with high social impact, including effective neural prostheses, robots for house care or personal assistance, *etc.* We believe that the methods presented in this thesis represent a number of steps in this direction, advancing towards a proven solution for reliable autonomous grasping in human environments.

Table of Contents

1	Introduction	1
1.1	Problem Statement	2
1.2	Our Approach	5
1.3	Applications and Impact	6
1.4	Thesis Contributions	9
1.5	Organization	9
2	Background and Related Work	11
2.1	Core Problems in Robotic Grasping	12
2.1.1	The Mechanics of Grasping	12
2.1.2	Contact Models	14
2.1.3	Grasp Analysis and Quality Metrics	18
2.1.4	Grasp Planning	20
2.2	Learning from the Human Hand	22
2.2.1	Grasp Taxonomies	22
2.2.2	Tendon Actuation	24
2.2.3	Hand Neuroprosthetics	26
2.3	Compliant and Adaptive Robotic Hands	28
2.4	GraspIt! - a Simulator for Robotic Grasping	30

3	Eigengrasps	32
3.1	Posture Subspaces for the Human Hand	33
3.2	Application for Robotic Hand Models	34
3.3	Effective Degrees of Freedom	37
4	Low-Dimensional Grasp Synthesis	39
4.1	Quality Function Formulation	41
4.2	Optimization Algorithm	42
4.3	Optimization Results and Discussion	46
5	Interactive Grasp Planning for Hand Neuroprosthetics	51
5.1	Grasp Planning in a Shared Control Framework	52
5.2	System Overview	54
5.3	Interactive Grasp Planner Implementation	56
5.3.1	Hand Position Parameterization	57
5.3.2	Quality Function from Scaled Contact Wrench Spaces	58
5.3.3	Computation of Form-closure Grasps	61
5.3.4	The Complete Planning Pipeline	63
5.4	Interactive Grasping Examples and Analysis	63
5.5	User Input with Variable Confidence	67
5.6	Grasp Planning Experiments with Hand Posture Input	70
5.6.1	Recorded Input from Non-human Primate	72
5.6.2	On-line Input from Human Operator	75
5.7	Discussion and Limitations	76
6	Analysis and Optimization for Underactuated Adaptive Hands	79
6.1	Eigengrasps and Compliant Underactuation	80
6.2	Underactuated Grasping as a Constrained Optimization Problem	81
6.2.1	Contact Constraints	82
6.2.2	Actuation Constraints	83

6.2.3	Quasistatic Grasp Equilibrium Formulation	85
6.3	Grasp Analysis for Underactuated Hands	86
6.3.1	Algorithm Design	87
6.3.2	Example Application and Results	89
6.4	Hand Design Optimization	93
6.4.1	Numerical Optimization	95
6.4.2	Global Optimization: a Case Study	97
6.4.3	Application: Construction of an Optimized Gripper	101
6.5	Towards the Construction of an Eigenhand	105
7	Contact Models for Compliant Fingertips	108
7.1	The Soft Finger Model and its Linearized Version	109
7.2	Computation using Finite Element Analysis	111
7.3	Computation using Analytical Surface Approximations	113
7.4	Applications	117
8	Conclusions	120
8.1	Thesis Summary	121
8.2	Lessons Learned	124
8.3	Directions for Future Research	126
8.4	Robotic Grasping: The Road Ahead	128
A	Biased Neighbor Generating Functions for Simulated Annealing	130
B	Derivation of A Linear Tendon Route Model	132
C	Maximum Frictional Torque as a Function of Contact Load	134
	Bibliography	136

List of Figures

1.1	Computer models of robotic hands spanning the spectrum from a simple gripper to anthropomorphic designs.	3
1.2	One view of the grasping pipeline and applications of the eigengrasp concept.	8
2.1	The Coulomb model friction cone and its linearized version.	15
2.2	Two views of a linearized friction ellipsoid with height 0.2.	15
4.1	Examples of desired contact locations for posture optimization. Left: complete set of pre-defined desired contact locations for the DLR, Robonaut and Human hands. Right: for a desired contact with index i , we define the surface normal $\hat{\mathbf{n}}_i$ and the current distance to the target object \mathbf{o}_i	41
4.2	Simulated annealing example over 70,000 iterations. Each image shows the best state found until iteration k	44
4.3	Evolution of quality function (top) and eigengrasp amplitudes (bottom) of the current search state during simulated annealing	45
4.4	The best value of the quality function at different moments during the optimization algorithm, depending on the dimensionality of the optimization subspace. All the tests were performed using the human hand and averaged over a set of 5 executions for each of 7 test objects.	46

4.5	Best hand postures found in a 2-dimensional eigengrasp space using simulated annealing optimization. Rightmost column also shows final grasps obtained by closing each finger until motion is stopped by contact with the object.	48
5.1	Interactive grasp planning using hand position input from a human operator. Top: system overview; Bottom: applied examples using a real Barrett hand (left) and a dexterous hand in a simulated environment (right).	55
5.2	Search directions defined for the Barrett and human hand models. The direction of the vector \mathbf{l} is predefined relative to the palm. Its magnitude, as well as the values of the angles θ and ϕ are variables defining a conical search area.	58
5.3	Contact wrench space example using a linearized Coulomb friction cone.	59
5.4	Multiple contact wrench spaces, each scaled based on the contact distance metric Δ_i	60
5.5	Example of a complete grasping task: initial approach, finger-preshaping using grasp planning result, continued approach and final grasp execution.	62
5.6	Examples of interactive grasping tasks; each image shows the grasp found for a different approach direction or target object. In all cases the object was successfully grasped and lifted off the table.	65
5.7	Examples of interactive grasping tasks executed in simulated environments. Bottom rows also show the user providing the approach direction via a magnetic tracker. All the presented grasps have form-closure.	66
5.8	Probability density functions for simulated annealing neighbor generation. Comparison between the original formulation (solid blue line) and neighbor generation function biased towards a target jump value of 0.5 (dashed red line).	69

5.9	Interactive grasp planning using hand posture and position input, but no direct control from operator.	72
5.10	Experimental setup and object set used for recording primate grasps.	73
5.11	Examples of form-closure grasps returned by the planner for noisy target postures specified as input. Left: recorded monkey grasps used as reference poses. Middle: grasps planned using reference pose as input with high confidence level ($\sigma = 0.95$). Right: grasps planned without using reference pose.	74
5.12	Left: difference between planned grasps and input grasps. Solid line shows the difference in the amplitude of the first eigengrasp, while dashed line shows the difference in wrist orientation. Right: average time spent to find each form-closure grasp. A value of -0.2 was used as a starting point for the Input Confidence axes to represent the case where planning was carried out without any kind of input.	75
5.13	Examples of interactive grasp planning using input provided by a human operator using a Cyberglove and a magnetic tracker.	76
6.1	Illustration of tendon routing points, marked with red spheres, as the tendon follows a revolute joint marked by a wire frame cylinder.	84
6.2	Set of five objects (glass, flask, mug, phone and toy airplane) used in our tests. The glass also shows a number of possible grasps generated by aligning the hand with its bounding box. For each possible grasp, the approach direction (shown by the arrows) was parallel to one axis of the bounding box. The rotation of the hand around the approach direction (not shown here) was set by aligning the hand with the other axes of the box.	89

6.3	Examples of unsuccessful (top row) and successful (bottom row) candidate grasps. For both cases, the left image shows the starting position used as input to the grasp analysis algorithm and the right image shows the outcome of the grasp as computed through simulation with a time-stepping dynamic engine.	91
6.4	The effect of hand design parameters on the likelihood of obtaining a stable grasp. For each combination of joint torque and stiffness ratio, the color indicates the number of stable grasps obtained from the optimization pool; a darker color means a higher number of stable grasps. For each object, the results were normalized to a scale of 0 to 1 by dividing by the maximum number of grasps found for that object.	95
6.5	Left: a two-fingered gripper with single tendon actuation used as a case study for the optimization framework. Right: detailed description for a joint of the proposed design.	97
6.6	Left: gripper model for the <i>GraspIt!</i> environment. Right: examples of grasps from the optimization pool.	99
6.7	Comparison of unbalanced joint forces as a measure of grasp stability between optimized and ad-hoc configuration	102
6.8	Prototype gripper constructed according to optimization results. Notice the different tendon route and rubber joint dimensions between the two distal joints.	103
6.9	Two grasps executed with the prototype gripper. Top row: centered starting position and symmetrical grasp. Bottom row: off-centered starting position requiring passive adaptation to an irregular shape.	104
6.10	Examples of grasps successfully executed using the prototype gripper.	104
7.1	View of the contact area for normal forces of 5N (left), 20N (middle) and 25N (right) from within a transparent cube.	112

7.2	Analytical surfaces approximating the local geometry of a grasped object. Magnifications show local approximations for the thumb and little finger.	115
7.3	Simulation of a manipulation task using an anthropomorphic robotic hand. Top row: object is picked up and rotated above the palm. Bottom row: by controlling fingertip force, rotational sliding is allowed at the contacts until object rests on palm. By closing the fingers a stable grasp is obtained.	118
A.1	Comparison of probability density functions for different annealing temperatures T_k . Solid blue line shows PDF of neighbor generation function without external input; dashed red line shows PDF of neighbor generation function when a jump with target u_t and confidence σ is specified.	131
B.1	Torque computation for tendon entry point	132

List of Tables

3.1	Eigengrasps defined for the robotic hand models used in this thesis.	36
4.1	Number of form-closed grasps obtained from 20 pre-grasps found in a 2D eigengrasp space (average and standard deviation over 5 executions for each hand and object).	49
6.1	Quasistatic analysis for grasp planning.	92
6.2	Results of gripper design optimization	102

Acknowledgments

Well, here we are. It's been a long journey. And, like all long journeys, many times you envision getting to your destination, you anticipate how it's going to feel, how somehow that moment will equal in magnitude all the emotions of the long road brought together. And then it happens so fast, you get swept into a new life, you barely notice that wow - the finish line has come and gone. You can think to yourself - wait a minute, that's not fair! All the effort, all the long hours, were supposed to build up to a grand finale. But that moment turned out to be nothing but another milestone, a blur, on the way to where? That's how it turned out for me at least, but it's all ok. I didn't need a magic moment at the end because I was lucky: I got to enjoy my years in grad school while they were actually happening.

If there is one thing I wish could tell every starting grad student, it is this: your PhD is not about your thesis. Sure, the thesis has its importance, and you better get one done if you ever want to graduate, but it's not the main point. The PhD is all about what you accumulate, what you learn, who you become. This is certainly true professionally speaking; for me it was also true from a personal standpoint. During these years, the special people that I have been in contact with have shaped who I am now. Getting the degree is more or less a byproduct of that experience. (That said, maybe you can read the rather long document that follows too; I have put a lot of work into it...)

The persons who are most entitled to thanks and gratitude should find them not in a scholarly document, but with every single interaction, be that in person, over the phone or in an email. There is nothing so special about this thesis that it can

serve as a vehicle for what I should be saying every day. However, knowing the right thing to do and actually doing it are unfortunately different, so I find myself trying to condense a lifetime of love, affection and guidance in a few paragraphs. I long hesitated to even attempt this, but the PhD is supposed to also teach one how to capture and express the essential with few words, so I guess I'll give it a shot.

First and foremost, I would like to thank my wife and partner, Gabriela, and my parents, Pixi and Alin. They are the persons in the world that my sentiments always seem to echo, and whose happiness reflects in mine. My parents are always a part of me, be they in the room down the hall, two blocks down the street, or two continents away. Anything that I might achieve in life, I owe, directly or indirectly, to what they are and what they've taught me. Gabriela is the spark that came into my life and set us on a new and exciting journey. She motivates me to set high goals, and then inspires and supports me to achieve them. Any moment feels incomplete if she is not sharing it with me, and I look forward to wherever the road will take us next. If the years we spent at Columbia are any indicator to go by, we can achieve anything together.

Looking back on the time when I headed out to start grad school, I realize now how little I knew about the life I was embarking on - it was more of a leap into the unknown than anything else. Luckily for me, I found a teacher, a mentor and a friend all in one person, my advisor, Prof. Peter Allen. A self-described "abrasive New-Yorker who won't take no for an answer", Peter is actually much more than that. True, he is always honest and direct, allowing you to know where you stand with him, and more importantly, where you stand, period. But he is also a true friend, always has the best interests of his students at heart, and is a person that I always knew I could count on when I needed advice and support. From an academic standpoint, I always admired, and benefited from, his unsurpassed knowledge of our (very broad) field and his infallible sense of its new directions and developments. I have greatly enjoyed working with Peter in grad school, look forward to meeting him

at every robotics conference around the world, and hope to continue to be his friend, colleague and mentee for many years to come.

During the first years of my PhD, I was lucky to overlap in the Robotics Lab with Andy Miller, who not only handed off his (grad) life's work to me, but also showed me the nuts and bolts of robotic grasping research and taught me how to write my very first paper. Andy is the rare person who is, at the same time, an excellent researcher, a natural born software engineer, and completely unassuming about any of it. Often during my PhD I had the impression that I was merely enlarging the path that he had already blazed. I thank him for his guidance, as well as the quality of his work, that served as a foundation for my own thesis.

It has been a pleasure to share grad life with my lab mates, fellow researchers in the exciting field of robotics. I would like to thank all of them for all the advice, collaborations, suggestions, small talk, robotics beer events and general good times. Paul Blaer enthusiastically welcomed me to the lab (“hmpf, grumble, I suppose I don’t see any show-stoppers so far”), and I never found a problem he didn’t have a practical solution for. Atanas Georgiev was a pleasure to converse with on any subject, makes a mean yogurt, and designed the world’s longest serial port adapter. Too bad we often shared lab time for only one hour in the morning (when I got in) and one hour in the evening (when he got in). During the last part of my PhD, I enjoyed working together with Corey Goldfeder, who has one of the most creative minds I have seen in our field, always bubbling up with new ideas; our work put together was more than the sum of its parts. Hao Dang was a one-man compute cluster, I would like to thank him for all the hard work building the grasp database. Some of the my best memories from the PhD come from scanning French cathedrals with Paul and Alejandro Troccoli, both great travel companions. Finally, I am glad I shared the lab with Austin Reiter during my last year at Columbia. I hope that, other than making me feel like I had perspective on things, our lunches together at the Interchurch Cafeteria also provided at least some useful advice for him. Many

thanks also go to Austin for helping me jump the final hurdle and deposit this thesis.

During my first week at Columbia, Prof. Shree Nayar welcomed me in his usual warm and friendly manner. I would like to thank him for that, but also for many other things, starting from an enthralling computer vision class during my first semester and all the way to my defense where he served as committee chair. Prof. Eitan Grinspun taught me the secrets of physical-based simulation, and gave me confidence when Finite Element Analysis was getting the better of me. I would like to thank Profs. Robert Howe and Dinesh Pai for their detailed comments and valuable suggestions. I was privileged to have them as members of my thesis committee. I am also grateful to Prof. Chad Jenkins, who showed me the ropes of event organization, taught me how to get involved with the robotics community at large and hosted me for my very first guest seminary.

I would like to thank the entire Visualization Group at IBM Research for two great summers there. If I had to design my own manager, mentor and colleagues from scratch, I would probably end up with as close approximations as I could manage of Dr. Tom Jackman, Dr. Jim Klosowski and Dr. Wagner Correa.

I would like to thank the staff and members of the Computer Science Department, my academic home. Paula Ryan, Remi Moss, Lily Secora, Elias Tesfaye, Patricia Hervey, Cindy Walters, Anne Fleming, Evelyn Guzman: we students often don't realize how lucky we are to have such competent and dedicated people working together for our benefit. I enjoyed planning and organizing student trips together with Remi and Lily, and thank them for their enthusiasm. I am also deeply grateful to Mary Van Starrex, who helped Gabriela and I get over some of the toughest times during our PhDs. Her warm and kind nature made the department a better place.

Last, but most certainly not least, I would like to thank the friends that Gabriela and I shared the grad school years, as well as those in Romania, who, on our vacations back home, made us feel like we had never left. By nature, grad school is a revolving door. No sooner have you made new friends than they graduate and leave; you are

happy for them moving on but sad to see them go. The optimist in me though thinks that we have new friends for life, and can't wait to meet them again, wherever in the world that might be. I would like to thank John and Alisa, Hila and Suhit, Arezu, Rafi, Janak, Saba, Nicola, Sameer and Prerana, Andrea, Stefano, Andrew, Benko and everybody else for the great times they've given us. Cheers!

As I was saying, here we are, and now it is really done. It's been a fantastic couple of years, but it's time to move on. Let's see what's out there!

Matei Ciocarlie
Menlo Park, CA
February 3rd, 2010

... I seem to have been only like a boy playing on the seashore, and diverting myself in now and then finding a smoother pebble or a prettier shell than ordinary, whilst the great ocean of truth lay all undiscovered before me.

–Isaac Newton

(doctoral research, on a good day)

When all you've got is a hammer, everything looks like a nail.

–Anonymous

(doctoral research, every other day)

Chapter 1

Introduction

What distinguishes a robot from a computing device? Arguably, it is not its “thinking” ability. For example, some of today’s most “intelligent” computing devices (apparently so much so that the word “smart” has even made its way into their name), are smartphones. In a portable embodiment, they have amazing computing power, multiple sensors (such as cameras, microphones, GPS, etc.) and myriad abilities: recognize faces or songs, get directions, answer questions by drawing on vast amounts of on-line data, and the list can continue. However, they are never considered to belong to the robot family. On the other hand, a seemingly “unintelligent” mechanical device on a factory assembly line, capable of doing one thing and one thing only (maybe placing a label onto a smartphone casing), has long ago earned its stripes as a “robot”. It would seem that the crucial characteristic of a robot is **the ability to interact with, and directly effect change on its environment**. This interaction can take multiple forms, such as self-locomotion, opening doors, calling an elevator, and many more. In this thesis, we are interested in a different type of interaction: the ability to acquire objects from the environment, a prerequisite for object transport, tool use, etc. Welcome to the realm of robotic grasping.

The contrast between today’s ubiquitous computing devices and relatively old factory robots also leads to a corollary question: why are there no omnipresent robots

in today's society? We can already identify applications that are seemingly waiting for the right technology to develop. For example, robots for house care and personal assistance is an area that promises to have high social impact, as the percentage of the population that would benefit from robotic personal assistance is constantly increasing [Census]. However, autonomous operation in real-life human settings is a challenging goal, involving tasks ranging from localization and locomotion to grasping and manipulation to safe human-robot interaction. Robust grasping ability is currently one of the weaker links in this chain. While computation power has made it out of the factory or research lab and into every day situations, robotic grasping and manipulation have not.

In order to make this leap, we must address two key differences between factory or lab-type settings and every day human environments. The first one is versatility: a factory robot is usually designed and programmed for grasping few (or even a single) object types. In contrast, many real-world applications require interaction with a wide range of objects. The second one regards the ability to perceive the environment. While a factory floor is highly structured and its configuration is well-known, a robot in the real world must use its sensors to operate in new and potentially unforeseen conditions. The vision of ubiquitous robotic assistants will not be realized without the ability to perform grasps of common objects *in a robust and repeatable manner*, subject to realistic levels of sensing error and noise. Addressing this problem has the potential to enable a wide range of new robotics applications, for which other major components are already in place.

1.1 Problem Statement

One of the traditional approaches to robotic grasping in human environments is to use extensive control mechanisms for highly complex hands equipped with many joints and actuators. It is easy to understand this direction: the most versatile end-effector

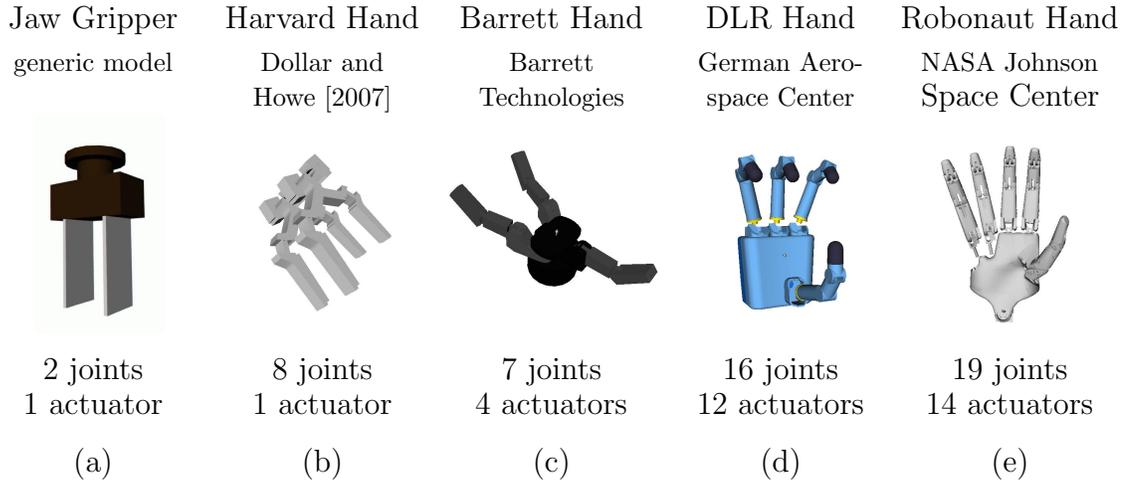


Figure 1.1: Computer models of robotic hands spanning the spectrum from a simple gripper to anthropomorphic designs.

known is the human hand, highly complex, but also unrivaled in its manipulation abilities. In an attempt to match these abilities, many anthropomorphic designs have been proposed over the past decades, some of them resulting in exceptionally well engineered, very promising robot hands. These include the work of Jacobsen et al. [1984], Butterfass et al. [1998] (shown in Figure 1.1d), Lovchik and Diftler [1999] (shown in Figure 1.1e) and Vande Weghe et al. [2004], to quote only a few. However, these robotic hands have proven difficult to operate autonomously. As the number of degrees of freedom starts to approach the case of the human hand, effective algorithms that can handle high-dimensional configuration spaces are required in order to take advantage of these designs.

If the problem of complex and high-dimensional control algorithms can be solved, highly dexterous hands promise, in the long term, to give robots true dexterity, allowing them to not only grasp, but also manipulate and use objects. However, they also have less appealing characteristics: they are typically expensive to build, and fragile in use. This combination of practical shortcomings can have negative implications in the short and medium terms: it elevates the barrier of entry for doing research in this area, makes the sharing and dissemination of results more difficult and prevents the

technology from permeating into the mainstream.

An alternative approach has been to use very simple end-effectors, such as parallel jaw grippers (Figure 1.1a), which have a number of advantages over their dexterous counterparts. They are much easier to use and require less effort in designing and running control algorithms. Just as important, they are cheap to build, enabling many research groups to use them in practice and also facilitating the exchange of results. However, they have intrinsically limited applicability, lacking the versatility needed for complex applications.

Until recent years, surprisingly few designs have been proposed in the spectrum of possibilities between these two extremes. One of the most common examples is the Barrett hand (Barrett Technologies, Cambridge, MA), based on a design originally introduced by Ulrich et al. [1988] (Figure 1.1c). Probably not by coincidence, it is also one of the most commercially successful robot hands, widely used in the research community. A recently revived effort of designing hands that belong in this category has seen a number of research prototypes emerge (an example is shown in Figure 1.1b); we will review these later in this thesis. Based on the key principles of underactuation and passive mechanical adaptation, these hands have produced very promising results even in the presence of sensing errors, while still imposing a relatively small production cost. However, the theoretical foundations of this line of work are still being laid out, and relevant design tools are still being extended and refined.

We can formulate our problem statement as follows. *Current robotic hands research often involves a compromise. Dexterous and/or anthropomorphic models are so complex that they have proven difficult to use even for simple tasks, while simpler, intuitive designs are limited in their ability. Few alternatives exist in between these two classes, as they are inherently difficult to design: too simple to draw direct inspiration from the human hand, but too complex for straightforward, intuitive concepts. As a result, the community is currently without a proven solution for reliably performing grasping tasks in unstructured environments.*

1.2 Our Approach

If we wish to reproduce human-like grasping it would seem natural to draw inspiration not only from the hardware of the human hand, but also from the software; that is, the way the hand is controlled by the brain. This may initially sound like an overly lofty goal: a large part of the human cortex is dedicated to grasping and manipulation, and it would seem reasonable to assume that all of this cognitive machinery is dedicated to finely controlling individual joints and generating highly flexible hand postures. However, results in both robotics and neuroscience research that we will review in this thesis point to the contrary, suggesting that a majority of the human hand control during common grasping tasks lacks individuation in finger movements.

In our work, we use low-dimensional hand posture subspaces to express coordination patterns between multiple degrees of freedom for robotic hands. In particular, we consider linear subspaces defined by a number of basis vectors that we refer to as *eigengrasps*. Each eigengrasp is a vector in the high-dimensional hand posture space; we use linear combinations of a relatively small number of these vectors to obtain a wide range of hand postures for grasping tasks.

A key aspect when using this approach is the trade-off between its computational advantages and the implied reduction in the range of directly accessible hand postures. An eigengrasp subspace is only useful in as much as it contains enough variance in hand posture to allow for successful completion of the grasping task. In our work, we start from the results of Santello et al. [1998], who applied dimensionality reduction methods on a large set of human grasping postures obtained from user studies. Their results show that a 2-dimensional subspace contains more than 80% of the variance in hand posture. The analysis of human digit coordination patterns during grasping is in general a very active area of research; in the following chapters we will also provide an overview of current results and discuss their implications for our approach to robotic grasping.

Our main interest in this thesis is the application of low-dimensional posture

subspaces for robot hands. There are two main approaches that rely on this concept. The first one is *exploratory*, aiming to derive optimal posture subspaces. In this thesis we rely on exploratory results from the field of kinesiology, obtained through human user studies. We will also present our results for analytical optimization of posture subspaces for robot hands, limited for now to non-anthropomorphic designs. The second approach, which is explored in depth in this thesis, is *constructive* in nature: given a particular set of eigengrasps, we aim to construct algorithms that take advantage of operating in a low-dimensional domain. Then, we show the applicability of these algorithms to different scenarios involving robotic grasping.

Eigengrasp-based algorithms represent a way of reducing the computational complexity associated with dexterous robot hands. We also investigate a related approach, constructive in the most literal sense of the word: providing a physical embodiment to the concept of low-dimensional hands using underactuation and passively adaptive mechanisms. Such designs can fill the gap between complex anthropomorphic hands and simple intuitive grippers, providing much of the versatility of the former, with the simplicity and reliability usually associated with the latter. However, in order to fully realize this potential, we must have efficient tools to analyze the execution of grasping tasks when using this class of devices; a significant part of the work described in this thesis is dedicated to analysis and optimization tools for underactuated compliant hands.

1.3 Applications and Impact

One of the key features of the low-dimensional eigengrasp framework we introduce is the ability to simplify the search for stable grasp postures when using a hand with a large number of degrees of freedom. We present an eigengrasp planning algorithm that can be used to find form closure grasps using dexterous hands that have traditionally been very difficult to plan for. The core of this algorithm is an optimization

procedure that operates along 2 eigengrasp directions; even when using such a reduced dimensionality space, we show that the planner is successful in deriving stable multi-fingered grasps for a large variety of target objects.

There are two main applications for a computationally efficient grasp planning algorithm for dexterous hands. The first one is *direct*: such an algorithm can be used to plan grasps that are then executed on real objects. The efficient nature of our algorithm makes it suitable for implementation in a field where computational efficiency is of paramount importance: hand neuroprosthetics. This direction implies the combination of user input and automatic grasp planning for controlling an artificial hand; we will discuss some of the aspects of this interaction in detail later in this thesis. We will also present an implementation of our eigengrasp backed planning method that acts as an interface between a human operator and an artificial hand. Our system can accept on-line operator input, adapt to changes in the input and enable the successful execution of grasping tasks.

It is important to note that direct application of our grasp planning algorithms requires extensive sensing capabilities in order to provide information on the grasped object. This is in contrast with an *indirect* application of low-dimensional planning: the same algorithm can be used strictly inside a simulated environment, where perfect sensing capabilities can be assumed. The goal of using an efficient grasp planning algorithm in simulation is to generate very large amounts of labeled grasp information for dexterous hands. This data can then be used as input for data-driven and learning-based grasping strategies, which are in turn used in the real world. The eigengrasp-based algorithms presented in this thesis are currently being used in a separate line of work as part of a complete pipeline for data-driven grasping. Our focus here is on the grasp planning algorithms used as the data generation component, which we will present in detail in the following chapters. For an in-depth treatment of the learning and other components of the pipeline, we refer the reader to the studies by Goldfeder et al. [2009a,b].

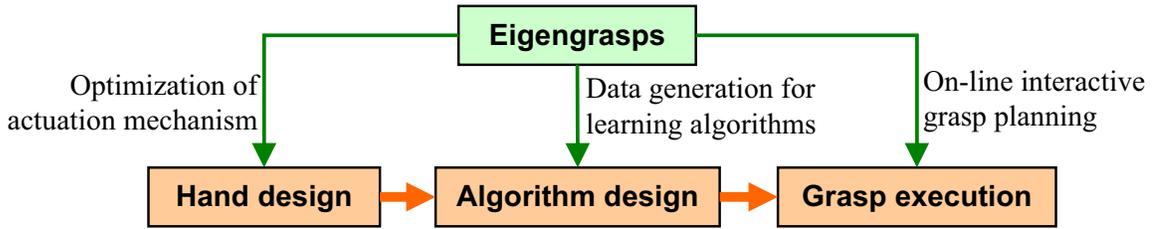


Figure 1.2: One view of the grasping pipeline and applications of the eigengrasp concept.

So far, the proposed applications mainly target the family of dexterous hands, with many degrees of freedom. In this thesis, we also present a number of analysis and optimization tools dedicated to low-dimensional, underactuated and passively compliant hands. Implementations range from compliant joints and adaptive kinematics to soft and compliant fingertips. The main application of our analysis methods is the design of robotic hands that are reliable in a wide range of grasping scenarios, yet are inexpensive to build and do not require extensive sensing or computation effort at run time. Using these tools, we can again put computational resources to work under an off-line premise, at hand design time rather than grasp execution time. As applied examples, we will show how we can compute hand design parameters, such as relative joint stiffnesses or tendon routing point placements, that increase the range of stable grasps that a given hand can execute. As complementary approaches for reducing the complexity of robotic grasping, we ultimately hope to combine eigengrasp-based algorithms with adaptive underactuation, leading to a new generation of artificial hands.

Overall, we present applications of low-dimensional hand posture subspaces spanning multiple stages of the robotic grasping pipeline, as synthesized in Figure 1.2.

1.4 Thesis Contributions

The main contributions of this thesis are:

- introduction of the *eigengrasp* concept as a basis for a linear posture subspace of a robotic hand;
- a computationally efficient grasp synthesis algorithm for dexterous robotic hands operating in eigengrasp space;
- an online, interactive grasp planning system where an automated planner interacts with live input from a human operator, as in the case of hand neuroprosthetics;
- a set of analysis and optimization tools for underactuated, passively adaptive and compliant hands, aiming towards hardware implementation of the eigen-grasp concept;
- an efficient method for constructing contact models for compliant fingertips, taking into account the extended range of forces and moments that such contacts can sustain.

1.5 Organization

Having discussed the motivation behind our work as well as its general context, we now describe the remaining contents of this thesis. In Chapter 2, we review some of the key principles in the field of robotic grasping which will be used throughout the thesis. We also review the most relevant previous work in the field, with a particular focus on three main areas: core problems in robotic grasping, the study of the human hand with its lessons for robot hands, and the design of robotic hands, especially underactuated and adaptive models.

Chapter 3 lays the cornerstone of our work, by introducing the eigengrasp concept and discussing its implications for robotic hands. Chapter 4 then presents the eigengrasp-based grasp synthesis algorithm which is the backbone of the applications presented in the next chapters. This algorithm uses a simulated environment as a computational platform for finding stable grasps for a wide range of hand designs, including a model of the human hand.

In Chapter 5 we discuss grasp planning for hand neuroprosthetics. We show how the previously introduced low-dimensional grasp planner can be used interactively, using on-line input from an operator. We show how the operator can provide part of the information needed for a stable grasp, and also guide the automated planner as it is supplying the data that is missing from the input. We show results for both human and primate operators, using appropriate hand kinematics and input sources for each case. This chapter also takes the eigengrasp planner out of the simulated environment and into the real world, as we demonstrate how its results can be applied to grasp objects with a real robotic hand.

Chapter 6 presents our analysis and optimization tools for underactuated and passively adaptive robotic hands. We focus on the optimization of a compliant tendon-driven mechanism for improving the force generation capabilities of an underactuated hand. After introducing the general analysis formulation, we show two applied examples, one using a numerical optimization approach and the other targeting a more constrained design problem that can be cast as a solvable global optimization. Then, in Chapter 7, we introduce our method for efficient construction of soft finger contact models, which is used by many of the algorithms presented in this thesis.

Finally, in Chapter 8, we review the main contributions of the thesis, and also identify what we believe to be the most promising directions for continuing the research and extending the results presented here.

Chapter 2

Background and Related Work

At its core, **robotic grasping** is the problem of immobilizing an object inside an end-effector in the presence of external disturbances. It generally enables the acquisition and transport of objects, in which case the external disturbances that must be resisted typically include gravity and inertial forces.

Two important problems that we do not address in this thesis are those of intended **object use** (other than transport) and **robotic manipulation**, both of which require additional components of the system far exceeding our scope. For example, the ability to use an object requires high-level semantic understanding of both the grasped object and the task being performed. These aspects are active research areas in their own right. Robotic manipulation usually refers to the ability of changing the pose of the object inside the end-effector without breaking the grasp (hence it is also referred to as in-hand manipulation). We believe that reliable grasping performance, while a useful ability in itself, is also a prerequisite for both of these tasks.

In the rest of this thesis we will use the term *robotic hand* with the general meaning of an end-effector that is primarily used for grasping. This includes the full spectrum that we have mentioned previously, from dexterous and anthropomorphic models (to which the term *hand* is more intuitively applied) to simpler designs such as grippers.

2.1 Core Problems in Robotic Grasping

The human hand is composed of a palm and multiple fingers; so are most robotic hands. In more general terms, the palm can be thought of as a base, while each finger is a kinematic chain. In this thesis, we focus mainly on serial robots; in this sense, a robotic hand becomes a collection of separate kinematic chains, and many of the tools originally developed for the study of industrial robotic arms apply to hands as well. What distinguishes a hand from its larger cousins on the factory floor is that it is purposefully designed to establish multiple contacts with a target objects, using not just a single tool-tip but rather many of its links. The nature of these contacts is paramount for grasping, as we will illustrate in this section.

2.1.1 The Mechanics of Grasping

Perhaps the most fundamental aspect of using kinematic chains in general, and robot fingers in particular, is that of forward kinematics. It is used to express the location and orientation of all the links in a chain as a function of the chain joint values. Most robotics textbooks (e.g. Tsai [1999]) cover this concept in detail, so we will not dwell on it here. We define the **hand posture** as the set of locations and orientations of all the links of all the fingers in the hand, relative to the palm.

We also define the set of **internal degrees of freedom**, or internal DOFs, of the hand, as the set of variables required to completely define the hand posture at any instant in time. We note that this is not always the same as the set of joints that comprise the hand: if two or more joints are rigidly coupled, and their values are always in a fixed relationship to each other, then they share a single DOF. We will refer to the set of hand DOFs using the vector $\boldsymbol{\delta} = [\delta_1, \delta_2, \dots, \delta_d] \in \mathcal{R}^d$ where d is the total number of DOFs of the hand. Forward kinematics is the function that allows us to compute hand posture for a given value of $\boldsymbol{\delta}$.

In order for the hand's state to be fully determined, we must specify not only its

posture, but also the location and orientation of the palm. We define **hand position** as the vector $\mathbf{p} = [t_x \ t_y \ t_z \ r_x \ r_y \ r_z] \in \mathcal{R}^6$ containing the 6 variables that encode this information (3 for translation and 3 for rotation), relative to the object that is the target of the grasp. For a given target object of known location and geometry, we define a **grasp** as a combination of hand posture and position. Therefore, a grasp is completely specified by the vector $[\mathbf{p} \ \boldsymbol{\delta}] \in \mathcal{R}^{d+6}$.

A grasp implicitly generates a set of contacts between the hand and the grasped object. The exact locations of these contacts will depend on the hand posture and position, but also on the geometry of the object itself, which, in the most general case, is arbitrary and impossible to parameterize. As a result, there is no general analytical formulation which can express the set of contacts between the hand and the object as a function of the variables that define the grasp.

We note that, as a simpler but more limited formulation, it is also possible to define a grasp by specifying just the set of contacts between the hand and the object, rather than the state of the hand itself. In some cases, the values of $\boldsymbol{\delta}$ and \mathbf{p} can then be retrieved from the set of the contacts by using inverse kinematics. We will review a number of approaches that use this formulation later in this chapter.

An vital tool for analyzing the behavior of the contact set is the **contact Jacobian**, which relates infinitesimal joint motion to motion at the contact locations. Equally important, it can be shown through a relatively simple derivation equating input and output virtual powers that the transpose of the contact Jacobian relates forces and torques applied at the points of contact to joint forces and torques [Mason and Salisbury, 1985]. For the rest of the thesis, we will use the convenient notation of a **wrench** to express the combination of a force and a torque; in three-dimensional space, a wrench is a six-dimensional vector. For an in-depth analysis of wrenches, as well as their correspondent in velocity domain, screws, we refer the reader to the textbooks by Tsai [1999] and Hunt [1978].

One transition that is left to explore is that from contact wrenches to resultant

object wrench. This is usually encapsulated in the **grasp map matrix**. The exact notation varies slightly among different studies. For example, Mason and Salisbury [1985] use the grasp matrix to relate contact velocities to object velocity, in which case its transpose relates object wrenches to contact wrenches. In this thesis, we will use the notation of Prattichizzo and Trinkle [2008], also used by Han et al. [2000], where the grasp map relates contact wrenches to the resultant object wrench.

We now have a complete set of tools for going from the joints of the hand to the state of the grasped object. However, the simplified framework that we have built so far silently assumes that a contact can apply any required wrench to the object, based on the joint forces and torques. Real-life contacts place important constraints on the wrenches that can be transmitted through the interface. We discuss these constraints in the next section.

2.1.2 Contact Models

A **contact model** is a description of the forces and torques that can be transmitted between two bodies through a contact. It usually expresses two constraints: (a) bodies can only push each other through the contact, not pull; (b) any wrench applied must be supported by contact friction. The space of legal wrenches defined by the contact model is also referred to as the **Contact Wrench Space**, or CWS.

In order to simplify the expression of contact constraints, it is common to use a **contact reference frame** where the axes are aligned with relevant directions for the contact. Throughout this thesis we will use a contact reference frame where the z axis is aligned with the contact normal, and its direction is pointing inside the body that is being considered. In this frame, any tangential force can be decomposed in two components, along the x and y axes, while the normal force is applied strictly in the z direction.

One of the most commonly used contact models is that of Coulomb Friction, which applies to rigid bodies creating a point contact. This model assumes that a contact

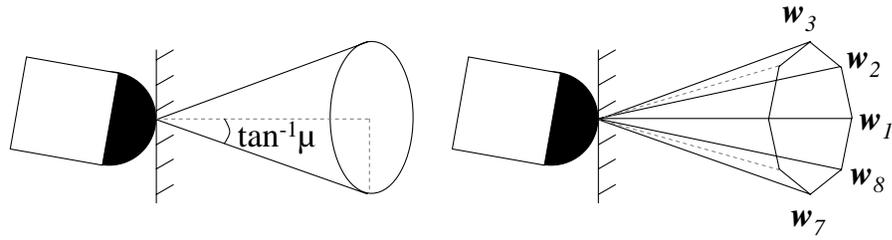


Figure 2.1: The Coulomb model friction cone and its linearized version.

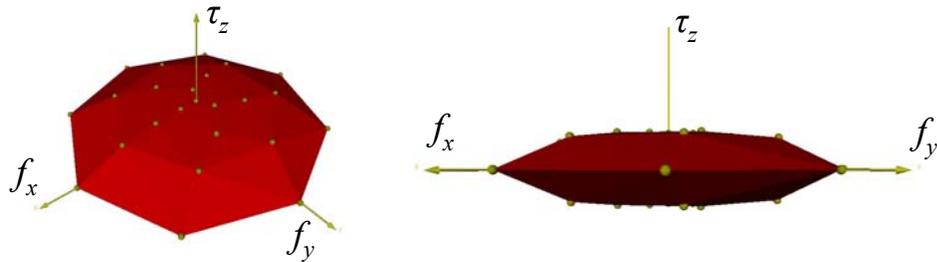


Figure 2.2: Two views of a linearized friction ellipsoid with height 0.2.

can apply any desired level of normal force, f_z . In addition, it can also apply some level of tangential frictional force. The magnitude of the tangential force is restricted by the magnitude of normal force as well as the contact friction coefficient μ . Overall, the total force applied at the contact must lie within a cone aligned with the contact normal, commonly known as the *friction cone*. The Coulomb model also specifies that no torques can be applied through the contact.

A significant drawback for using this model in practice is the quadratic nature of the constraints. A common approximation, used for example by Miller and Allen [1999], is to linearize the friction cone into a *friction pyramid* by defining a number of samples \mathbf{w}_i on its boundary. This process is illustrated in Figure 2.1. In this case, total contact force is expressed as a weighted linear combination of the sample wrenches. This model is commonly referred to as **Point Contact with Friction**, or PCwF.

The Coulomb friction model describes the behavior of the contact as long as it

is confined to a single point. However, if the two bodies are locally similar, or if one of them is deformable, contact occurs over some area. As a result, it is also possible to apply a frictional moment of magnitude τ_z about the contact normal. In order to express the constraint relating the magnitudes of frictional force and moment, we use the limit surface model introduced by Goyal et al. [1991]. Howe and Cutkosky [1996] further extended this work, and showed that, in the general case, we can use the following approximation: for a given magnitude of the normal force, the relationship between tangential friction and frictional torque describes a three-dimensional ellipsoid. The height of this ellipsoid depends on the pressure distribution inside the contact, and can be determined experimentally. This approach is usually referred to as a **Soft Finger** model, or SF.

We note that it is possible to linearize the soft finger constraint in a manner similar to that applied in the case of the friction cone of the Coulomb model. The result is a linear description of the friction constraint; an example is presented in Figure 2.2.

The main feature of the SF model is that it enables modeling frictional torque. However, it relies on an accurate value for the height of the friction ellipsoid, and ignores many other characteristics of the contact. Many other models have been proposed for soft finger contacts, covering a wide range in both complexity and computational efficiency. A significant part of these approaches target the human finger, in the context of biomedical or haptic applications, and focus on the deformation undertaken by the finger during contact, rather than the space of wrenches that the contact can sustain.

One of the most accurate ways of simulating complex and irregular geometry or layered structures in contact is Finite Element Analysis (FEA). While this method has been used for the analysis of soft fingertips [Dandekar et al., 2003], high computational requirements make it difficult to use for interactive dynamic simulations. Better computational performance can be achieved by using analytical models instead. Xydas and Kao [1998] proposed the power-law model, and used FEA as well as experimen-

tal data to derive its parameters for specific materials and fingertip shapes. Barbagli et al. [2004] compared four analytical models of the human fingertip, with different rotational friction properties. Pauly et al. [2004] used a point-cloud representation for efficient simulation of quasi-rigid objects in contact; for any vertex of an object that was deemed to be part of the contact surface, the Boussinesq analytical model was used to compute contact traction and the local displacement that it leads to.

Fast analytical models are of particular interest in the field of haptics, where contact forces computed during the simulation are fed back to a human user, creating the illusion of interaction with virtual objects. Tada and Pai [2008] model the human finger as a thin shell enclosing a fluid-like material, and incorporate subject-specific geometry in the simulation. An example of haptic force rendering using a soft finger model is also presented by Frisoli et al. [2006], using an extension of the traditional god-object haptic rendering method introduced by Zilles and Salisbury [1995].

Finally, modeling soft contacts is also of interest in robotic control applications. Specific pad materials and their impact on grasping and manipulation tasks were analyzed by Shimoga and Goldenberg [1992] as well as Chang and Cutkosky [1995]. Doulgeri et al. [2000] used sensory feedback to design a controller that can be used with a robotic finger when the dynamic properties of the soft fingertip are initially unknown. This method was extended by Han et al. [2001] and applied for two-fingered grasps and manipulation tasks. Doulgeri and Fasoulas [2003] also discuss the control of rolling manipulation using deformable fingertips.

The linearized versions of the Point Contact with Friction and Soft Finger models are the main contact models used throughout this thesis when studying grasps. In general, we assume that the grasped object is rigid. In the case where the links of the robotic hand are assumed to be made of metal or plastic, we will use the PCwF model. For robotic fingertips covered with a compliant material, such as a layer of rubber, we will start from the SF model. Later, in Chapter 6, we will also show how we have augmented the SF model for computationally efficient simulations.

2.1.3 Grasp Analysis and Quality Metrics

Having introduced contact constraint models, we can continue to the analysis of a grasp as a whole. In general, grasp analysis algorithms aim to:

1. study the wrenches that a grasp can apply to an object via a set of contacts under the constraints that:
2. each contact wrench must be inside its respective CWS (i.e. friction rules must be obeyed), and
3. the hand must be able to apply the desired contact wrenches through a legal combination of joint forces and torques.

A criterion that is commonly used to characterize a grasp is that of **form-closure**. In this thesis, we use a definition following Mason and Salisbury [1985]: a grasp is said to have form-closure if and only if it can resist an infinitesimal disturbance regardless of its direction¹. Assuming infinite motor power and material resistance, this property also means that the grasp can resist any disturbance, regardless of magnitude, by simply scaling up the applied joint forces. We will informally refer to grasps that satisfy the form-closure criterion as **stable grasps**. We note that, if we disregard constraints 2 and 3 above, form-closure is trivial to check: the necessary and sufficient condition is that the grasp map matrix is full row rank [Prattichizzo and Trinkle, 2008].

Taking into account constraint 2 in the list above, the total space of legal wrenches that a grasp can apply to the object is often referred to as the **Grasp Wrench Space**, or GWS. We note that the GWS and the space of disturbances that a grasp can resist are dual concepts, as in order to resist an external wrench, a grasp must apply an equal

¹ In other works, this property is named force-closure, while form-closure is referred to as the same ability, but achieved using only frictionless contacts. This has regrettably led to some degree of confusion in the community.

wrench in the opposite direction. Ferrari and Canny [1992] presented two methods for building the GWS based on the linearized versions of the CWS introduced earlier. The first one builds a “norm 1” GWS as the convex hull of all individual CWS. The second approach is to build an “infinity norm” GWS as the convex hull of the Minkowski sum of the contact wrenches that make up each CWS.

The Minkowski sum operation allows us to consider the independent contributions of contacts on different fingers, but it quickly becomes intractable for large numbers of contacts. In this thesis we therefore use the norm 1 GWS. We note that it is also possible to build fast approximations of the infinity norm GWS, particularly when only a portion of the space is of interest [Borst et al., 1999]. For more details about the practical implementation of the GWS construction algorithm we also refer the reader to the work of Miller and Allen [1999].

Ferrari and Canny [1992] show that a grasp has form-closure if and only if the wrench space origin is contained in the hull that defines the GWS. They also introduce a pair of numerical **quality measures**. The ϵ metric is equal to the radius of the largest six-dimensional ball, centered at the origin, that can be enclosed inside the hull. The volume metric is simply equal to the volume of the GWS. We note that the ϵ metric can be considered a measure of the worst-case performance of the grasp, while the volume metric measures its overall performance. In this thesis we will use the ϵ metric as our primary grasp quality measure.

If we know in advance what task the robot will perform while the object is in its grasp, then we might also know the space of disturbances that we expect to encounter, also known as the Task Wrench Space, or TWS. In this case, it is natural to also measure the ability of the grasp to resist the expected disturbances. Li and Sastry [1988] used a six-dimensional ellipsoid to model the TWS and measured how well it fits inside the GWS. Borst et al. [2004] proposed a different method of building the TWS, based on a distribution of wrenches acting on the target object.

Overall, the family of methods that rely on the GWS constructed as described here provide us with a powerful and computationally efficient analysis tool, while satisfying condition 2 in our list. However, they are generally concerned only with the distribution of contacts, and ignore the configuration of the hand itself. As a result, condition 3 is not always met.

One possible approach for meeting all three goals that we have listed is to assemble all the constraints as a Linear Programming problem [Kerr and Roth, 1986, Trinkle, 1992]. Naturally, this is only possible when using the linearized versions of the friction constraints. Buss et al. [1996] showed that quadratic friction constraints can be implemented as positive-definiteness constraints, making it unnecessary to use linear approximations. Han et al. [2000] further improved this method, showing that the resulting system can be solved efficiently by formulating it as a Linear Matrix Inequality, rather than a Linear Program.

The above methods can be used not only to verify force satisfiability criteria, but also to check how “easily” these criteria are met, thus providing numerical quality measures for the grasp. This is usually done by turning one of the constraints in the system into an **optimization objective**. For example, Trinkle [1992] derives the legal contact forces that are as far as possible from the boundaries of their respective CWS. Alternatively, it is possible to optimize for the lowest joint effort necessary to resist a given disturbance. These methods are collectively referred to as **grasp force optimization** tools: they serve not only to quantify the best performance of the grasp, but also to derive the joint torques that should be applied in order to realize this level of performance.

2.1.4 Grasp Planning

Grasp planning is one of the main applications for numerical grasp analysis tools. Its general purpose can be stated as follows: given a robotic hand and a target object, the goal is to find one or more grasps that either satisfy or optimize a given grasp

quality criterion or metric.

As in the case of grasp analysis, grasp planning can be formulated as a problem strictly in the space of contacts. Given an object to be grasped, the goal is to find a set of contacts on its surface according to a given quality criterion. Recent examples include the work of Roa and Suarez [2007a] and Liu et al. [2004]. It is also possible to find contact regions, rather than individual points, such that the quality criteria will be met as long as one contact is placed anywhere inside each region [Nguyen, 1986, Ponce et al., 1993, Roa and Suarez, 2007b]. However, contact space approaches rarely guarantee that the contacts are physically satisfiable by a real robotic hand. Rezzoug and Gorce [2003] solve for the inverse kinematics of a simplified hand model using supervised learning, and produce a hand configuration such that the fingertips satisfy a number of given point contacts (if possible). An alternative to the use of inverse kinematics is presented by Platt et al. [2002, 2004], starting with the hand in contact with an object and combining multiple control laws for performing incremental contact adjustments.

A different approach to grasp planning is to search for a hand configuration such that the resulting contacts with the object provide a good grasp, again using a quality metric criterion. In this case, the variables that are being determined are the hand posture and position, described earlier. For dexterous hand models, the high-dimensionality of this search space quickly renders brute-force approaches intractable. One possible alternative attempts to take inspiration from human grasping; we will review this approach in detail in the next section. For comprehensive overviews regarding autonomous grasp synthesis for robotic hands we also refer the reader to the detailed reviews authored by Shimoga [1996] and Bicchi and Kumar [2000].

It is important to note that many of the grasp planning algorithms that have been presented assume that a complete 3D model of the grasped object is available. This approach is valuable for theoretical analysis, off-line computation of good grasps of known objects, or if an extensive set of sensors is available for on-line 3D model

reconstruction. However, when a robot is operating in an unknown environment, the amount of sensory information can be insufficient for constructing complete 3D models of novel objects. Saxena et al. [2008] present a learning approach where logistic regression is used to infer good grasping points for a simple gripper based directly on 2D images, without building an explicit object model. Other methods for operating in unstructured environments include explicitly modeling the uncertainty associated with inaccurate range sensors, as shown by Hsiao et al. [2007], and using tactile sensing to compensate for other sensing errors, as demonstrated by Edsinger and Kemp [2006].

2.2 Learning from the Human Hand

Robot hand researchers have always felt a natural inclination towards using the human hand as a model. Jacobsen et al. [1984] identify two reasons for this. First, the human hand is living proof that such design choices can function exceptionally well in practice. Second, it is hoped that robotics researchers can draw on the great body of biomedical literature that studies the human hand, or even on their own personal experiences using it. The eigengrasp approach that is the core of this thesis can be considered an example of this direction as well. In this section we review some of the work on grasp taxonomies that initiated the low dimensional approach presented here. We also review some of the work on tendon actuation, in the context of both human and robotic hands, that can be thought of as a foundation for tendon actuated low-dimensional hands. Finally, we discuss current results in a field of confluence between human and robotic hands, namely hand neuroprosthetics.

2.2.1 Grasp Taxonomies

As we have mentioned previously, one of the main difficulties in understanding human hand control is the large number of degrees of freedom involved. This flexibility gives

rise to an enormous set of possible hand configurations. One possible explanation for human efficiency in selecting appropriate grasps assumes that humans unconsciously simplify the large search space through learning and experience. Consequently, most human grasps would derive from only a few discrete postures, which can be classified according to various criteria. The attempt to formalize this process has resulted in the concept of **grasp taxonomy**.

Early work by Napier [1956] assumed that most grasps fall into one of two distinct categories: “power” grasps, requiring the ability to resist arbitrary forces of (potentially) high magnitude, and “precision” grasps, requiring small adjustments of posture in order to finely control the direction of forces that the hand is applying to the object. Numerous investigators have since extended Napier’s original taxonomy, including the widely referenced work by Cutkosky [1989] which proposed further subdivisions according to finger shape, such as “prismatic” and “circular” grasps.

Iberall [1997] reviewed a large field of work on grasp taxonomies, from areas such as anthropology, medical research, rehabilitation and robotics. The author noted that “across this diverse set, themes are repeated, suggesting the possibility of a unifying view”. She also introduced the concept of the Virtual Finger, “an abstract representation for a collection of individual fingers and surfaces applying an oppositional force”.

The grasp taxonomy concept generated significant interest in robotic grasping research. For example, Stansfield [1991] used it as the basis for a rule-based system for preshaping the hand for grasping. Miller et al. [2003] also used Cutkosky’s grasp taxonomy concept to define a number of starting positions, or pre-grasps, when searching for good grasps of a given object using a non-anthropomorphic robotic hand. Cipriani et al. [2006] applied this concept for prosthetic hands, which we discuss later in this section, assuming that the human operator can only select from a small set of pre-grasp shapes.

A related body of work attempts to synthesize grasps based on human examples,

without explicitly using grasp taxonomies. We note that this approach is generally limited to using human hand models, or anthropomorphic robotic hands. For example, Aleotti and Caselli [2006] used a Cyberglove to record human grasp trajectories and postures and replicated them on the same target objects using NURBS. Li et al. [2007] used a shape matching approach, sampling an object into a dense cloud of oriented points and matching against a small database of known human hand poses.

2.2.2 Tendon Actuation

Grasp taxonomies provide information about the range of postures needed for grasping, but they tell us little about how to achieve these postures in practice. In the human hand, this is role of the musculo-skeletal system. This complex system is the subject of a large field of ongoing biomedical exploratory research, most of which is beyond the scope of this work. We will briefly review a few constructive approaches, which attempt to model this system using computational tools, as they are also relevant to the effort of designing artificial counterparts.

The only component in the system capable of active force generation is the muscle, usually modeled as described by Zajac [1989]. It is important to note that, in the presence of external forces, the muscle is also capable of additional elongation, responding with a passive, spring-and-damper-like force. Artificial motors typically do not exhibit this behavior, thus another source of compliance is necessary if one desires to reproduce it. The forces produced by the muscles are then transmitted to the joints through the tendon network. Since tendons are usually significantly stiffer than muscles, it is common to model them as inextensible, in which case tendon excursion is considered to equal the change in length of the muscle.

The tendon network itself presents a number of modeling challenges. A single tendon will usually affect multiple joints on the way to its final point of insertion. An et al. [1979] present clinical data for an average human hand, including joint locations, tendon insertion points and moment arms for a neutral hand pose. Tendon

networks also exhibit multiple confluence and bifurcation points, especially in the case of the finger extensor mechanisms. One possible approach, presented by Tsang et al. [2005], is to ignore such tendon interconnections. In contrast, Valero-Cuevas et al. [1998] use a complex rhombus model for the finger extensors. By simulating tendon movement as it slides across the bone surface, as well as constraints such as pulleys and insertion points, Valero-Cuevas and Lipson [2004] are able to analyze the torque generating capacity of such complex networks and compare a number of proposed network topologies. Sueda et al. [2008] also demonstrate a biomechanical approach able to simulate complex musculoskeletal systems, with both tendon routing and sliding constraints, by modeling tendons as strands following a trajectory specified with cubic B-splines. We also note that it is not always necessary to explicitly model the musculo-skeletal system in order to capture its effects on the final grasp. An example is shown by Kry and Pai [2006], who analyze human joint compliance starting from high frequency recordings of joint velocities and fingertip forces, using only a kinematic model of the hand.

A relatively large number of robotic hand designs use tendon actuation, but their level of anthropomorphism varies greatly. The complexity of the human musculo-skeletal system prevents exact artificial replicas. It thus becomes critical to understand the effect that approximations, or even ad-hoc tendon network designs have on grasping abilities. In particular, Fu and Pollard [2006] use a linear programming approach to integrate tendon connections in the grasp constraints discussed in Section 2.1.3 and study their effect on grasp quality measures. Kurtz and Hayward [1991] discuss dexterity measures for tendon actuated hands, while Bicchi and Prattichizzo [2000] and Pollard and Gilbert [2002] optimize tendon arrangements for human-like robotic hands.

One of the most common “shortcuts” employed when designing robotic tendons is to avoid complex network configurations; each joint (or sometimes pair of joints) is controlled by a single tendon, with no bifurcation or confluence points. An example of

this approach is the Utah/MIT hand [Jacobsen et al., 1984], which uses four tendon-actuated fingers. Each finger has four joints and is controlled by six tendons grouped in three antagonistic pairs. Both the kinematic structure of the hand and the tendon routes are inspired by the human hand, but the tendon network has no confluence or bifurcation points. A simpler, non-anthropomorphic design is the Barrett hand. Based on the design introduced by Ulrich et al. [1988], it has three identical fingers and its tendon mechanism allows the distal joint of each finger to operate even if the proximal link is stopped due to contact.

Another example of anthropomorphism in a robotic hand is the Robonaut hand [Lovchik and Diftler, 1999]. Developed to perform repair missions during space walks, human-like kinematics were necessary in order to enable the hand to use tools developed for human astronauts. However, the requirements of operation in harsh environments made long tendon cables impractical; as a result the Robonaut hand uses a combination of flexible drive shafts and short cables for actuation. A more recently developed anthropomorphic hand model is the Shadow hand (The Shadow Robot Company, London, UK). This design uses human-like kinematics, actuated through a combination of antagonistic tendons and passive springs, with no confluence points between tendons. Finally, one of the most human-like robotic hand designs in existence is that of the ACT hand [Vande Weghe et al., 2004]. Intended to model the human hand as closely as possible it includes a very close replica of human tendon networks such as the complex finger extensor mechanism. The kinematic structure and bone geometry are highly anthropomorphic as well.

2.2.3 Hand Neuroprosthetics

An area that is uniquely positioned to gain from insights into both human and robotic grasping is the field of hand prosthetics. Such devices combine a degree of human control with artificial hardware and algorithms. In particular, the discovery of the relationship between the activity of the neurons in the motor cortex and movement of

the upper limb [Georgopoulos et al., 1986] has spurred an effort to use these signals to control computers and robots. Clinical brain-machine interface prosthetics based on this research could provide restoration of function to those with amyotrophic lateral sclerosis, quadriplegia, or other pathologies that cause the loss of motor function.

Consider the goal of building an anthropomorphic prosthetic arm and hand that are controlled by cortical output. A number of complex factors comprise dexterous grasping and manipulation, including positioning the arm, orienting the wrist, and shaping the fingers. Together, a high number variables must be controlled in order to successfully complete a grasping or manipulation task.

Taylor et al. [2002] enabled a primate to directly control the linear velocity of the endpoint of a robot arm through 3 DOFs in real time. This control was achieved by measuring the activity of individual cortical neurons that correspond to individual preferred directions of each neuron in space. More recently, Velliste et al. [2008] demonstrated the additional continuous cortical control of a robotic pincer, while Artemiadis et al. [2007] were able to decode the correlation between cortical activity and finger aperture. However, the nature of dexterous grasping is very different from arm movement or simple pinching. The human hand is a complex manipulator whose function is to conform to the shape of the object to be grasped, then close stably on it in a way that allows the desired type of manipulation. It is not expected that recorded neural data will provide enough information about grasp shape to completely describe a grasp, at least for the foreseeable future.

In an attempt to bridge this gap, many research groups have turned to the grasp taxonomy results that we have already discussed. Cipriani et al. [2006] presented an algorithm that assumes that the human operator can only select from a small set of pre-grasp shapes, relying on the passive mechanical adaptability of the Cyber-Hand design [Carrozza et al., 2006] to complete the grasp. Tsoli and Jenkins [2007] compared different dimensionality reduction techniques applied to human hand motion capture data; their results showed that a human operator can perform simple

grasping tasks by controlling an artificial hand through a 2-dimensional input device like a computer mouse. Zecca et al. [2002] discussed the use of electromyographic (EMG) signals for robotic hand control as an alternative to neural recordings. However, translation of EMG information into joint positions requires the use of complex learning methods [Afshar and Matsuoka, 2004, Bitzer and van der Smagt, 2006], and is also limited to few channels of information.

2.3 Compliant and Adaptive Robotic Hands

The algorithms presented in the previous section have shown promise in achieving reliable grasping performance using sparse control information. A complementary approach, gathering increasing momentum in the research community, is to also adapt to the grasped object at a mechanical, rather than computational level. This direction promises to impact not only the field of neuroprosthetics, where control information is limited, but also fully autonomous operation in unstructured environments, constrained by limited sensing ability. In the previous sections we have already touched on the subject of mechanical adaptation for robotic hands, when discussing the Barrett Hand and the CyberHand. It is time now to formalize this concept and review the related body of work.

This approach towards robotic hand design focuses on two key principles: underactuation and mechanical adaptation. The former is a relatively well-established concept, used for example in both the Robonaut [Lovchik and Diftler, 1998] and DLR [Butterfass et al., 1998] hands. However, it is traditionally implemented via rigid coupling between joints, such as metallic cables or gears. More recent work has highlighted the advantages of combining underactuation with passive compliance, allowing the hand to adapt to the surface of the object at a mechanical rather than computational level.

It is important to clearly specify the difference between underactuation and me-

chanical adaptation, and a definition accepted across the community has yet to emerge. In this thesis, we use the following definition, inspired from the conventions of Birglen et al. [2008]:

- as mentioned before, the degrees of freedom of the hand are the set of variables whose values fully determine the posture of the hand.
- for **fully actuated hands** there is a one-to-one correspondence between joints, degrees of freedom and actuators. The number of actuators, joints, and degrees of freedom are all equal.
- for **non-adaptive underactuated hands**, the number of actuators is smaller than the number of joints, but *equal to the number of degrees of freedom*. Coupled joints are rigidly linked, so the posture of the hand is fully specified by knowing only the positions of the actuators.
- for **adaptive underactuated hands**, the number of actuators *is smaller than the number of degrees of freedom*. The values of some of the joints in the hand will depend on the presence or absence, as well as the shape, of the grasped object. Actuation levels alone can not fully determine the shape of the fingers.

We note that, in practice, underactuation is a necessary, but not sufficient condition for mechanical adaptation. In order to achieve the latter, underactuation must usually be complemented by passive compliance.

There are multiple ways of achieving passive adaptation with a robotic hand design. Perhaps the earliest example is the Soft Gripper introduced by Hirose and Umetani [1978], using tendons for both flexion and extension. Ulrich et al. [1988] pioneered the use of a breakaway transmission mechanism which is now used in the Barrett hand. Dollar and Howe [2007] used tendon actuation for flexion together with compliant, spring-like joints that provided extension forces to design the SDM Hand. Gosselin et al. [1998] used four-bar linkages to construct the MARS hand, which later

evolved into the SARAH family of hands [Laliberte et al., 2002]. Gosselin et al. [2008] also proposed a tendon-driven design for a robotic hand with 15 degrees of freedom and a single actuator. For more details, we also refer the reader to the comprehensive reviews included in the studies of Dollar and Howe [2006] and Birglen et al. [2008].

A key aspect of designing mechanically adaptive hands is that the small number of actuators limits the flexibility in choosing grasping strategies at execution time. In a sense, the traditional grasp planning task is replaced by careful optimization of the design parameters, performed off-line and before the hand is even built, in order to increase the reliability of the grasps. Therefore, efficient tools are needed for the analysis and optimization of this class of devices. For example, Dollar and Howe [2006] have optimized the actuation and compliance forces of a tendon-driven design. Birglen et al. [2008] present a remarkably detailed and encompassing optimization study for underactuated hands, focusing mainly on four-bar linkages but with applications to other transmission mechanisms as well. Generally, optimization of a highly underactuated, and thus deceptively simple hand, is a complex problem; in other words, *simple is hard!*

2.4 GraspIt! - a Simulator for Robotic Grasping

A vital development and testing platform, used throughout the work described in this thesis, is the publicly available *GraspIt!* simulation engine developed in the Robotics Laboratory at Columbia University and originally introduced by Miller and Allen [2004]. Dedicated to the study of robotic grasping, this simulator can accommodate a wide variety of hand and robot designs. Each grasp can be evaluated using the GWS ϵ and volume quality metrics described in Section 2.1.3. Visualization methods allow the user to create arbitrary 3D projections of the 6D grasp wrench space and see the weak point of the grasp. The dynamics engine within *GraspIt!*, introduced by Miller et al. [2003], computes the motions of a group of connected robot elements, such as

an arm and a hand, under the influence of controlled motor forces, joint constraint forces, contact forces and external forces. This allows the dynamic simulation of an entire grasping task, as well as the ability to test custom robot control algorithms.

In our work, we have used *GraspIt!* for three main purposes:

- as a platform for testing and validating algorithms in cases where real-world testing was impossible or impractical. For example, real-world testing of a grasping algorithm using 4 different dexterous hands or thousands of grasped objects exceeds the possibilities of most research groups; a simulated platform makes such tests possible.
- as a computational back-end used on-line while performing real-world grasping tasks. In this cases, the simulator provides a replica, rather than a substitute for the real world.
- as a tool for generating very large amounts of labeled grasp data. This information can then be used for multiple purposes, such as hand design optimization or data-driven grasping algorithms.

All the new tools and algorithms presented in this thesis have in turn been integrated into *GraspIt!*, resulting in a new version of the simulator. In parallel with the work described in this thesis, we have also maintained and updated the codebase. In particular, the new version is available under the more permissive GNU General Public License [GPL]. This effort comprised the creation of a new, GPL-compatible rapid collision detection and contact determination system, as well as numerous other code updates. As of the time of this writing, the latest version of *GraspIt!* is 2.1.0, which includes all the work described here. It can be downloaded, under the GPL license, at <http://www.cs.columbia.edu/~cmatei/graspit>.

Chapter 3

Eigengrasps

In order to introduce the concept of low-dimensional posture subspaces for robotic hands, which forms the core of this thesis, we will start from the study of human grasp taxonomies. Synthesizing a significant body of work that we have reviewed earlier, grasp taxonomies assume that humans simplify the huge space of possible grasps through learning and experience, enabling them to quickly choose good grasps for a wide variety of objects.

Santello et al. [1998] investigated this hypotheses by collecting a large set of data containing grasping poses from subjects that were asked to shape their hands in order to mime grasps for a large set ($n = 57$) of familiar objects. Principal Component Analysis of this data revealed that *the first two principal components account for more than 80% of the variance*, suggesting that a very good characterization of the recorded data can be obtained using a much lower dimensionality approximation of the DOF space. In our work, we refer to the principal components of the dataset of hand configurations described above as **eigengrasps**.

In this chapter we will introduce our eigengrasp-based framework, discuss some of the theoretical implications of using low-dimensional human grasp data for robotic hands and present the robotic hands that we have applied it to. This lays the groundwork for the next chapters, where we will present eigengrasp-based algorithms and their applications.

3.1 Posture Subspaces for the Human Hand

While numerical analysis of human hand postures can reveal the "synergies" in the data, it tells us very little about the *causes* of this intrinsic low-dimensional nature. Two explanations seem natural: the first one assumes that inter-digit coordination is caused by mechanical constraints in the anatomy of the hand. This direction suggests building robotic hands with highly interconnected finger actuation mechanisms. An example is the prototype developed by Brown and Asada [2007], using a low-dimensional control system along directions similar to the ones presented by Santello et al. [1998]. The second explanation assumes that motor control synergies take place at a higher level in the Central Nervous System, as discussed for example by Mason et al. [2001] and Cheung et al. [2005]. This approach implies the use of low-dimensional control algorithms for dexterous robotic hands, such as the ones presented in this thesis. However, the nature of human control synergies is still an open question and an active area of research, and combinations of the two approaches discussed above also seem very likely.

Another important aspect concerns the relationship between eigengrasps and the task being performed. Todorov and Ghahramani [2004] have shown that the execution of different manipulation tasks (such as flipping pages or crumpling paper) is characterized by different sets of principal components. Interestingly, Thakur et al. [2008] have identified a posture subspace even in the less constrained setting of haptic exploration tasks. Mason et al. [2001] and Santello et al. [2002] have also shown that hand posture during the reach phase of a complete reach-to-grasp action is described by a different (and lower-dimensional) principal component spectrum than the grasp phase. These results show that, when using a low-dimensional control space for robotic hands, the choice of the subspace has to be correlated with the proposed task.

Finally, all the studies discussed so far have used principal component analysis, and thus have addressed only linear subspaces that can be extracted from hand posture

data. Linear decomposition has been successfully used in the past on different types of biometric data, ranging from face appearance [Turk and Pentland, 1991] to dynamics of arm motion [Fod et al., 2002]. However, non-linear dimensionality reduction methods can potentially reveal different manifold structures of the same data. Tsoli and Jenkins [2007] compared a number of such methods, including Isomap and Locally Linear Embedding, for extracting 2-dimensional non-linear manifolds from human hand motion data. Their results show that, while low-dimensional manifolds can be obtained using a number of different methods, non-linear approaches can provide better separation between the low-dimensional projections of different task domains and thus simplify the task of low-dimensional teleoperation.

3.2 Application for Robotic Hand Models

When performing human user studies, the usefulness of a hand posture subspace can be quantified by how well it approximates a given set of input data. This exploratory approach is natural in the context of studying the human hand. In this thesis, we are also interested in a constructive approach, oriented towards application for artificial hands: given a hand posture subspace, we will use it to synthesize new hand postures for accomplishing a particular task. We see this effort as complementary to current attempts of understanding and extracting relevant low-dimensional data: if eigengrasp based algorithms can be proven effective, they would only benefit from further optimization of the operation subspace.

As we have mentioned before, this thesis is a study of robotic grasping, rather than object-specific manipulation. We are mainly interested in dexterous grasps that can resist a wide range of disturbances. It therefore seems natural to quantify a posture subspace by its ability to generate stable grasps. To achieve this goal, we will present an algorithm that actively searches an eigengrasp space for appropriate hand postures. We base our approach on published results obtained from human grasping data, which

can also be applied to robotic models using an empirical mapping as described below. While we have found our choices to produce good results for achieving stable grasps of a large variety of objects common in human environments, the optimal choice of eigengrasps for non-human hands, as well as the choice of which eigengrasps to use for different or more specialized tasks, are open questions and interesting directions for future research.

In our work, we have applied the eigengrasp concept to a total of 4 dexterous hand models: the Barrett hand (Barrett Technologies, Cambridge MA), the DLR hand [Butterfass et al., 1998], the Robonaut hand [Lovchik and Diftler, 1998] and finally a 20-DOF model of a human hand. For the human hand we have directly used the eigengrasp subspace obtained by Santello et al. [1998], taking advantage of the fact that it has been derived through rigorous study over a large number of recorded samples. Since such data is not available for robotic hand models, we have derived eigengrasp directions attempting to define grasp subspaces similar to the one obtained using human hand eigengrasps. In most cases, such decisions could be based directly on the similarities with the human hand: for example, the human metacarpophalangeal (MCP) and interphalangeal (IP) joints can be mapped to the proximal and distal joints of robotic fingers. In the case of the Barrett hand, changes in the spread angle DOF were mapped to human finger abduction. All our hand models, as well as the 2 dominant eigengrasps used for each, are shown in Table 3.1.

The eigengrasp concept allows us to design flexible control algorithms that operate identically across all the presented hand models. The key to this approach is that the eigengrasps encapsulate the kinematic characteristics of each hand design. Control algorithms that operate on eigengrasp directions do not need to be customized for low-level operations, such as setting individual DOFs, and can concentrate on the high-level task. The algorithms that we will present in the following chapters treat all hand models from Table 3.1 identically, without the need for any hand-specific tuning or change in parameters.

Model	DOFs	Eigen-grasp	Description	Min	Max
Barrett	4	1	Spread angle opening		
		2	Finger flexion		
DLR	12	1	Prox. joints flexion Finger abduction Thumb flexion		
		2	Dist. joints flexion Prox. joints extension Thumb flexion		
Robonaut	14	1	Thumb flexion MCP flexion Index abduction		
		2	Thumb flexion MCP extension PIP flexion		
Human	20	1	Thumb rotation Thumb flexion MCP flexion Index abduction		
		2	Thumb flexion MCP extension PIP flexion		

Table 3.1: Eigengrasps defined for the robotic hand models used in this thesis.

3.3 Effective Degrees of Freedom

In the applied example of grasp planning, we will study whether the eigengrasp subspace contains the hand postures needed for stable grasps of the target objects. A corollary question is whether results obtained using a small set of eigengrasps would imply that the other DOFs of the hand are useless. Turning again to the human hand for a preliminary answer, Santello et al. [1998] have found that the 2 dominant eigengrasps encapsulate most of the variance in posture over a large set of grasps. However, they have also showed that eigengrasps 3 through 6 (in decreasing order of importance), while accounting for less than 15% of the posture variance, do not represent noise and are related to the object to be grasped. Furthermore, the study was performed in the absence of the real object, as subjects were asked to reproduce grasps from memory. This suggests that, even if we choose to perform the grasp planning stage in a low dimensional space, during the final stages of the grasp the shape of the object will force the hand to deviate from eigengrasp space in order to conform exactly to its surface.

Our eigengrasp framework implies a two-stage approach to the task of automated grasp synthesis: first, hand posture is optimized in a low-dimensional eigengrasp space. The dimensionality reduction makes this process computationally tractable even for complex dexterous hand models. In the second stage, starting from the best hand posture found in eigengrasp space, the hand is closed by simultaneously flexing all the finger joints until contact with the target object stops all motion. This step does not require the control algorithm to perform any more pose refinement at a computational level, but only to issue a binary “close all fingers” command after which the final pose is determined implicitly through contact with the object. We will use this version of the refinement stage in Chapters 4 and 5.

When using a dexterous hand, the refinement stage is relatively straightforward. It takes advantage of the versatility of complex kinematic chains, where multiple DOFs allow the hand to better match the surface of the object. However, it is natural to ask

if this is a case of under-utilizing the abilities of hand: what is the purpose of being able to fine-tune joints individually, if we are letting the surface of the target object determine their values implicitly? One obvious answer is that these abilities can be used for in-hand or fine manipulation tasks, once the grasp has been completed. However, we believe that, if looking strictly for stable grasping ability, we can find more efficient alternatives.

One option is to decrease the number of actuators to more closely match low-dimensional control directions, and perform the second stage, of pose refinement, through mechanical adaptation. In a sense, a fully actuated hand can be considered the ultimate self-adapting hand, albeit at a very high construction cost. In Chapter 6 we will also perform a number of preliminary studies for complementing the eigengrasp framework with an adaptive underactuated mechanism, aiming to obtain most of the implicit pose refinement abilities at a fraction of the cost.

Chapter 4

Low-Dimensional Grasp Synthesis

The eigengrasp based grasp planning algorithm that we present in this chapter is the backbone of the applications presented in the next two chapters. Its main goal is to find hand postures in a given low-dimensional subspace that conform to the shape of a target object. We then analyze how many of these posture directly lead to stable grasps. In a sense, our search algorithm is an analysis tool that can tell us (albeit only from its own point of view) if the subspace is suitable for dexterous grasping tasks.

We approach automated grasp synthesis as an optimization problem, seeking to maximize the value of a high-dimensional quality function Q that characterizes a given combination of hand posture and position:

$$Q = f(\boldsymbol{\delta}, \mathbf{p}), \quad \boldsymbol{\delta} \in \mathcal{R}^d, \quad \mathbf{p} \in \mathcal{R}^6, \quad (4.1)$$

where d is the number of intrinsic hand DOFs, $\boldsymbol{\delta}$ is the complete set of DOFs which define the hand posture (assuming a non-adaptive hand design, as defined in Section 2.3) and \mathbf{p} contains the position and orientation of the palm.

We will first present our implementation of the quality function, then discuss the optimization algorithm that is applied to maximize it over a space of possible hand postures and positions. In general, most quality function formulations are highly

non-linear, with complex constraints as well as gradients that are difficult, or even impossible to compute analytically. These problems are compounded by the high dimensionality of the optimization domain. Consider for example the case of the human hand model, where $d = 20$: this results in a 26-dimensional optimization domain, rendering most optimization algorithms intractable. However, we can choose a basis comprising a eigengrasps, with $a \ll d$, and a hand posture placed in the subspace defined by this basis can be expressed as a function of the amplitudes α_i along each eigengrasp direction:

$$\boldsymbol{\delta} = \boldsymbol{\delta}_m + \sum_{i=1}^a \alpha_i \mathbf{e}_i \quad (4.2)$$

where $\boldsymbol{\delta}_m$ is a “mean” posture that describes the origin of the eigengrasp subspace. Each eigengrasp \mathbf{e}_i is a d -dimensional vector and can also be thought of as a direction of motion in joint configuration space. Motion along one eigengrasp direction will usually imply motion along all (or most) degrees of freedom of the hand.

$$\mathbf{e}_i = [e_{i,1} \ e_{i,2} \ \dots \ e_{i,d}] \quad (4.3)$$

Once this subspace is defined, a hand posture can be completely determined by the amplitude vector $\boldsymbol{\alpha} = [\alpha_1 \ \dots \ \alpha_a] \in \mathcal{R}^a$. Therefore, when hand posture optimization is performed in eigengrasp space, the grasp quality function over this subspace takes the form

$$Q = f(\boldsymbol{\alpha}, \mathbf{p}), \quad \boldsymbol{\alpha} \in \mathcal{R}^a, \quad \mathbf{p} \in \mathcal{R}^6 \quad (4.4)$$

where $\boldsymbol{\alpha}$ is the vector of eigengrasp amplitudes. When operating in a 2-dimensional subspace, we therefore have a total of 8 variables to optimize, including 2 eigengrasp amplitudes and 6 variables for wrist position and orientation, independent of the particular hand model that is being used for the grasping task.

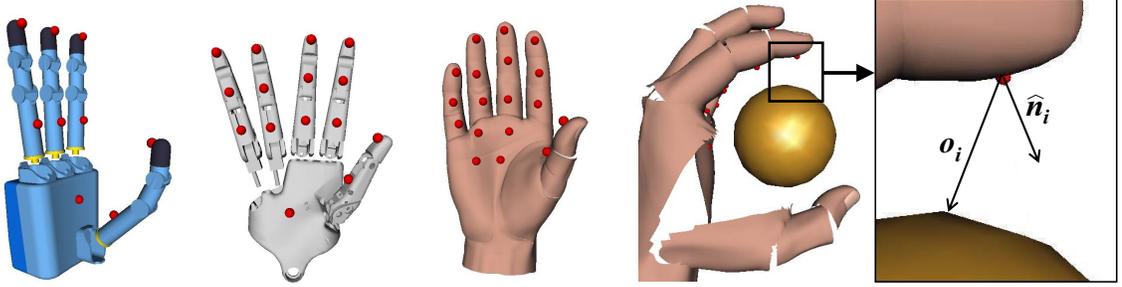


Figure 4.1: Examples of desired contact locations for posture optimization. Left: complete set of pre-defined desired contact locations for the DLR, Robonaut and Human hands. Right: for a desired contact with index i , we define the surface normal $\hat{\mathbf{n}}_i$ and the current distance to the target object \mathbf{o}_i .

4.1 Quality Function Formulation

Most grasp quality metrics that we have discussed in our literature review are based on the locations of the contacts between the hand and the target object. Our context is somewhat different: we need a quality metric that can also assess the quality of a *pre-grasp*, where the hand is very close, but not in contact with the target. For each hand model, we pre-define a number of *expected* contact locations by sampling each link of the fingers, as well as the palm, as shown in Figure 4.1. The value of the quality function is maximized for those hand postures that bring each expected contact location as close as possible to the target object. We are therefore searching for postures where the hand is wrapped around the object, generating a large contact area using all the fingers as well as the palm. As shown in Figure 4.1, for each desired contact location on the hand, identified by the index i , we define the local surface normal $\hat{\mathbf{n}}_i$ as well as the distance \mathbf{o}_i between the desired contact location and the target object. We then compute a measure Δ_i of the distance (both linear and angular) between the desired contact and the surface of the object:

$$\Delta_i = \frac{|\mathbf{o}_i|}{\gamma} + \left(1 - \frac{\hat{\mathbf{n}}_i \cdot \mathbf{o}_i}{|\mathbf{o}_i|}\right) \quad (4.5)$$

where γ is a scaling parameter required to bring the range of useful linear distances (measured in mm) in the same range as the normalized dot product between $\hat{\mathbf{n}}_i$ and \mathbf{o}_i (in our study we use a value of $\gamma = 50$). For a given hand posture, the total value of the quality function is then computed as:

$$Q = \sum_{\substack{\text{all desired} \\ \text{contacts}}} (1 - \Delta_i) \quad (4.6)$$

In most cases, the hand postures that maximize the value of Q create an enveloping grasp of the object, especially for complex models grasping objects similar in size to the hand. The optimized value of this function can be seen as a measure of how well the hand shape can be set in order to match a given object while operating in a low-dimensional subspace. In Chapter 5 we will also present an alternative quality function formulation that includes a built-in notion of grasp wrench space analysis.

4.2 Optimization Algorithm

After choosing the formulation of the quality function Q , the optimization is performed using the simulated annealing algorithm with the fast cooling schedule and neighbor generation function presented by Ingber [1989]. The stochastic nature of this algorithm makes it a particularly good choice for our task: since new states are generated as random neighbors of the current state, computation of the quality function gradient is not necessary, and the algorithm works well on non-linear functions. Furthermore, the possibility of a “downhill move” to a state of lower quality allows it to escape local optima which can trap greedier methods such as gradient ascent.

The complete optimization procedure is presented in Algorithm 1, which uses the following conventions. The variables that make up a given state (such as CurrentState or NewState) are the entries of the eigengrasp amplitude vector $\boldsymbol{\alpha}$ and the hand position vector \mathbf{p} . These variables are the target of the optimization. The Quality function for a given hand state is computed as in (4.6). We have imple-

Algorithm 1 Simulated annealing over a grasp quality function.

```

for all variables of CurrentState do
    CurrentState.variable = RandomValue()
QualityCurrent = Quality(CurrentState)
Iterations = 0, QualitySaved = 0
while Iterations  $\neq$  MaxIterations do
    // Generate a legal new state as a neighbor of current state
    repeat
        // Use simulated annealing neighbor generation function
        for all variables of NewState do
            NewState.variable = Neighbor(CurrentState.variable)
            Apply ForwardKinematics(NewState)
            legalState = true
            if collisions detected or joint limits exceeded then legalState = false
        until legalState == true
        QualityNew = Quality(NewState)
        if QualityNew > QualitySaved then
            Insert NewState in SavedStatesList
            QualitySaved = lowest quality value in SavedStateList
        end if
        // Simulated annealing probability of "jumping" to new state.
        ProbabilityJump = Probability(QualityCurrent, QualityNew)
        if ProbabilityJump > 0.5 then
            CurrentState = NewState
            QualityCurrent = QualityNew
        end if
        Iterations = Iterations + 1
    end while

```

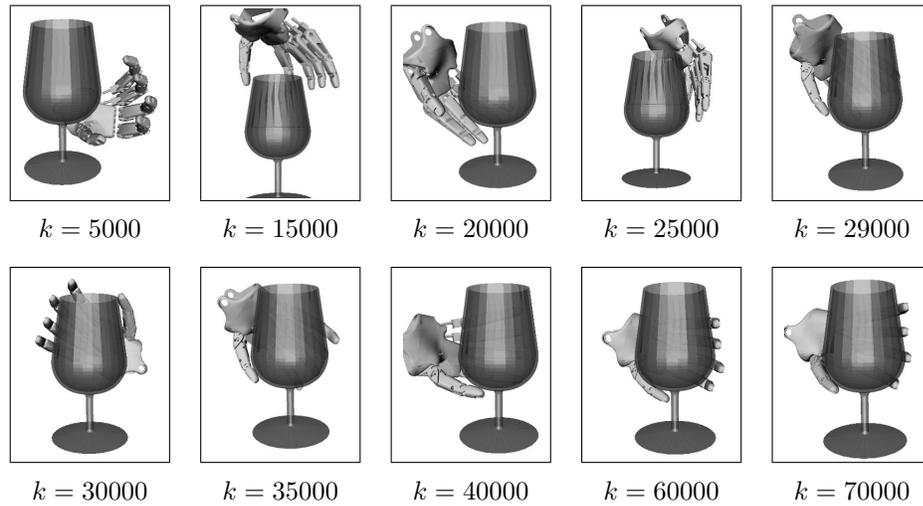


Figure 4.2: Simulated annealing example over 70,000 iterations. Each image shows the best state found until iteration k .

mented this algorithm using the *GraspIt!* simulation engine introduced in Section 2.4, which performs the `ForwardKinematics` computation and contact and collision detection. Finally, the functions `Neighbor`, for computing a “neighbor” of a variable, and `Probability`, for deciding whether a “jump” to a new state is performed, are implemented as described by Ingber [1989]. Briefly, the simulated annealing algorithm implements the following guidelines: (a) during early iterations, it allows large changes in the search variables and often jumps to worse states in order to sample the entire domain of the optimized function; (b) as the algorithm progresses, it predominantly samples increasingly smaller neighborhoods of the current solution and only allows jumps that improve its quality measure.

A detailed example of the execution of this algorithm, involving the Robonaut hand grasping a glass, is presented in Figure 4.2. The figure shows the temporary solution (best state found so far) at various points during the optimization. Figure 4.3 also shows how the current search state evolves over the full iteration range. We can observe what is considered typical behavior for a simulated annealing implementation: at first, the search goes through random states, accepting bad positions as well as good positions. As the annealing schedule progresses, the search space is sampled

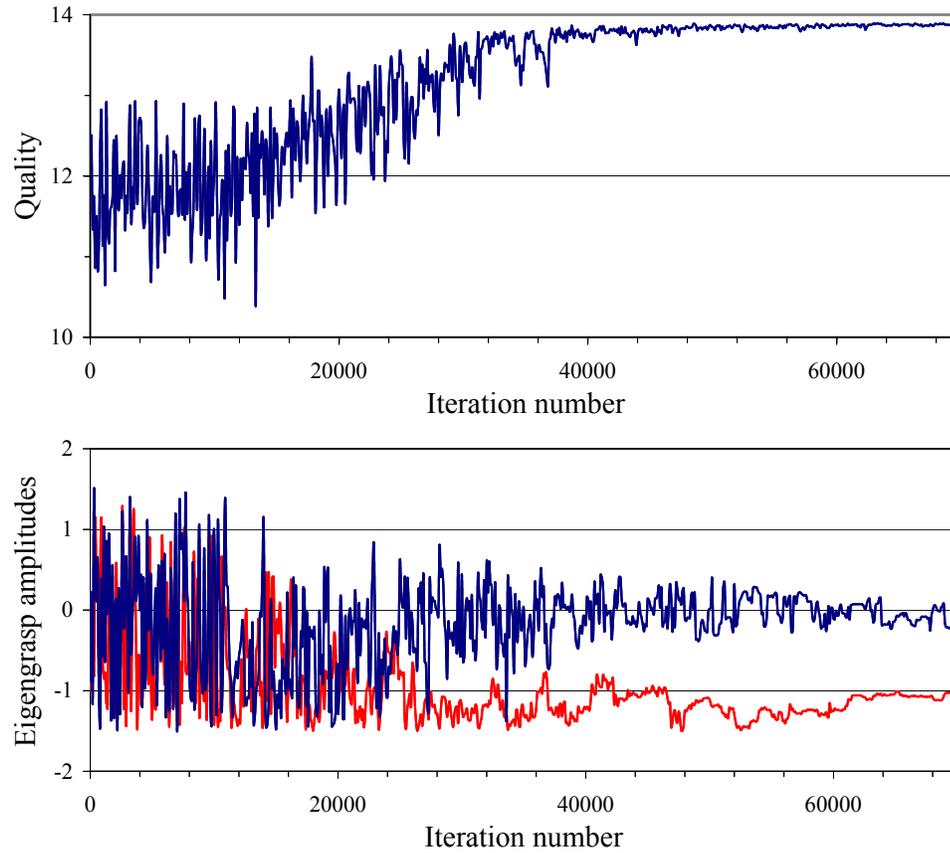


Figure 4.3: Evolution of quality function (top) and eigengrasp amplitudes (bottom) of the current search state during simulated annealing

more often in the vicinity of the good states, while bad states are no longer accepted.

Due to the stochastic nature of simulated annealing, different executions of the optimization algorithm can result in slightly different hand postures. However, the same stochastic nature enables it to “jump out” of unpredictable local optima (such as the intermediate “peaks” in Figure 4.3) and, with a high probability, converge to the same regions of the optimization space, leading to consistency between different executions. Finally, in the later stages, the search is confined in a small neighborhood around the best state, which is progressively refined. The total time required for the optimization presented here was 143 seconds, or 2.0 milliseconds per iteration, using a commodity desktop computer. The most significant amount of computation was

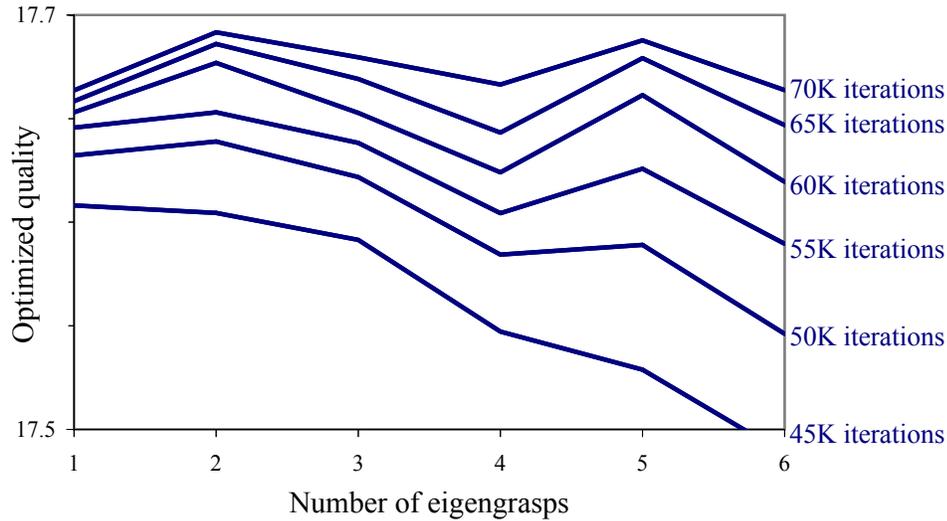


Figure 4.4: The best value of the quality function at different moments during the optimization algorithm, depending on the dimensionality of the optimization subspace. All the tests were performed using the human hand and averaged over a set of 5 executions for each of 7 test objects.

spent checking the feasibility of each generated state (*i.e.* checking for collisions and inter-penetrations). We also notice that increasing the number of iterations beyond 70,000 yields highly diminished returns; all the optimizations reported in the rest of this chapter were performed over an identical range of 70,000 iterations, or approximately 150 seconds of computation.

4.3 Optimization Results and Discussion

In this section we present quantitative testing results of the optimization method presented above. We discuss the best dimensionality of the optimization subspace, the nature of the hand postures that can be found through our optimization algorithm in this subspace, and its overall applicability for the task of dexterous grasp planning.

In order to study the impact of the dimensionality of the search space on the results of the optimization, we compared the results obtained using the human hand

in an eigengrasp subspace of dimensionality ranging from 1 to 6. All the tests were performed on a set of 7 objects with diverse geometry, such as a flask, shoe, hammer *etc.* To reduce the influence of the stochastic component of simulated annealing, the optimization for each combination of object and number of eigengrasps was repeated 5 times and the results were averaged. The complete results, showing how the value of the quality function varies with the dimensionality of the space at various points in the optimization, are presented in Figure 4.4.

The results show that, in our optimization range, a 2-dimensional subspace provides the best results. A more detailed analysis also reveals that, in the early stages of the optimization, a 1-dimensional space is qualitatively similar, while in the latter stages a higher-dimensional space can provide a viable alternative. This is an expected trend, as an increase in the dimensionality of the search space intuitively requires additional computational power to provide benefits. However, we must note that these results could also be indicative of our specific optimization algorithm, or of the set of chosen objects, rather than the intrinsic nature of the eigengrasp subspace. In particular, it is difficult to explain exceptions to the overall trend, such as the relative benefit yielded by a 5-dimensional space compared to both 4 or 6 dimensions. This is compounded by the fact that it is very difficult to find intuitive explanations for human eigengrasps ranked below the first 2. Furthermore, to the best of our knowledge, no set of objects has been accepted as a definitive benchmark of robotic grasping performance. Based on our current results, we have chosen a 2-dimensional subspace as offering the best compromise between computational effort and optimization result; all the experiments that will be presented Chapter 5 were performed using 2 eigengrasps.

In order to test the effectiveness of our framework for the task of dexterous robotic grasp planning, we have applied the 2-dimensional eigengrasp optimization using all four previously discussed robotic hand models on a set of six objects. Figure 4.5 shows the result of the annealing search for each hand-object combination. Our focus

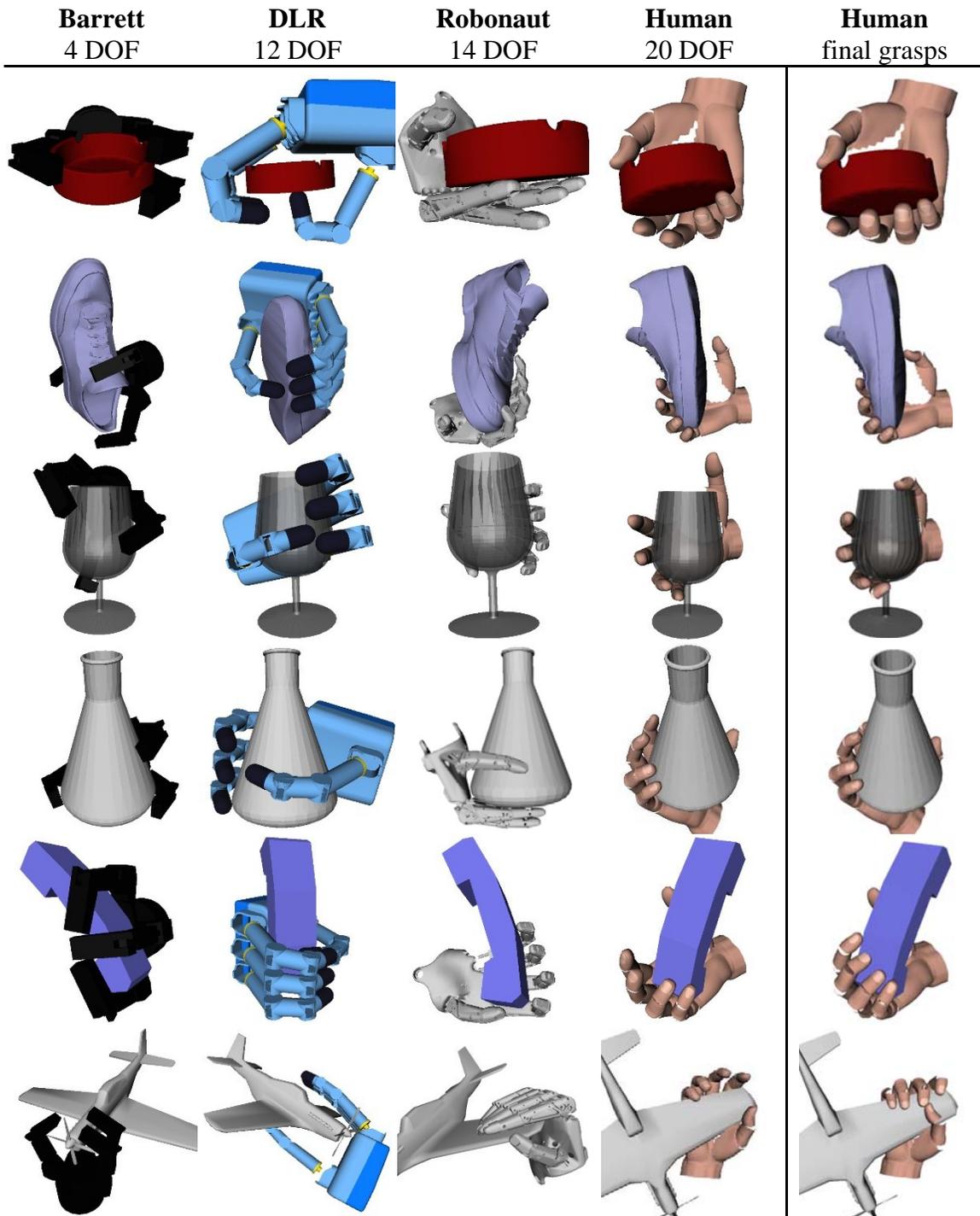


Figure 4.5: Best hand postures found in a 2-dimensional eigengrasp space using simulated annealing optimization. Rightmost column also shows final grasps obtained by closing each finger until motion is stopped by contact with the object.

	Ashtray		Shoe		Glass		Flask		Phone		Plane	
	Avg.	SD	Avg.	SD	Avg.	SD	Avg.	SD	Avg.	SD	Avg.	SD
Barrett	2.8	2.2	1.0	1.0	5.8	5.3	3.4	1.7	1.4	2.1	4.0	3.1
DLR	11.0	3.6	6.0	3.4	0.8	0.8	3.2	2.7	2.8	3.6	1.8	0.8
Robonaut	7.0	2.3	9.0	1.7	14.4	3.7	14.4	3.7	10.0	2.8	3.6	2.3
Human	14.6	2.3	11.0	2.5	11.2	1.6	13.4	3.5	8.4	2.3	1.8	1.3

Table 4.1: Number of form-closed grasps obtained from 20 pre-grasps found in a 2D eigengrasp space (average and standard deviation over 5 executions for each hand and object).

in this section is to evaluate the best hand postures that can be found in eigengrasp space. Therefore, with the exception of the rightmost column, Figure 4.5 presents the best hand posture found by the optimization algorithm without any additional refinements, allowing a direct assessment of the optimization method through visual inspection of its output.

These results show that, when the search is confined to a low-dimensional eigengrasp space, it does not reach a global optimum of the quality function where all the desired contact locations touch the target object. However, the local optimum found in eigengrasp space can be used as a *pre-grasp* by performing the additional adjustment where the hand leaves the planning subspace in order to conform to the surface of the object: execution of the binary “close all fingers” command, allowing all fingers to close until motion is stopped by contact with the object (Figure 4.5, rightmost column). We use form-closure as the analysis criterion for the resulting grasps, as our goal is the synthesis of stable grasps with no weak points.

In order to perform a quantitative analysis of the pre-grasps obtained through posture optimization, we can apply this adjustment to the 20 distinct solutions with the highest quality values found by one execution of the optimization algorithm.

We consider two solutions as being distinct if either the distance between the hand positions they define exceeds 20% of the object size, or the difference between wrist orientations exceeds 20 degrees. After closing the fingers, we count the number of distinct optimized pre-grasps that result in form closure. In order to account for the stochastic element, we repeated the test for each hand-object combination 5 times. The average number of form-closed grasps (as well as the standard deviation) for all cases are presented in Table 4.1. Each optimization was performed over 70,000 iterations, with an average running time of 158 seconds. In the case of the human hand, Figure 4.5 exemplifies this process by showing both a set of final grasps and the corresponding pre-grasps found in eigengrasp space.

These findings confirm our expectations of eigengrasp space as a *pre-grasp* or *grasp planning space*: in general, closing the fingers of a dexterous hand starting from a random configuration is not enough to obtain a stable grasp. Our results show that if the starting position is the result of the eigengrasp optimization algorithm we can obtain multiple solutions: on average, 20 optimized pre-grasps result in 7 form-closed grasps for a given hand and object. Interestingly, our algorithm performs at its best for the more dexterous designs, with kinematic structures approaching that of the human hand. This result can be explained by the fact that all the eigengrasps subspaces that we use originate from a study of human grasping. When using non-anthropomorphic robotic hands, for which this mapping is less intuitive, the effectiveness of the planning method is also decreased. Future methods for subspace mapping between hands should also take into account their relative size (for example, the palm and finger span of the DLR hand are approximately twice as large as those of its human counterpart). Overall, the results confirm our starting hypothesis: a low-dimensional algorithm can take advantage of highly dexterous hand designs for synthesizing stable grasps in a computationally efficient way.

Chapter 5

Interactive Grasp Planning for Hand Neuroprosthetics

In this chapter we focus on the problem of dexterous grasping for hand neuroprosthetics. Research in this field aims to develop prosthetic hands controlled by cortical output; this unique interface between biological and artificial systems holds great promise for those affected by loss of motor function. It also raises new challenges in designing artificial hardware and algorithms that are well suited for this type of interaction.

One of the key challenges for hand neuroprosthetics is that, as we have discussed in our literature review, recorded neural data is not expected to provide enough information to completely describe a grasp, at least for the foreseeable future. Direct cortical control of a robotic hand will require methods to a) make use of incomplete or noisy information obtained through neural intent and b) translate it to actuation of a robot with a particular (and possibly non-physiologic) kinematic configuration.

The approach that we investigate in this thesis is to integrate an automated grasp planner that acts as an interface between the operator and the artificial hand used for grasping tasks. The planner compensates for the missing information that can not be provided directly by the operator, and enables the completion of the task.

In the next sections we will outline the requirements for this component, present its implementation using the tools introduced in previous chapters, and show its application for a range of interactive grasping scenarios. We use real data from human and primate operators to complete grasping tasks using both real robotic hands and simulated environments. While we have not yet integrated this planner into a complete prosthetic system using on-line neural data, it is our directional goal. In this thesis, we aim to show that an automated grasp planner can indeed satisfy the initial requirements for such integration, and present the algorithms that we use for achieving this level of performance.

5.1 Grasp Planning in a Shared Control Framework

We consider that a grasp planning system will incorporate the following criteria in order to be appropriate for neural-prosthetic shared control:

- *functionality and interactivity*: to be able to find a stable grasp of a target object fast enough to allow for on-line interaction. In our work, we consider that computational performance is appropriate if it is fast enough for an operator to actively assist and provide input for the grasp planning task. This generally means that the planner must approach the speed of natural human behavior, and the time for the grasping task must be on the order of seconds rather than minutes.
- *usage of available user input*: to be able to produce grasping behavior that reflects partial or full control by the operator in particular degrees of freedom. We note that this requirement can be seen not only as a constraint, but also as an advantage of a shared control paradigm: the presence of user input can reduce the computational effort of finding a grasp, thus contributing to meeting

the interactivity constraint above.

- *adjustable cortical/computer control*: the level of operator control versus synthetic behavior should be directly adjustable along different dimensions, allowing both learning and adaptation to subjects of differing levels of ability. The different nature of controlling hand posture and position has also led us to make different assumptions about the level of information available for each of these components; we will expand on this topic later in this chapter.
- *biomimetic synthesis*: in the absence of complete user control, grasps will be created such that automatically controlled parameters will resemble observed physiologic behavior as much as possible.

We recall that a grasp is defined as a combination of hand posture and position. As we have noted in our literature review, current studies have demonstrated that cortical neuron activity can be decoded to provide the desired velocity of a robotic end-effector in three-dimensional Cartesian space [Taylor et al., 2002]. In our work, we make the assumption that operator data contains information for directly specifying the variables that define the hand position. Controlling finger posture has proven to be significantly more difficult. The activity of some motor neurons are correlated with finger movements, indicated by joint angles [Georgopoulos et al., 1999] or finger aperture [Artemiadis et al., 2007]. However, current results have shown success in decoding only a limited number of information channels [Taylor et al., 2003].

In our work, we propose an interactive grasp planning method that uses the eigen-grasp framework for reducing the dimensionality of the hand configuration space. This approach provides two key advantages. First, planning in a low-dimensional space allows us to achieve the computational rates necessary for interaction even in the case where hand posture is entirely controlled by the automated planner, with no operator input. Second, it enables the effective use of low-dimensional input for specifying a grasp. As an example, consider the case where the operator can only provide

one-dimensional input to the grasp planner. When operating in the full-dimensional hand posture space, specifying a desired value for a single DOF has very little practical effect in determining a particular grasp. However, the amplitude along a single eigengrasp direction can encapsulate a significant amount of the variance required for establishing a grasp.

5.2 System Overview

The central component of our grasp planning system runs on the environment provided by the *GraspIt!* simulator. The simulator receives operator input and sends it to the grasp planner, which process the input and uses *GraspIt!* as a computational platform to search for potential grasps. It is important to note that, in our current implementation, the planner requires knowledge of the target object geometry, as well as its initial position relative to the hand. It can be used in controlled environments (e.g. for operator training); for unstructured environments it requires a complementary system for object recognition and localization, such as the one demonstrated by Kragic et al. [2001].

To illustrate the behavior of the complete system, an example application is shown in Figure 5.1. Here, the goal is to execute a grasping task in the case where the operator can directly set the position of the hand, but **has no control over finger posture**. Even though the grasp planner runs in a virtual environment, its application is not confined to simulation. Based on how the results are displayed and utilized, we distinguish between two usage scenarios:

- grasp planning results are used to perform grasping tasks using a real robotic hand. The operator can hold the hand and approach the target object; the position of the hand relative to the target is tracked using a Flock of Birds (Ascension Corp., VA) magnetic tracker, and provided as input to the automated planner.

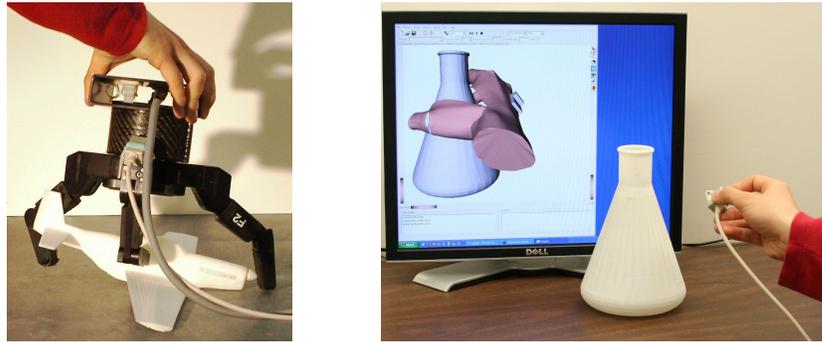
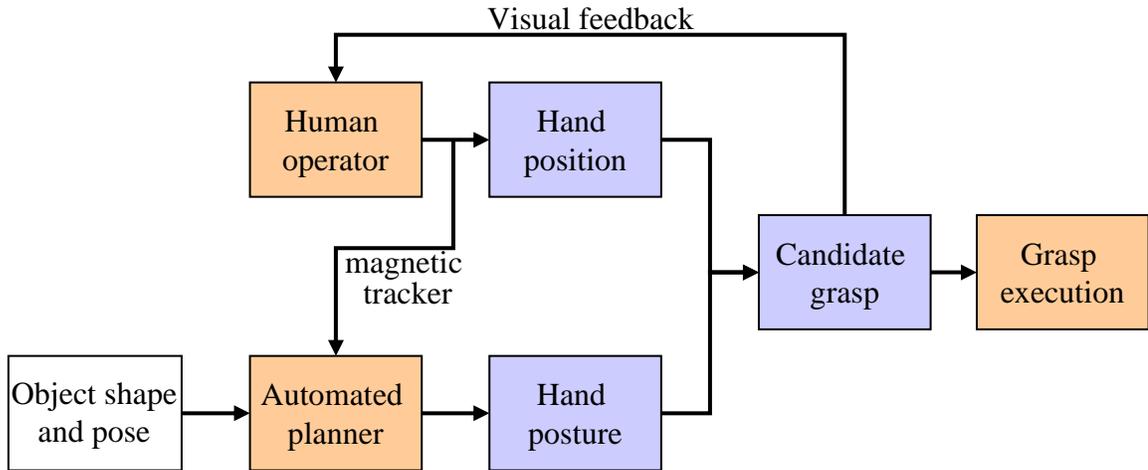


Figure 5.1: Interactive grasp planning using hand position input from a human operator. **Top**: system overview; **Bottom**: applied examples using a real Barrett hand (left) and a dexterous hand in a simulated environment (right).

- grasp planning results are displayed in a virtual environment, providing the operator with visual feedback on the completion of the task. The user can change the position of the virtual hand by directly manipulating the virtual tracker. The advantage of this implementation is that it allows us to test the applicability of our approach to many dexterous hands, including a model of the human hand.

In both cases, finger posture is entirely controlled by the automatic component, which selects an appropriate hand shape by combining information about the geom-

entry and pose of the target object with the position input provided by the operator. The only additional information needed from the user is a binary “click” command for completing a grasp similar to the one described in Section 3.3.

We have used this interactive grasping application to illustrate the high-level behavior of our system. However, this is not the only application scenario for the shared control grasp planning framework that we introduce. Later in this chapter we will discuss how hand posture input provided by the operator can be included in the computation, and show how recorded data from a non-human primate can be used as planner input as part of an experimental setup for training a primate subject to operate a robotic grasping device.

5.3 Interactive Grasp Planner Implementation

So far, we have placed the automated grasp planner in the context of a complete system for executing interactive grasping tasks. We will now present the details of the grasp synthesis algorithm itself, which is based on the eigengrasp planner introduced in Chapter 4. We recall that the optimization algorithm that we rely on uses a low-dimensional subspace when searching for hand postures that match the shape of a grasped object. This is the starting point for the interactive grasp planner as well. An important difference is that the operator can specify desired values for some (or all) of the variables that are comprised in the optimization domain.

Compared to the optimization method presented in the previous section, the current system has to satisfy the criteria that we have introduced in the previous sections. Solution grasps must be found at a fast enough rate to enable on-line interaction with the operator and usage of real-time input. However, we can rely on the presence of user input to simplify the hand position component of the search, just as using a low-dimensional subspace simplifies the hand posture component. In order to complete a grasping task, the output must also be in the form of explicit form-closure grasps

rather than optimized pre-grasps. In this section, we show how the eigengrasp planner can be adapted to satisfy these requirements and be used in our shared control framework.

5.3.1 Hand Position Parameterization

In general, in order to uniquely identify a grasp, six variables are needed to specify hand position (three for translation and three more for rotation). In the context of our application, we expect the user to specify a desired approach direction to the target; however, the presence of such external input does not fully eliminate the spatial component of the grasp planning search. First, it is not practical to wait until the user has brought the hand into a final grasping position before starting the search for an appropriate finger posture, as this behavior would decrease the interactivity of the system. Rather, it is preferable to start the search early, and attempt to predict where the user intends to place the palm. Second, this prediction allows the system to offer feedback to the user: as soon as an anticipated grasp is found, the grasp planner can shape the fingers accordingly. The user can then decide if the grasp is satisfactory and either continue towards the target or choose another approach direction if the system is unable to find an acceptable solution.

This behavior can be implemented efficiently by re-parameterizing the spatial component of the grasp planner as shown in Figure 5.2. For each hand model, we define a preferred search direction \mathbf{l} based on the kinematics of the hand, usually normal to the palm. Then, starting from a hand position specified by the operator, we search for good grasps in a conical region around the search direction using 3 variables: the distance $|\mathbf{l}|$ along the approach direction, as well as two angular variables, θ and ϕ . The operator is instructed to approach the object along a direction that is generally similar to the search cone; however, the search directions are defined in order to make this a natural choice. In the examples in Figure 5.2 this means that the user is asked to keep the palm approximately facing the target, as opposed to other possibilities

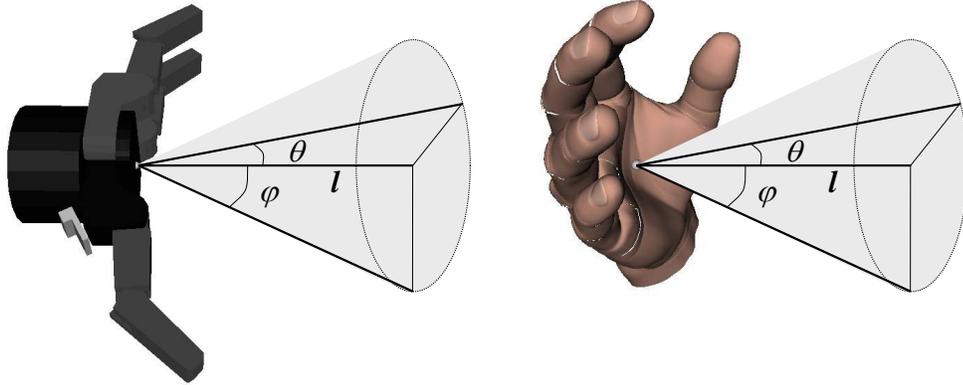


Figure 5.2: Search directions defined for the Barrett and human hand models. The direction of the vector l is predefined relative to the palm. Its magnitude, as well as the values of the angles θ and ϕ are variables defining a conical search area.

like a sideways or backwards approach.

The role of this parameterization is to reduce the number of hand position variables that are used for grasp planning, focusing on areas where good grasps are most likely to be found. Using this heuristic, the search will automatically ignore states where, for example, the hand is facing away from the target object. However, the user is not expected to specify an exact palm position for a good grasp; by searching along the approach direction l the planner attempts to anticipate the intended final grasp. The angular variables θ and ϕ allow the planner to compensate for noisy measurements in the intended hand position, and allow for more flexibility in the search for solution grasps. By adding these three variables to the eigengrasp amplitudes describing hand posture, we obtain a low-dimensional domain that can be searched fast enough to respond to on-line changes in the palm position input provided by the human operator.

5.3.2 Quality Function from Scaled Contact Wrench Spaces

When the posture optimization algorithm is used for on-line grasping tasks, we use a formulation of the quality function Q that is better adapted for interactive opera-

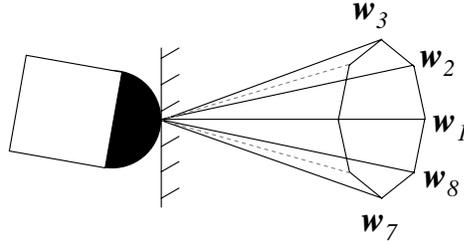


Figure 5.3: Contact wrench space example using a linearized Coulomb friction cone.

tion. Recall that, in the form presented in Chapter 4, our formulation rewards hand postures that bring all the fingers, as well as the palm, as close to the surface of the object as possible. For the application presented here, it is necessary to also reward hand postures that create stable, but not necessarily enveloping grasps (consider as an example the case of a fingertip pinch grasp applied on a thin object). We therefore propose an alternative quality function which is fast to compute and can assess the *potential* quality of a pre-grasp posture using a modified version of the Grasp Wrench Space (GWS) ϵ metric introduced by Ferrari and Canny [1992], which we reviewed in Section 2.1.2.

Starting from the linearized version of the Contact Wrench Space (CWS), for each contact i we assume that the space of forces and torques that can be transmitted is bounded by the convex hull of a finite set of 6D wrenches $\mathbf{w}_{i,j}$ where $1 \leq j \leq k$. The convex hull of these wrenches forms the CWS. For example, in the case of Coulomb friction, the force components of $\mathbf{w}_{i,j}$ sample the contact friction cone (Figure 5.3), and the respective torque components are null. In order to define the GWS, the contact wrenches from all contacts are first expressed relative to a common coordinate system. This coordinate system is usually anchored at the center of mass of the object and the choice of axes directions is arbitrary. We denote the matrix that transforms a wrench from the local coordinate system of contact i to the global object coordinate system by $\mathbf{R}_i \in \mathcal{R}^{6 \times 6}$.

In our implementation, we are usually assessing the quality of a pre-grasp shape

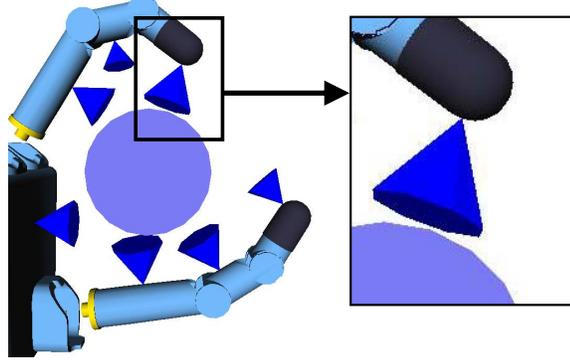


Figure 5.4: Multiple contact wrench spaces, each scaled based on the contact distance metric Δ_i .

where the fingers are not in contact with the target. Therefore, we assume that the hand can apply *potential* contact wrenches using the desired contact locations shown in the previous chapter in Figure 4.1. When computing the GWS, we scale the potential wrenches at each desired contact proportional to the inverse of the distance metric Δ_i computed as in (4.5):

$$\text{GWS} = \text{ConvexHull} \left\{ \bigcup_{\text{all desired contacts}} (1 - \Delta_i) \mathbf{R}_i \bigcup_{j=1}^k \mathbf{w}_{i,j} \right\} \quad (5.1)$$

Thus, if the value of Δ_i is small, the contact will have a significant contribution to the GWS, and states that bring it closer to the object surface will be rewarded by a higher quality value. If, on the contrary, the desired contact is far from the object, it will not significantly affect the grasp quality measurement. If the contact is far enough from the object so that its corresponding weight of $1 - \Delta_i$ is negative, it is completely excluded from the computation.

After building the scaled GWS, we compute the ϵ quality measure as described by Ferrari and Canny [1992] and Miller and Allen [1999]. The process is illustrated in Figure 5.4 for the DLR hand grasping a disc. In this example, each contact is modeled by a friction cone, approximating Coulomb friction for rigid bodies, but other local contact models can also be used. For example, we can use the Soft Finger model

reviewed in Section 2.1.2. In Chapter 6 we will also show our algorithm for efficiently computing a version of the linearized friction ellipsoid that characterizes this friction model. This method enables the use of rubber-coated fingertips for our robotic hands, without compromising the accuracy of the grasp quality computations.

5.3.3 Computation of Form-closure Grasps

The automated grasp planner searches for solution grasps in two stages. The first stage is the posture optimization algorithm presented in Chapter 4, using the quality function formulation described above. For interactive tasks, each run of the simulated annealing algorithm is performed over 2000 iterations, taking advantage of the fact that the search domain is 5-dimensional (2 eigengrasp amplitudes and 3 hand position variables), as opposed to the 8-dimensional domain used for fully autonomous searches. After reaching this number of iterations, the search is restarted by resetting the annealing temperature. As a result, the planner does not get stuck if one particular search fails; rather, the search is restarted and takes advantage of any changes in the approach direction provided by the operator.

The user-specified reference wrist position is updated continuously during the search. The results of the optimization are therefore always relative to the current position of the wrist. However, we recall that the low-dimensional optimization procedure can still only produce *pre-grasp* shapes; in order for the system to allow successful completion of the task, *final* grasping postures satisfying the form-closure requirement are necessary. In order to achieve interactive rates, this expensive computation is only performed using the best pre-grasps found during each run of the annealing optimization, which are queued and sent to the second stage of the planning process.

For each candidate pre-grasp resulting from the first stage, we use the contact detection engine within *GraspIt!* to compute the final grasp that results by closing the fingers on the object. Once the contacts between the hand and the object have

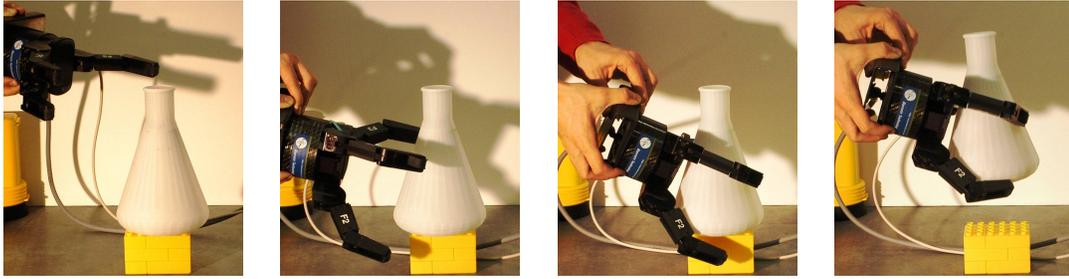


Figure 5.5: Example of a complete grasping task: initial approach, finger-preshaping using grasp planning result, continued approach and final grasp execution.

been determined, we compute the exact quality value of the final grasp using the GWS ϵ quality in its original form presented by Ferrari and Canny [1992]. If the grasp is found to have form-closure, it is saved, along with its associated quality value, as a potential solution, and used by the next component of the system, which is responsible for interaction with the human user.

When computing the final grasping posture resulting from a candidate pre-grasp, we take into account specific mechanical properties of the hand, such as passive mechanical adaptation to the shape of the target. All of the results involving the Barrett hand presented in this paper take into account its adaptive actuation mechanism which allows distal joints to close even when proximal joints controlled by the same motor have been stopped due to contact.

In our implementation, the two planning phases described in this section (simulated annealing search for pre-grasps and final grasp testing for form-closure) run in separate threads. As soon as a candidate pre-grasp is found, it is queued for testing, but the search for new candidates continues independently of the testing phase. Also, candidate pre-grasps are independent of each other, and can be tested simultaneously. This parallelism allows us to take advantage of the current evolution in multi-core architectures, largely available on standard desktop computers.

5.3.4 The Complete Planning Pipeline

We can now provide a complete step-by-step walk-through of a grasping task that combines user input and automated grasp planning. To illustrate the stages in the pipeline, Figure 5.5 shows the execution of a grasp going through the following steps:

- as the user approaches the target object, the grasp planner searches for a good grasp in a cone-shaped area around the given approach direction; when a solution is found, it is used to set the hand posture, allowing the user to react. If multiple solutions are found, the one that is closest to the current user approach direction is chosen for presentation (*i.e.*, the solution with the lowest values for the angular variables θ and ϕ).
- the planner continuously attempts to improve the current result, by finding new grasps that are closer to the current position established by the user.
- if the planner is unable to find a grasp in the current search area, or if the user is not satisfied with the resulting hand posture, the user can reposition the hand and attempt to grasp a different part of the target object.
- if the user is satisfied with the hand posture, he or she continues along the current approach direction. As the real hand position approaches the target grasp, the fingers are gradually closed around the object. The user can therefore predict where the object will be touched and finally issue a "close all fingers" command which completes the grasping task.

5.4 Interactive Grasping Examples and Analysis

Figure 5.6 presents the application of our method using the Barrett Hand in a real environment, while Figure 5.7 shows interactive grasps performed in a simulated environment using the DLR hand, the Robonaut hand and the human hand model.

In most cases, the images show only the final grasp applied by the user. In order to better evaluate the interactive nature of our application and the evolution of the grasping task from approach direction, pre-grasp and final grasp, a video clip showing a number of complete examples is also available online at <http://www.cs.columbia.edu/~cmatei/interactive>.

For any given grasping task, the exact computational effort required to find a stable grasp depends on the complexity of the hand and target object, as well as the approach direction chosen by the user. On average, the first stage of the grasp planning algorithm processes approximately 1000 hand postures per second, while the second testing phase, running in parallel, can evaluate approximately 20 candidate pre-grasps per second. In most cases, solution grasps are found at interactive rates: in the example presented in Figure 5.5, the grasp planner found 8 stable grasps in 13.6 seconds of computation. These are representative numbers for the behavior of the system, which generally requires less than 2 seconds to find a solution grasp for a new approach direction. All of our tests were performed using a commodity desktop computer equipped with a 2.13GHz Intel Core2 CPU.

The ability of the system to allow for successful task completion in a short time is more difficult to quantify, as it also depends on how well the user reacts to the behavior of the automated components. All the results presented in Figures 5.6 and 5.7, as well as in the accompanying video, were obtained at interactive rates, usually requiring between 5 and 15 seconds from first approach to final grasp execution. For the more difficult tasks, taking up to 30 seconds to complete, we found two main reasons that led to the increased execution time: either the planner repeatedly failed to find form-closure grasps for selected approach directions, or the human user could not interpret some of the finger postures selected by the planner and had to attempt different grasps. These cases represent a small minority of our tests and examples; however, the tests were performed by well-trained users familiar with the inner workings of the planning algorithm.

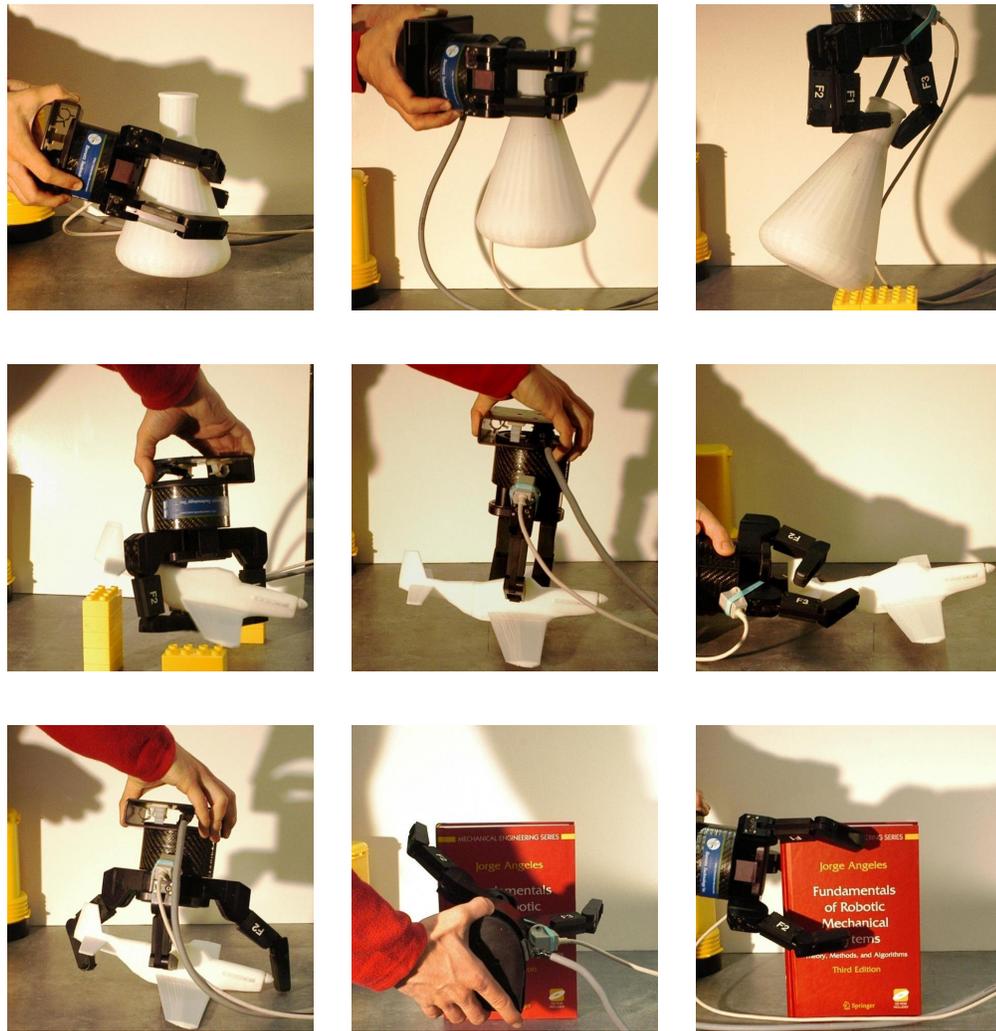


Figure 5.6: Examples of interactive grasping tasks; each image shows the grasp found for a different approach direction or target object. In all cases the object was successfully grasped and lifted off the table.

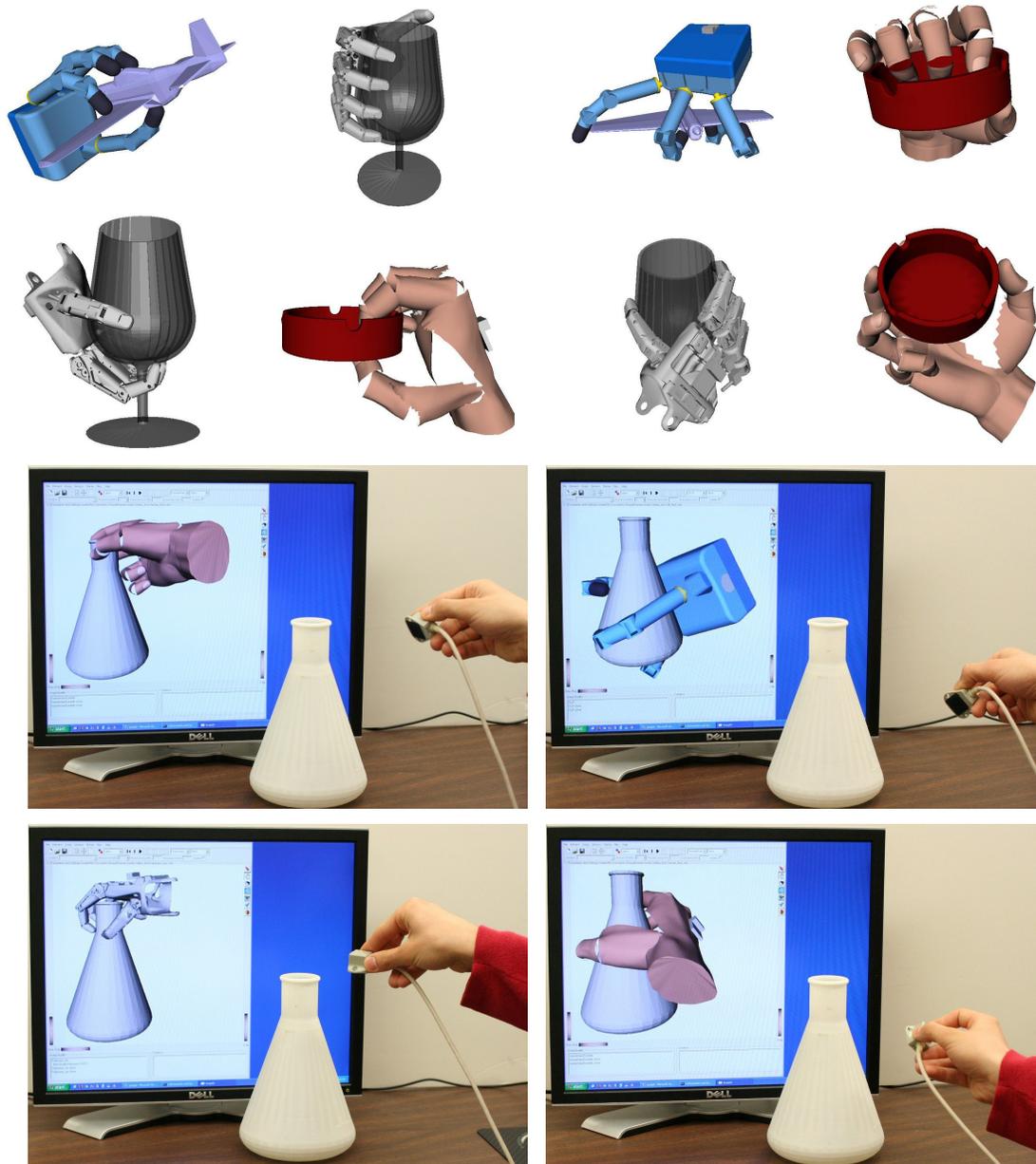


Figure 5.7: Examples of interactive grasping tasks executed in simulated environments. Bottom rows also show the user providing the approach direction via a magnetic tracker. All the presented grasps have form-closure.

5.5 User Input with Variable Confidence

So far, we have presented an application for our interactive grasp planning system where the human and automated components played distinct roles: the operator directly controlled the hand position, while the planner had sole control of the posture. While the two components were in constant interaction, each was responsible for one subset of the grasp search domain.

We now extend the interaction framework to include operator input for all the dimensions of the search, including hand posture and position. The nature of the expected hand posture input however is substantially different from the one for position. While in the previous scenario we assumed that the operator has full control over hand position, posture input must be assumed to be both low-dimensional and noisy. The eigengrasp framework is natively suited for accepting low-dimensional input, but we need additional mechanisms to cope with noise in the measurements. In this section we describe how we have augmented our planning algorithm to accept operator input with varying levels of noise, by assigning it a tunable confidence level.

An important aspect concerns transfer of operator information to a robotic hand that is non-anthropomorphic. In the previous section, we have shown how position information provided by the operator can be used for a wide range of robotic hands. However, we currently do not have a method for transferring posture information between hands with dissimilar kinematics. In our current implementation, we have applied such data only to anthropomorphic hands; application of on-line posture information to robotic models is an interesting direction for future research.

We recall from Chapter 4 and Section 5.3 that we approach interactive grasp planning as the optimization of the quality function:

$$Q = f(\boldsymbol{\alpha}, \bar{\mathbf{p}}), \boldsymbol{\alpha} \in \mathcal{R}^2, \bar{\mathbf{p}} \in \mathcal{R}^3 \quad (5.2)$$

where $\boldsymbol{\alpha}$ is the vector of eigengrasp amplitudes and $\bar{\mathbf{p}} = [l \ \theta \ \phi]$ is the reparameterized vector of hand position variables from section 5.3. We now consider the case where the

operator specifies a desired *target* value for one of the variables in this optimization. The simulated annealing approach lends itself well to using external inputs, due to its anisotropic nature: each variable is treated independently, allowing us to control the degree to which the external target values are relied upon.

Consider a variable $x \in [x_{min}, x_{max}]$ that is part of the domain of the grasp quality function Q (either an eigengrasp amplitude or a position variable). We assume that, at the annealing step k characterized by the annealing temperature T_k , the value of this variable is x_k . The algorithm requires that a “neighbor” value x_{k+1} be generated randomly for annealing step $k+1$. In general, the simulated annealing algorithm finds an optimal solution if neighbors are chosen using the following guidelines: a) among early iterations of the algorithm, it allows large changes of the search variables and samples the entire domain of the optimized function; b) as the algorithm progresses, it predominantly samples increasingly smaller neighborhoods of the current solution, performing fine-grained optimization.

The neighbor generation process is seeded by sampling a **uniform** distribution $U[-1, 1]$ to obtain a random variable u called the *generating variable*. This variable is used as input to the *neighbor generating function*

$$y_k = y(u, T_k) \in [-1, 1] \quad (5.3)$$

In our implementation, we use the generating function introduced in Ingber [1989]. Its probability density function at different annealing temperatures is shown by the solid blue line in Figure 5.8. We notice that this function is designed to satisfy criteria a) and b) presented above. After the value of y_k has been determined, the new value of x_{k+1} is generated as:

$$x_{k+1} = x_k + y_k (x_{max} - x_{min}) \quad (5.4)$$

We now assume that, for the variable x , there exists a target value x_t specified by an external operator, along with a confidence level $\sigma \in [0, 1]$, with $\sigma = 0$ meaning lowest confidence and $\sigma = 1$ meaning highest confidence. The target value x_t is first

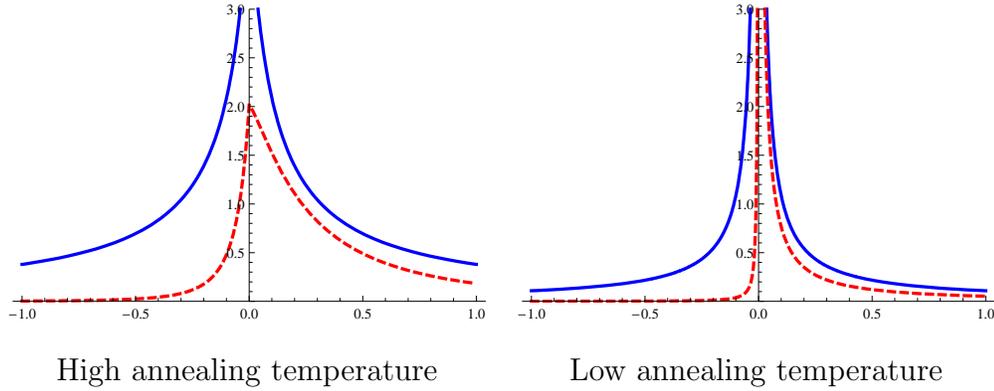


Figure 5.8: Probability density functions for simulated annealing neighbor generation. Comparison between the original formulation (solid blue line) and neighbor generation function biased towards a target jump value of 0.5 (dashed red line).

normalized relative to the current value x_k as well as the total range of the variable to obtain the value of the target jump:

$$target_jump = \frac{x_t - x_k}{x_{max} - x_{min}} \tag{5.5}$$

As can be seen from (5.4), a value of the neighbor function of $target_jump$ would create a jump such that $x_{k+1} = x_t$. The value of the generating variable u that is required for this jump is therefore:

$$u_t = y^{-1}(target_jump) \tag{5.6}$$

We are now ready to compute a new value for x_{k+1} . Again, we start with the generating variable u , but instead of a uniform distribution, we use a **normal distribution of mean u_t and variance $1 - \sigma$** (such a distribution can be obtained from the random number generator, *e.g.* using the Box-Muller transform). The generating variable u , which is now distributed mainly around the value of u_t , is used to compute the neighbor function $y(u)$ which, in turn, is used to compute x_{k+1} as discussed above. The probability density function of the neighbor generator with a u_t value of 0.5 is shown as the dashed red line in Figure 5.8; additional examples and details can be found in Appendix A.

By composing the neighbor generating function with a normal distribution centered at the desired jump value, we bias the annealing algorithm to spend more effort in the vicinity of the target. However, the stochastic nature and the main characteristics of the algorithm (large jumps early, small jumps late) are preserved. Furthermore, the algorithm can identify and refine solutions with $x \neq x_t$. By changing the value of the confidence level σ , the user can further influence the behavior of the algorithm and request that most of the effort is spent in a smaller or larger vicinity of the target. In particular, we note that if, at any point in the search, the confidence level is set to 1, the algorithm is guaranteed to jump to the target value in a single step.

The one-dimensional discussion presented here applies to each of the input variables. The user can establish independent target values and confidence levels in each of the dimensions of the quality function input, and hand position and posture can be optimized using the same framework. This allows current methods for cortical control of robot endpoint translation to be combined with cortical hand orientation and posture information, as they become available, using different confidence levels in each controlled degree of freedom. The parameter σ will be a tunable training parameter during experimentation. While not related directly to a measurable property of any control signal on an absolute scale, the value of σ for a particular degree of freedom can be heuristically increased as a subject becomes more adept at controlling it.

5.6 Grasp Planning Experiments with Hand Posture Input

We present the results of two experiments designed to test the planning system in terms of functionality, interactivity, and adjustable operator/computer control. The first experiment uses recorded data from a monkey to plan grasps in the observed eigengrasp space of a monkey hand. It is intended as an initial feasibility study for future integration of our planner in a primate cortical-control setup. In the second

experiment, we test the ability of the planner to compose synthetic grasps in near-real time using partial kinematic control provided by a human operator. In both cases, grasps are planned in simulated environments, in order to enable the use of anthropomorphic hand models. For each experiment, the planning system proceeds through the following steps:

- grasp information is recorded from monkey or human operator. Recorded data includes all finger joint angles as well as wrist position and orientation.
- the recorded operator hand posture is projected into a low-dimensional eigengrasp subspace, resulting in a set of eigengrasp amplitudes.
- a partial description of the grasp, containing eigengrasp amplitudes and wrist position and orientation, is provided to the planner at different levels of confidence. We note that, instead of the complete set of 24 DOFs that can exactly identify the input grasp, the planner is only provided with a noisy and very low-dimensional approximation of the hand posture, simulating the level of information that is expected to be available through neural recordings.
- the planner searches for a form-closure grasp of the object given the input data.

Once a solution is found, we directly compare the planned grasp against the complete description of the recorded, or “live” grasp. We measure the ability of the system to adapt to noisy and incomplete input, as well as the average time it requires to compose stable grasps.

Unlike the interactive grasping experiments of Section 5.4 where the operator directly set the hand position, in the current application scenario the operator does not have direct control over any of the variables involved in the grasp. His role is to provide input to the automated planner and to oversee the completion of the task. This mode of operation is described in Figure 5.9, and can be contrasted with the previous implementation as shown in Figure 5.1.

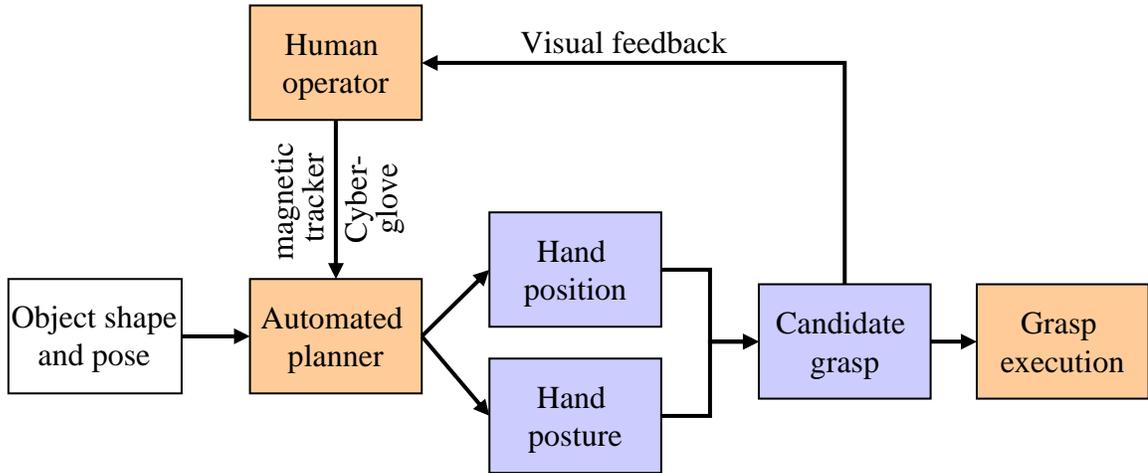


Figure 5.9: Interactive grasp planning using hand posture and position input, but no direct control from operator.

5.6.1 Recorded Input from Non-human Primate

Monkey hand kinematic information was collected from a rhesus monkey fitted with a customized glove mounted with 23 reflective markers on the right hand and lower arm. Hand motion was recorded by a Vicon motion analysis system while an industrial robot presented objects of different shapes and orientations within reach of the subject. The experimental setup, as well as the set of objects used, are shown in Figure 5.10. At the beginning of each trial, the monkey was trained to reach and grasp the presented object, squeezing top and bottom mounted pressure sensors. If force greater than a threshold registered on the sensors, the trial was saved and the monkey was given a water reward. Marker data was then processed in order to derive a kinematic model of the monkey hand and measure the angles of each joint of the hand during entire trials. Principal Component Analysis (PCA) was applied to the joint angle data to find the eigengrasps that characterize monkey hand motion during grasping. PCA results showed that a 3-dimensional subspace contains 85% of the variance in hand posture, suggesting the use of 3 eigengrasp amplitudes for grasp planning experiments.

Recorded grasps were provided as input to the automated grasp planner, as dis-

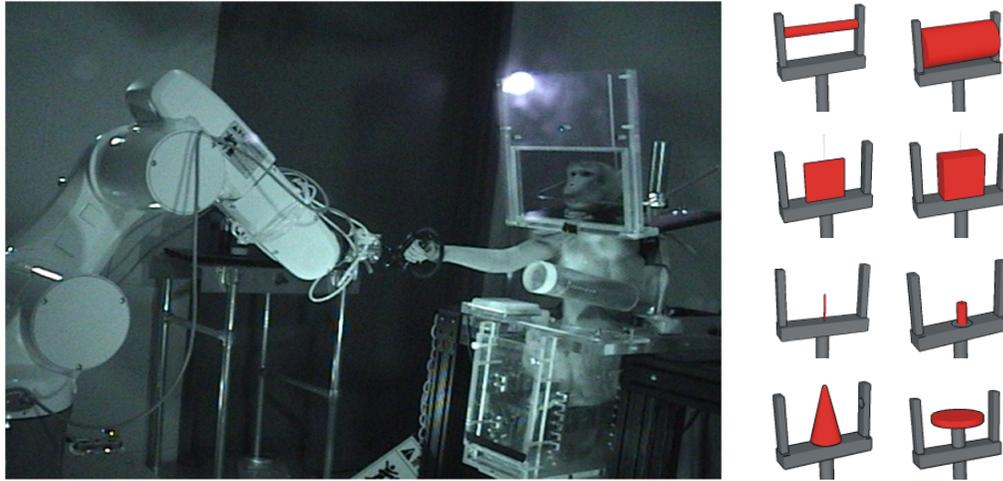


Figure 5.10: Experimental setup and object set used for recording primate grasps.

cussed above. We also varied the level of confidence in the input data, simulating incomplete or low-fidelity external control, and resulting grasps were compared to recorded ones. To illustrate this process, two examples are shown in Figure 5.11. These examples are representative for the general behavior of the grasp planner: when the input pose is used with a high confidence level, the generated grasp is in the vicinity of the target posture. However, in the absence of on-line input, the planner can generate grasps at random locations around the object.

In order to quantify the robustness of the system as well as its sensitivity to the recorded input, we compared the values of generated grasp variables to ones recorded from the monkey for a large set of more than 500 planned grasps, using a large variety of input postures over the complete set of test objects. The results, presented in Figure 5.12, show the mean difference in selected variables between generated and recorded grasps, normalized to the maximum range of each variable. To account for the stochastic nature of the algorithm, the results were averaged over multiple form-closure grasps for each combination of target object and level of confidence. The results show that planned hand orientation (red dashed line) was very sensitive to the confidence level specified by the user; the distance between the orientation of the planned form-closure grasps and the orientation specified as input decreased towards

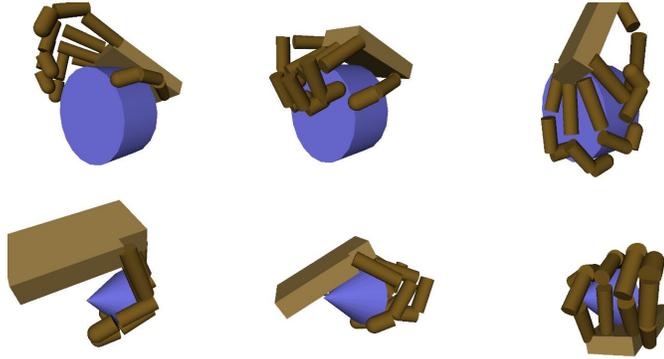


Figure 5.11: Examples of form-closure grasps returned by the planner for noisy target postures specified as input. **Left:** recorded monkey grasps used as reference poses. **Middle:** grasps planned using reference pose as input with high confidence level ($\sigma = 0.95$). **Right:** grasps planned without using reference pose.

0 as the confidence level approached 1. On the other hand, even with high confidence levels, the system did not find form-closure grasps where the hand conformed *exactly* to the eigengrasp amplitudes specified in the input (blue solid line). However, the system was effective in finding form-closure grasps within a given neighborhood of the specified eigengrasp input. This result can be partly explained by the fact that the geometry of the monkey hand model is not exact, so slightly different grasps will be needed to achieve form-closure.

In our experiments, the average time required to find a form-closure grasp using operator input was 3.3 seconds, approaching the speed required for real-time operation. All the experiments were performed on a commodity computer with a dual-core Intel Pentium 1.8 GHz processor. As shown in Figure 5.12, computation time generally increased with tighter bounds placed on the given inputs. While these results appear counter-intuitive, we note that the presence of on-line input can be restrictive, requiring the planner to find a particular solution in line with operator intent, rather than opening up entire ranges of variables for sampling. In the final real-time system, situations where the planner cannot find a form-closure grasp in a given amount of time will be aborted; an important aspect of training is letting the execution of

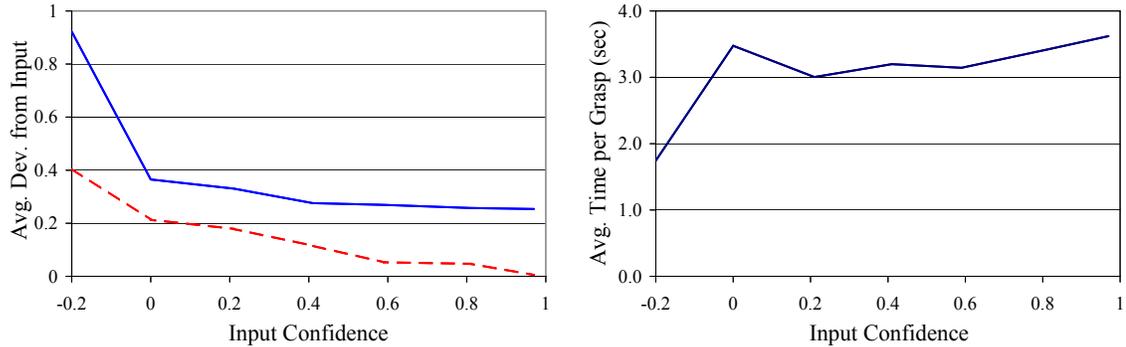


Figure 5.12: **Left:** difference between planned grasps and input grasps. Solid line shows the difference in the amplitude of the first eigengrasp, while dashed line shows the difference in wrist orientation. **Right:** average time spent to find each form-closure grasp. A value of -0.2 was used as a starting point for the Input Confidence axes to represent the case where planning was carried out without any kind of input.

the grasp fail if the extrinsically controlled variables are too far from a form-closure solution for the automated planner to be effective.

5.6.2 On-line Input from Human Operator

In the case of grasp planning experiments using data from a human operator, we again used the 2-dimensional eigengrasp subspace presented by Santello et al. [1998]. Unlike the monkey experiments, in this case the human operator interacted directly with our system: as the operator grasped a target object, on-line data was provided to the automated planner in real time using a Cyberglove and a Flock of Birds magnetic tracker. Due to the low-dimensional posture representation, as well as measurement noise, this data only provided an approximation of the actual grasp.

This system can be used interactively by presenting the planned grasps to the operator as soon as they are computed. This allows the operator to assess the connection between the example grasp and the planned result, effectively learning to grasp using very few dimensions of hand control. A more extensive set of examples of this interaction, showing the operator’s hand, as well as the output of the system

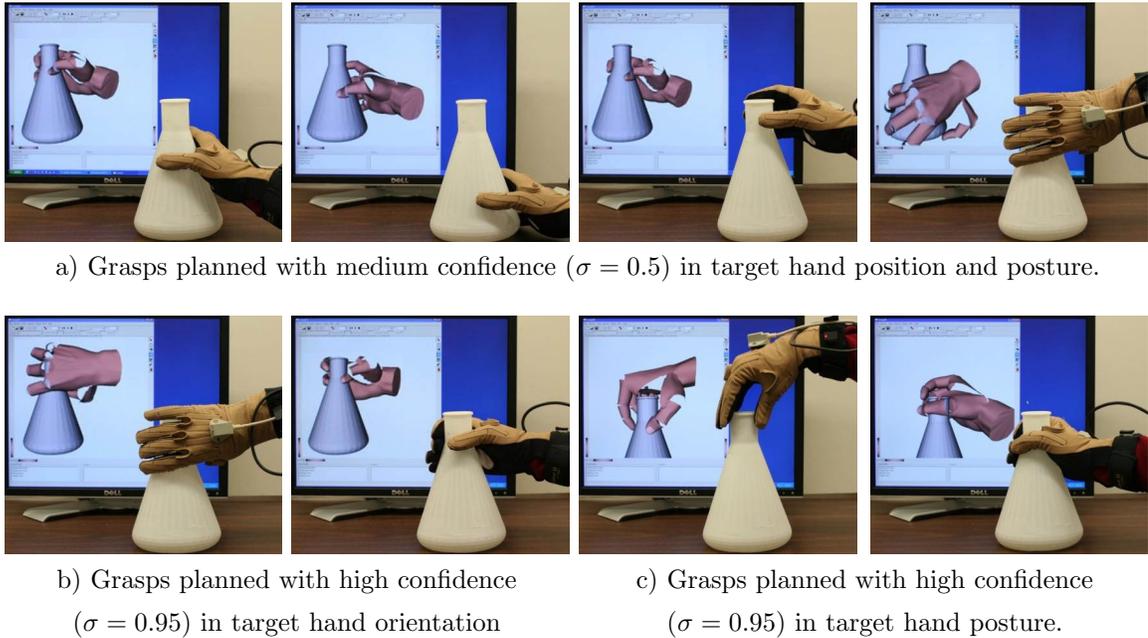


Figure 5.13: Examples of interactive grasp planning using input provided by a human operator using a Cyberglove and a magnetic tracker.

is presented in Figure 5.13.

The responsiveness of the system was measured by attempting a large number of grasps of a target object, with different hand postures as well as approach directions. For a given target grasp, if the planner failed to provide a form-closure solution within 10 seconds, the attempt was deemed a failure and the operator selected a new approach direction. In our test, the planner succeeded in finding a solution for the operator-provided target posture in 86% of the cases (55 out of 64). The average time required to find a new form-closure grasp for a given target pose was 3.9 seconds.

5.7 Discussion and Limitations

We started this chapter by identifying the need for a smart interface between the biologic and artificial components of a hand neuroprosthetic system. To fill this role, we proposed an automated grasp planner based on our low-dimensional eigengrasp

framework. We presented our algorithms which allow the planner to operate interactively while accepting and using on-line input from an operator. We also presented a number of application scenarios, in both real and simulated environments and using input from both human and non-human primate operators. We believe that our results show that this approach holds promise for future integration into a complete cortically controlled prosthetic system. However, the current implementation has also underlined a number of limitations and uncovered some interesting research questions.

A key requirement for our interactive planner in its current version is knowledge of the grasped object's shape and position. Such complete object models require complex sensing capabilities and time to generate (e.g. by combining multiple range scans of an object to eliminate occlusions). One alternative, performing object recognition and pose determination, requires that the grasped object be part of a pre-defined set. In the future, we would like to be able to design a system for unstructured settings where all the data needed by the grasp planner can be obtained fast, using a fully integrated sensing mechanism.

In the next chapter, we will discuss an approach that has the potential to greatly reduce the need for on-line sensing when executing grasping tasks. To achieve this, we propose using off-line optimization of a passively adaptive hand mechanism to increase the range of successful grasps, even in the case of low-dimensional user control and incomplete or noisy sensor data. We believe that the interactive grasp planning system presented in this chapter can greatly benefit from advances in this complementary direction.

The human operator tests presented in this chapter were all performed by well-trained users familiar with the inner workings of the planning algorithm. As a next development step, we also intend to test our system in human user studies with untrained subjects. These studies will allow us to quantify more precisely how the interaction paradigms that we have chosen affect user experience. It is clear that providing the operator with more means to influence the behavior of the planner

would be beneficial. We also believe that the communication channel can be improved in the other direction, by giving the operator more cues about the current state of the planner. Such interaction features must complement improvements to the core planning algorithm itself.

Two important questions regard the eigengrasp space that is the core of our planning algorithm. As we have mentioned before, transferring low-dimensional eigen-grasp data from the operator to the prosthesis currently means that the artificial hand must have the same kinematic model as its biological counterpart. An interesting problem to be faced is determining the set of low-dimensional eigengrasps that would allow this transfer of information if the robotic hand kinematic structure is substantially different from the biological one.

Furthermore, we have predicated our ability of extracting useful posture information from few channels of communication on using the posture subspace dimensions (eigengrasps) that encapsulate most of the variance. It is currently unclear to what degree actual neural recordings will match these eigengrasp directions. It is possible that our subspace will need to be adapted to the nature of cortical data that is available. In a sense, our framework attempts to provide an interaction between the human brain and a set of artificial control algorithms. How far each of these two components will have to come, and where in between the meeting point will be, are exciting questions, the answers to which lie ahead of us.

Chapter 6

Analysis and Optimization for Underactuated Adaptive Hands

In recent years, research on robotic grasping has focused increasing attention on passively adaptive hands. Such designs are usually defined as having the ability to passively comply to the shape of a grasped object, at a mechanical rather than computational level. As a result, they require less complex control algorithms, which in turn reduces the need for extensive sensing capabilities and increases reliability in unstructured environments.

While the ability to compensate for sensing errors is certainly important, it is another intrinsic advantage of passively adaptive hands that makes them particularly relevant to the topic of this thesis. Since fine posture adjustments are performed through passive compliance, a mechanically adaptive hand can afford to use fewer actuators than a non-adaptive model. The combination of mechanical adaptation and underactuation promises to result in robotic hands that are effective in unstructured environment, while maintaining a low production cost and enabling fast design iterations.

6.1 Eigengrasps and Compliant Underactuation

In the previous chapters, we have introduced and used grasp planning algorithms that operate in low-dimensional hand posture subspaces. We have shown how this approach offers computational advantages and enables low-dimensional operator interaction. Underactuated hands can provide an additional piece to this puzzle: a hardware implementation of the eigengrasp concept. Underactuation would be complemented by passive adaptation, used during the final refinement step, when the hand leaves the eigengrasp space to conform exactly to the shape of the target object. The ideal "eigenhand", constructed based on these principles, would provide much of the grasping ability of a fully-actuated, dexterous hand, but with a simpler design and at a fraction of the cost.

While the combination of underactuated adaptive hands and low-dimensional grasp algorithms certainly seems natural and promising, the road to a practical implementation crosses a number of important research questions that we must address before realizing this potential. One of the most important ones is the relationship between the desired co-actuation scheme, in our case the eigengrasp subspace, and the physical transmission mechanism of the hand: the computation of the eigengrasp subspace must be constrained by what is achievable in practice. So far, we have freely used eigengrasps requiring any (linear) relationship between co-actuated joints. This can certainly be achieved using a fully instrumented hand, but there is no guarantee that any particular co-actuation scheme can be implemented in an underactuated model.

Even if a desired underactuation scheme is designed and implemented, passive adaptation capabilities do not come for free. A number of commonly used transmission mechanisms prevent mechanical compliance and must therefore be avoided (*e.g.* traditional gear assemblies, fixed pulleys with inelastic tendons, *etc.*). In our literature review we have mentioned some of the alternatives that do enable passive adaptation to be implemented in practice (*e.g.* tendon-driven compliant joints, four-bar linkages

and breakaway transmissions). However, these approaches pose their own constraints on the kinematic chain and the co-actuation schemes that can be physically built, requiring careful optimization of the hand design parameters. Furthermore, passive adaptation must also be accounted for at the fingertip level, for robotic hands equipped with compliant fingertips. This phenomenon greatly increases the ability to create stable, encompassing grasps with subsets of fingers: by matching the shape of the grasped object and creating a larger contact area, soft fingertips are able to apply a larger space of frictional forces and moments than their rigid counterparts.

In general, a common thread for designing and using adaptive hand designs is that on-line sensing and computation efforts specific to a particular grasp must be complemented by off-line analysis and optimization, carried out before the hand is even built, in order to ensure positive outcome for an entire range of tasks. Interestingly, the results of this optimization effort are easiest to overlook when it is most successful, and produces a deceptively simple yet highly efficient hand.

In this chapter we introduce a number of analysis and optimization tools for underactuated compliant hands. We focus on compliance at the kinematic chain level, and propose a quasistatic analysis method for underactuated compliant hands. We present a number of applications of this method, focusing on both on-line computation specific to a particular grasping task and off-line optimization to increase the range of grasps that can be performed using a given hand model. In the next chapter, we will also show how we have extended our system to evaluate local compliance, achieved at the point of contact between a soft fingertip and the grasped object.

6.2 Underactuated Grasping as a Constrained Optimization Problem

The starting point for our optimization framework is the quasistatic equilibrium relationship that characterizes a stable grasp, which we have introduced in Section 2.1.3.

We briefly review the general formulation here, then extend it to the case of under-actuated compliant hands.

We have already noted that the passive adaptation concept can be implemented in hardware using multiple actuation methods; the choice of which method to use is one of the first decisions to be made when starting the design of a passively compliant hand. In this thesis, we construct our framework using the mechanics of a tendon-actuated hand combined with compliant, spring-like joints. This allows us to provide a concrete example and implementation of the optimization results. We found the relative ease of constructing a prototype using this actuation paradigm particularly appealing, and believe it has the potential to lower the barriers for experimenting with new hand designs and disseminating research results. However, other actuation methods have their own merits, which must be considered in future iterations.

6.2.1 Contact Constraints

Consider a grasp with p contacts established between the hand and the target object. For any contact i , the total contact wrench \mathbf{c}_i must obey two constraints. First, the normal component must be positive (contacts can only push, not pull). Second, total contact wrench must obey friction laws. In order to model these constraints, we start from the linearized formulation introduced by Anitescu and Potra [1997] and further discussed by Miller and Christensen [2003], which we review here using the Coulomb friction model for illustration.

Coulomb friction constraints state that tangential forces at the contact are limited by the friction coefficient μ_i as well as the magnitude n_i of the force applied in the direction of the contacts normal $\hat{\mathbf{n}}_i$. From a geometric standpoint, the tangential friction component of the contact force has to lie inside a "friction circle" of radius $\mu_i n_i$. This constraint can be linearly approximated by expressing the frictional component of the contact force as a weighted linear combination of k discrete vectors on

the boundary of the friction circle:

$$\mathbf{c}_i = [\hat{\mathbf{n}}_i \mathbf{D}_i] [n_i \boldsymbol{\beta}_i]^T \quad (6.1)$$

Here the columns of the matrix $\mathbf{D}_i \in \mathcal{R}^{3 \times k}$ are the k vectors that sample the friction circle and $\boldsymbol{\beta}_i = [\beta_i^1, \beta_i^2, \dots, \beta_i^k]$ is the vector of weights (in practice we use $k = 8$). Additionally, all the weights must be positive, and their sum is bounded by the magnitude of normal force:

$$[\mu_i - \mathbf{e}] [n_i \boldsymbol{\beta}_i]^T \geq 0 \quad (6.2)$$

$$n_i, \boldsymbol{\beta}_i \geq 0 \quad (6.3)$$

where $\mathbf{e} = [1, 1, \dots, 1] \in \mathcal{R}^k$.

Constraints (6.1) through (6.3) refer to a single contact i . We now assemble them in matrix form for the complete system:

$$\mathbf{c} = \mathbf{D}\boldsymbol{\beta} \quad (6.4)$$

$$\boldsymbol{\beta}, \mathbf{F}\boldsymbol{\beta} \geq 0 \quad (6.5)$$

where the vector of unknowns $\boldsymbol{\beta}$ contains the entries $[n_i \boldsymbol{\beta}_i]^T$ for $i = 1 \dots p$ in block column form, the matrix \mathbf{D} contains the entries $[\hat{\mathbf{n}}_i \mathbf{D}_i]$ in block diagonal form and the matrix \mathbf{F} contains the entries $[\mu_i - \mathbf{e}]$ also in block diagonal form.

An important advantage of this formulation is that it can be directly extended to consider different friction models by simply changing the set of vectors that are linearly combined to compute the friction component of the contact wrench. In the next chapter we will show how we can compute the appropriate entries in the \mathbf{D}_i matrices to capture more complex frictional phenomena, such as soft finger contacts which can also apply frictional torque in addition to tangential friction.

6.2.2 Actuation Constraints

We can now move on to the analysis of the complete grasp, as a collection of multiple contacts. In general, a grasp is in equilibrium if the following conditions are satisfied:

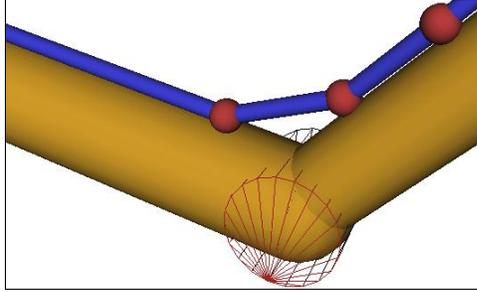


Figure 6.1: Illustration of tendon routing points, marked with red spheres, as the tendon follows a revolute joint marked by a wire frame cylinder.

- contact forces are balanced by joint forces (hand is in equilibrium);
- resultant object wrench is null (object is in equilibrium);
- contact constraints are met for all contacts that are comprised in the grasp.

We can assemble this grasp description into the following formulation:

$$\mathbf{J}_c^T \mathbf{D}\boldsymbol{\beta} = \boldsymbol{\tau} \quad (6.6)$$

$$\mathbf{G}\boldsymbol{\beta} = 0 \quad (6.7)$$

$$\boldsymbol{\beta}, \mathbf{F}\boldsymbol{\beta} \geq 0 \quad (6.8)$$

where $\boldsymbol{\tau}$ is the vector of joint forces, \mathbf{J}_c is the Jacobian of the contact locations and \mathbf{G} is the grasp map matrix which relates individual contact wrenches to the resultant object wrench.

So far, this analysis applies to a hand design regardless of its actuation method. To adapt it to the case of underactuated hands, we must look in more detail at the joint force vector $\boldsymbol{\tau}$, which is a result of the actuation mechanism. As we have discussed before, in this study we chose to focus on an actuation method that combines tendons and spring-like compliant joints. We use the common tendon-pulley model which assumes that the tendon travels through a number of routing points that it can slide through, but which force it to change direction as it follows the kinematic

structure. As a result of this change in direction, the routing points are the locations where the tendon applies force to the links of the finger. This model is illustrated in Figure 6.1, with the routing points marked with spheres. For clarity, the route shown is on the surface of the links, but in general the tendon can also be tunneled through the inside of the links.

We assume that the hand contains a total of d tendons, each with multiple routing points across different links. In this case, joint forces can be expressed as

$$\boldsymbol{\tau} = \mathbf{J}_d^T \boldsymbol{\delta} + \boldsymbol{\theta} \mathbf{k} \quad (6.9)$$

where \mathbf{J}_d is the Jacobian of the tendon routing points and $\boldsymbol{\delta} \in \mathcal{R}^d$ is the vector of applied tendon forces. $\boldsymbol{\theta}$ is a diagonal matrix of joint angle values and \mathbf{k} is the vector of joint spring stiffnesses (without loss of generality, we assume 0 is the rest position for all springs).

6.2.3 Quasistatic Grasp Equilibrium Formulation

By combining equations (6.6) through (6.9), we obtain a complete description of the equilibrium state of the grasp:

$$\mathbf{J}_c^T \mathbf{D} \boldsymbol{\beta} = \mathbf{J}_d^T \boldsymbol{\delta} + \boldsymbol{\theta} \mathbf{k} \quad (6.10)$$

$$\mathbf{G} \boldsymbol{\beta} = 0 \quad (6.11)$$

$$\boldsymbol{\delta}, \boldsymbol{\beta}, \mathbf{F} \boldsymbol{\beta} \geq 0 \quad (6.12)$$

In practice, one of the conditions above is used as an optimization objective, rather than a hard constraint, with two important advantages. First, it provides more information for problems where all the constraints are not feasible in their exact form. Second, problems that have a solution in the exact form will often have an infinity of solutions; formulating an optimization objective allows us to choose the optimal one. Which of the above constraints is to be used as an optimization objective depends

on the nature of the problem; we will provide a number of concrete examples in the following sections.

As a result, this formulation is extremely versatile, and can be adapted to a number of practical problems in underactuated grasp analysis. For example, the set of unknowns can also be chosen depending on the problem:

- if the unknown variables include only the contact wrench magnitudes β (and implicitly all the individual contact wrenches \mathbf{c}_i), we are computing whether a particular set of actuator forces results in a stable grasp;
- if we extend the set of unknowns to also include the vector δ , we are trying to compute the best actuator forces for a grasp characterized by a particular set of contacts;
- we can even extend the set of unknowns to include components of \mathbf{J}_d^T or \mathbf{k} , in which case we are computing the best hand design parameters for executing a given grasp (or set of grasps).

6.3 Grasp Analysis for Underactuated Hands

An important aspect of underactuated grasping is that different fingers, as well as different links within a finger, make contact with the object at different times. With a fully actuated robot equipped with ideal sensors this phenomenon can be detected and the motor forces modulated so that the hand continues to close without applying any force at these contacts. If the hand lacks the actuation mechanism needed to perform precise modulation of contact forces, the links that have already made contact will apply some level of force to the object as the hand continues to close.

We can analyze this process by re-formulating the equilibrium conditions as follows. For a given hand posture and set of contacts, the goal is to determine the contact forces β and actuation forces δ that balance the system, or, if exact equilibrium is

not feasible, result in the smallest magnitude wrench on the object:

$$\begin{aligned} \text{minimize } \|\mathbf{G}\boldsymbol{\beta}\| &= \boldsymbol{\beta}^T \mathbf{G}^T \mathbf{G} \boldsymbol{\beta} \text{ subject to :} \\ [\mathbf{J}_c^T \mathbf{D} \quad -\mathbf{J}_d^T] [\boldsymbol{\beta} \quad \boldsymbol{\delta}]^T &= \boldsymbol{\theta} \mathbf{k} \\ \boldsymbol{\delta}, \boldsymbol{\beta}, \mathbf{F}\boldsymbol{\beta} &\geq 0 \end{aligned}$$

This is a standard Quadratic Program, with linear constraints. The matrix that defines the quadratic (and only) component of the objective function is positive semidefinite by definition, as it is the product of the matrix \mathbf{G} and its transpose. Therefore, the optimization problem is convex, so whenever the conditions are feasible, a global minimum can be determined. In this study, we used the Mosek [Mosek] package to solve all the optimization problems of this form.

There are three possible results to the optimization problem presented above:

- the problem is unfeasible; this indicates that no legal contact forces exist that can balance the system. The fingers will slip on the surface of the object.
- the problem is feasible and a non-zero global optimum is found; the contacts are stable but some level of *unbalanced force* is applied to the object. If this force is not balanced externally (*i.e.* by interactions between the target object and another surface in the environment), the hand will have to reconfigure itself, also causing the object to move.
- the problem is feasible and the global optimum is zero; the contacts are stable and contact forces balance each other on the object producing a null resulting wrench. The hand-object system is stable in its current configuration.

6.3.1 Algorithm Design

We can now present the analysis method applied to a complete grasp execution. The goal is, for a given starting position and finger closing direction, to predict if the

Algorithm 2 Grasp analysis algorithm.

```

maxUnbal = 0

repeat
  close fingers incrementally until a new contact is made
  formulate contact force quadratic program
  unbal = Optimize(quadratic program)
  if program unfeasible or unbal > unbalThreshold then
    return unstableGrasp
  maxUnbal = MAX(unbal, maxUnbal)
until all fingers stopped
set desired level of actuation forces
formulate contact force quadratic program
finalUnbal = Optimize(quadratic program)
if program unfeasible then
  return unstableGrasp
graspQuality = f(maxUnbal, finalUnbal)
return graspQuality

```

outcome is a successful grasp and, if so, to also assign it a numerical quality metric. A step-by-step execution is presented in Algorithm 2.

We note that, in the presented form, our algorithm does not provide an exact formulation of the grasp quality metric, but rather suggests that it can be computed as a function of both the maximum level of unbalanced force created while closing the fingers and the unbalanced force of the final grasp, after actuator forces have been set to the desired levels. An ideal grasp will minimize both of these values. However, the weight placed on each of these goals can be adapted to the particular characteristics of the hand and the environment. In the results presented in the following sections, we chose to use only the final unbalanced force as the returned quality metric; other choices are also possible.

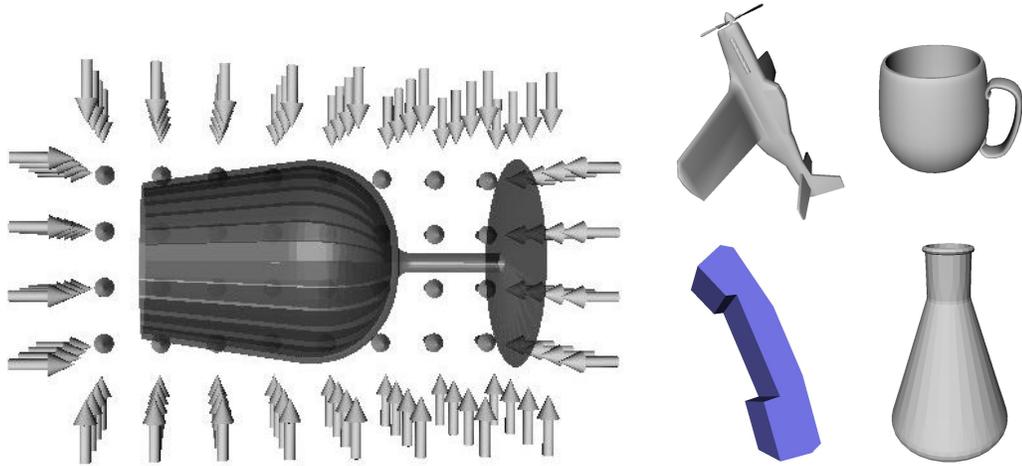


Figure 6.2: Set of five objects (glass, flask, mug, phone and toy airplane) used in our tests. The glass also shows a number of possible grasps generated by aligning the hand with its bounding box. For each possible grasp, the approach direction (shown by the arrows) was parallel to one axis of the bounding box. The rotation of the hand around the approach direction (not shown here) was set by aligning the hand with the other axes of the box.

The final grasps can also be pruned according to other quality metrics, such as the widely used Grasp Wrench Space (GWS) ϵ metric proposed by Ferrari and Canny [1992]. In our implementation, we prune all final grasps that have an ϵ value below 0.05. This metric builds the GWS by considering only contact frictional constraints. The presence of additional underactuation constraints would require further pruning of the GWS; as such, the ϵ value we currently use can be considered an upper bound for the true GWS quality of an underactuated grasp.

6.3.2 Example Application and Results

To showcase this analysis method, we used a model of the Harvard Hand introduced by Dollar and Howe [2007]. This hand uses a single actuator to drive eight joints that articulate four fingers, relying almost exclusively on passive adaptation for grasping a wide range of objects. Our planning method, implemented using the *GraspIt!*

simulation engine, goes through the following stages:

- create a large number (between 150 and 1000) of possible grasping positions for each object in our test set. Figure 6.2 shows the objects in our set, and exemplifies the sampling process for creating candidate grasps. This was done by aligning the hand with the bounding box of each object and advancing towards the object until first contact is made.
- analyze each possible grasping position using the quasistatic analysis algorithm presented in this section. Sort the grasps in order of the quality metric.
- use *GraspIt!*'s dynamics engine to simulate the execution of a grasp candidate in order to provide ground truth and a computational performance baseline. This engine, presented by Miller and Christensen [2003], uses a time step integration method to compute body velocities and accelerations in response to actuator and contact forces. If the dynamic execution of a grasp results in the object firmly held in the hand against gravity, the grasp is deemed to be successful.

To illustrate this process, Figure 6.3 shows an example using the mug as a test object. The top row shows a grasp where the hand was not centered on the object, and the handle of the mug was inside the finger span. The grasp analysis algorithm reported that contacts would slip and labeled the grasp as unstable, confirmed by the dynamic simulation. The lower row shows a grasp labeled as stable throughout its execution; dynamic simulation confirmed that the fingers fully enclosed the mug and created stable grasping forces.

Our first test was intended to provide a baseline performance measure: for each object, we tested all candidate grasps using the dynamic engine. The percentage of successful grasps over the entire set of objects was 17% (the complete results for each object are shown in Table 6.1, third column). The result shows that this hand is indeed an effective grasping device, but a random choice of approach direction yields an unsatisfactory success rate. Furthermore, the average time for complete

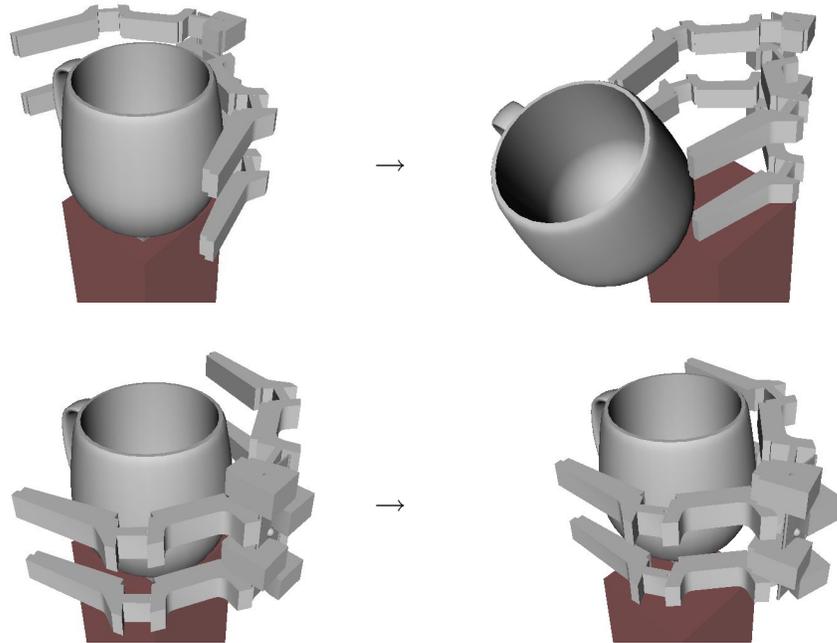


Figure 6.3: Examples of unsuccessful (top row) and successful (bottom row) candidate grasps. For both cases, the left image shows the starting position used as input to the grasp analysis algorithm and the right image shows the outcome of the grasp as computed through simulation with a time-stepping dynamic engine.

dynamic analysis of all the grasp candidates was 20 minutes per object, restricting its applicability in on-line grasping scenarios.

In contrast, the quasistatic analysis algorithm is significantly faster. In our experiments, totaling more than 1500 grasps over 5 objects, the time required for quasistatic analysis of a stable grasp ranged between 100 and 200 milliseconds. Bad grasps are evaluated even quicker, as either finger slip or a high level of unbalanced force lead to an early exit with an *unstableGrasp* label. In general, the time required for analyzing a complete set of candidate grasps densely sampled along the object bounding box ranged between 16 and 75 seconds. All of our experiments were performed on a commodity desktop workstation equipped with a 2.13GHz Intel Core2 CPU.

Object	#G	%G	#GA	%GA	time
Flask	542	13%	8	100%	45.18s
Plane	849	9%	10	90%	75.14s
Mug	337	14%	10	90%	29.56s
Phone	177	44%	10	100%	16.17s
Glass	220	39%	10	80%	36.9s

#G: total number of grasp candidates generated

%G: percentage of candidates from this list that result in a stable grasp

#GA: number of best candidates taken from the list ordered through grasp analysis

%GA: percentage of those that result in a stable grasp

time: time required to perform the grasp analysis and return the best candidates

Table 6.1: Quasistatic analysis for grasp planning.

After this analysis was done, we selected the 10 most stable grasps from the ordered list of candidates. One exception was the flask, for which only 8 stable grasps were found (this is intuitively explained by the conic shape of this object which makes it difficult for this hand model to hold against gravity). This subset was then tested using the dynamic engine. Our complete results are presented in Table 6.1. We note that the quasistatic approach provides an efficient and reliable method of pruning down a very large number of possible grasps to a small number of reliable candidates.

For application in real life environments, this method places a number of requirements on the sensing capabilities of the system. One possibility is to acquire a model of, or recognize, the object to be grasped. The method presented above can then be applied to find reliable grasps for execution. Another possible option would use tactile sensors and proprioception to analyze the grasp currently being executed. Both of these options require extensive sensing, which runs against the stated motto of simplicity and low-cost designs. An interesting alternative is to optimize the hand off-line

so that a wider range of grasps can be executed with increased robustness. We believe that complementing grasp-specific on-line computation with off-line hand design optimization can prove a fruitful direction for improving robotic grasp performance. We explore this option in the following section.

6.4 Hand Design Optimization

Using our framework to tackle a hand design optimization problem implies a somewhat different approach than we have used in the previous section. Grasp analysis usually focuses on *a single grasp* at one time, and aims to compute the optimal contact or actuator forces specific to that grasp. In contrast, the study of hand design parameters normally implies solving an optimization problem over *a set of grasps*. Indeed, a specialized hand that can perform a single type of grasping task will not be very useful in unstructured environments. Rather, our goal is to optimize performance over a range of expected scenarios.

The first step of the hand optimization method that we propose is thus to create a batch of grasps that we expect the hand to be able to perform. We refer to this set as the *optimization pool*. The optimal contact and actuator forces specific to each grasp are still unknown; now they are joined by a set of unknowns representing actuation parameters which are shared by all the grasps in the optimization pool.

The ideal scenario would intuitively be to assemble a *global optimization problem* which would allow us to directly compute the optimal design parameters over the entire optimization pool. However, such a global approach is not always possible to implement. Consider for example the problem of optimizing the location of the routing points (and thus the tendon route) on their respective links. The effects of the tendon route on the equilibrium condition are encapsulated in the Jacobian of the routing points, \mathbf{J}_d . Changing the location of a routing point on a link has a highly non-linear effect on \mathbf{J}_d . Furthermore, even if we had a linear relationship between

the tendon route parameters and the routing point Jacobian, the result must then be multiplied by the unknown vector of actuation forces δ . As a result, computing both actuation forces *and* optimal tendon route parameters results in a higher order equality constraint which can not be handled by the same optimization tools.

The general case therefore enables us to quantify a given hand design (by separately computing the quality of each individual grasp in the optimization pool), but not to directly compute a global optimum for the design parameters. We envision two possible solutions to this problem:

- a “numerical” approach, where hand performance is independently quantified for many different combinations of parameter values, and the design that yields the best results is chosen. While the computational requirements of this method may appear prohibitive, it is important to remember that: a) this analysis is performed once, at design time, with a potentially large time budget; b) with the advent of fast multi-core architectures, computation is becoming less and less expensive, particularly when it can be performed off-line and remotely; c) the “inner loop” of our hand optimization problem, individual grasp analysis, can be performed efficiently using algorithms such as the one that we introduced in this chapter.
- a “global optimization” approach, where new constraints are added to the formulation in order to cast it as a solvable optimization problem, such as a Linear or Quadratic Program. Apart from computational efficiency, this method also has the advantage of producing a provable global optimum. Its drawback is that the additional constraints that are required limit its applicability to a subset of possible hand designs.

We illustrate both of these approaches in the following sections.

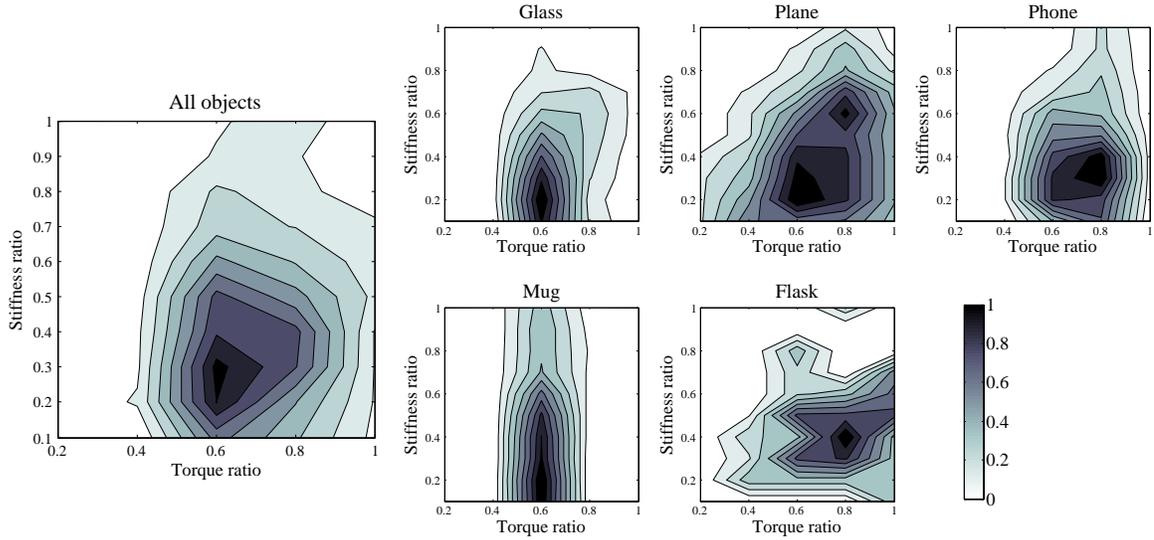


Figure 6.4: The effect of hand design parameters on the likelihood of obtaining a stable grasp. For each combination of joint torque and stiffness ratio, the color indicates the number of stable grasps obtained from the optimization pool; a darker color means a higher number of stable grasps. For each object, the results were normalized to a scale of 0 to 1 by dividing by the maximum number of grasps found for that object.

6.4.1 Numerical Optimization

We used the numerical approach to investigate how grasping performance can be improved by changing hand design parameters for the Harvard Hand. We focused on two such parameters: the actuator torque ratio between the proximal and distal joints of each finger and the spring stiffness ratio between the same joints. These parameters are determinant for the behavior of the hand, as they affect both the posture of the hand before touching an object and the forces transmitted after contact is made. In particular, we investigated all possible combinations ranging from 0.2 to 1.0 (in steps of 0.2) for the torque ratio and from 0.1 to 1.0 (in steps of 0.1) for the stiffness ratio.

The optimization pool consisted of 2000 possible grasps distributed evenly across the 5 models in our test set. All grasps were created using the same method described in Section 6.3 and illustrated in Figure 6.2. For each torque and stiffness combination,

we tested all the candidate grasps and reported the number of them that are stable throughout their execution. To enable direct comparison across different objects, each set of results was normalized to a scale of 0 to 1 through division by the maximum number of grasps found for that object. Figure 6.4 shows these results for each of the five objects, as well as their average over the entire set.

The contour maps reveal which areas of the optimization range offer the best performance; in particular they suggest a torque ratio of 0.6 and a stiffness ratio of 0.3. The overall resemblance between the patterns suggests that the global optimum of the average profile is a good compromise, likely to work well on all objects. However, the patterns exhibit enough variation to illustrate the importance of performing this analysis over a large set of models, spanning a wide range of shapes and grasping scenarios. We also note that our torque ratio is in agreement with the optimal value found in the optimization study by Dollar and Howe [2006], the results of which were used in the construction of the Harvard Hand prototype.

The focus of the present study is the analysis method itself rather than a particular design choice or optimization task. We therefore chose only one of the many aspects of a hand model that can be analyzed in similar fashion. These include kinematic chain design, link lengths and shapes, number of fingers, *etc.* In this light, the computational performance of the analysis method becomes a key aspect: a more efficient algorithm will allow for more design iterations, investigating more parameters over a larger domain. The analysis presented here consisted of a total of 20,000 grasps for each object (400 candidates for each of the 50 combinations of force and torque ratios); the typical time spent per object was 15 minutes. This performance level suggests the possibility of scaling up to significantly larger test sets.

In the applied example presented here, we used an exhaustive approach, testing the entire range of the optimized parameters. Our results show that this approach is feasible (at least for a relatively small optimization domain), but more advanced numerical optimization algorithms can also be used. Examples include simulated

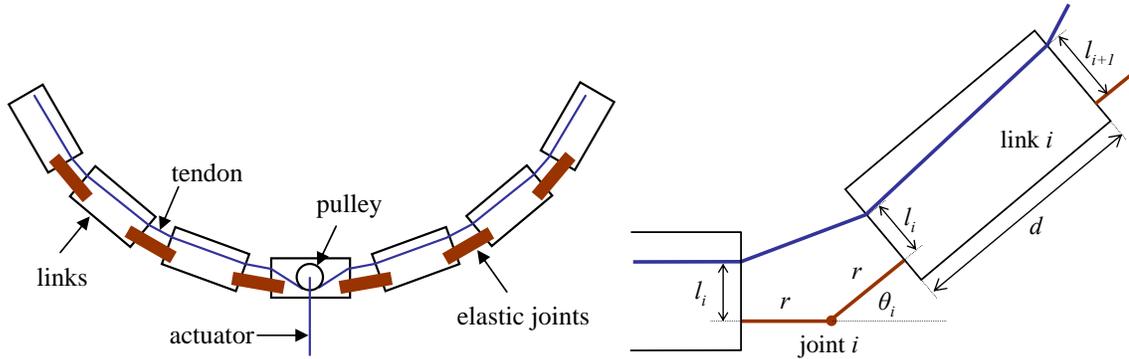


Figure 6.5: Left: a two-fingered gripper with single tendon actuation used as a case study for the optimization framework. Right: detailed description for a joint of the proposed design.

annealing or gradient ascent using numerical computation of the gradient. While such algorithms also present disadvantages (like the threat of stopping in local optima in the case of gradient ascent), they can be better equipped to handle larger optimization problems than the one we have tackled here. We intend to further pursue such possibilities as part of our future work.

6.4.2 Global Optimization: a Case Study

In order to illustrate our global optimization approach to the hand design problem, we will build up a concrete example, using as a testbed a two-finger model (which we will refer to as a gripper, rather than a hand). We will first describe the starting model, then discuss the reasons for choosing this particular design.

The basic gripper model is presented in Figure 6.5. A single tendon provides flexion forces for both fingers, which are co-actuated using a pulley mechanism, similar to the one used in the Harvard Hand. Note that the pulley allows one finger to continue flexing even if the other finger is blocked by contact with the object. Extension forces are provided by spring-like joints. In practice, these joints can be constructed using a compliant, rubber-like material; this design enables distal joints to flex even when

proximal joints are stopped, also providing mechanical adaptability. We assume that the kinematic behavior is that of ideal revolute joints, with the center of rotation placed halfway between the connected links.

The tendon itself follows a route in the flexion-extension plane of the gripper. This prevents the links from leaving this plane without the application of external forces, leading to an essentially two-dimensional design. However, the tendon route inside this plane is not specified, and is one of the targets of the optimizations.

Figure 6.5 also shows in detail the design parameters of the gripper. The tendon route is determined by the location of the entry and exit points for each link; more specifically, the parameter that we use is the distance between the tendon entry or exit point and the connection between the link and the joint. We also make the simplifying assumption that, for a joint i , the exit point from the proximal link and the entry point in the distal link have the same value for this parameter, which we call l_i . The current value of the joint is θ_i . r is the joint radius (shared by all the joints), while the length of the links is denoted by d .

The reason for using this design and formulation is that they yield a compact and, more importantly, linear relationship between the construction parameters and the joint forces applied through the tendon. If we consider the parameter vector $\mathbf{p} = [l_0 \ l_1 \ l_2 \ l_3 \ l_4 \ l_5 \ r \ d]$, we obtain a relationship of the form:

$$\boldsymbol{\tau} = \boldsymbol{\delta} (\mathbf{B}\mathbf{p} + \mathbf{a}) + \boldsymbol{\theta}\mathbf{k} \quad (6.13)$$

where the matrix $\mathbf{B} \in \mathcal{R}^{8 \times 8}$ and the vector $\mathbf{a} \in \mathcal{R}^8$ depend only on the joint values $\theta_0 \dots \theta_5$. A sketch for the derivation of these matrices is presented in Appendix B.

Furthermore, since we are using a single tendon, $\boldsymbol{\delta} \in \mathcal{R}$. Without loss of generality, we can normalize its value to $\delta = 1$. The joint force relationship, and by extension the grasp equilibrium conditions, are now fully linear, in all of the unknowns.

Having established the general characteristics of the gripper, the next step was to generate a pool of grasps over which to optimize its performance. We first created a kinematic model of the gripper for the *GraspIt!* environment, *assuming each joint*

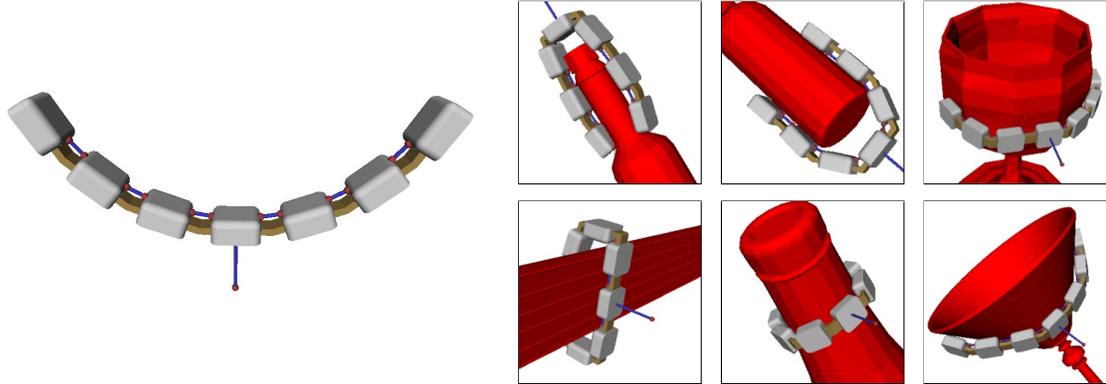


Figure 6.6: Left: gripper model for the *GraspIt!* environment. Right: examples of grasps from the optimization pool.

could be controlled independently. Then, using the interaction tools in the simulator, we manually specified a number of grasp postures over a set of 3D models of common household objects. The set comprised 70 grasps distributed across 15 objects; the process is illustrated in Figure 6.6. Each grasp was defined by the set of gripper joint angles, the location of the contacts on each link and the contact surface normals. We note that this is a purely "geometric" description of a grasp, with no reference to the actuation mechanism.

Most of the grasps in the pool used different postures for the two fingers of the gripper. We thus added to the set the "transpose" of each grasp, obtained by rotating the gripper by 180 degrees around the wrist roll axis, essentially reversing the roles of the left and right finger. The complete optimization pool thus comprised 140 grasps. The inclusion of both the original and the transposed grasps also ensured that the final optimized parameters, presented in the next section, were symmetrical, with identical results for both fingers.

A key restriction during the creation of the optimization pool was that all the grasps therein were required to have form-closure. *GraspIt!* integrates a number of analysis tools for establishing the form-closure property by building the Grasp Wrench Space, as described by Ferrari and Canny [1992]. This formulation is equivalent to

the ability of a set of contacts to apply a null resulting wrench on the object while satisfying contact friction constraints, but disregarding any kinematic or actuation constraints.

For each grasp in our optimization pool, we can apply the equilibrium formulation from Section 6.3, this time using the actuation mechanism modeled as described earlier in this section. The complete relationship is:

$$(\mathbf{J}_c^j)^T \mathbf{D}^j \boldsymbol{\beta}^j = \mathbf{B}^j \mathbf{p} + \boldsymbol{\alpha}^j + \boldsymbol{\theta}^j \mathbf{k} \quad (6.14)$$

$$\mathbf{G}^j \boldsymbol{\beta}^j = 0 \quad (6.15)$$

$$\boldsymbol{\beta}^j, \mathbf{F}^j \boldsymbol{\beta}^j \geq 0 \quad (6.16)$$

where we use the superscript j to denote the index number of the particular grasp from the optimization pool that we are referring to. The unknowns are the grasp contact forces $\boldsymbol{\beta}^j$, the hand parameter vector \mathbf{p} and the vector of joint spring stiffnesses \mathbf{k} . Note that \mathbf{p} and \mathbf{k} do not have a superscript as they are shared between all the grasps in the pool.

To obtain a global optimization problem, we assemble the above relationships in block form over the entire pool containing a total number of g grasps. The matrices for individual grasps $(\mathbf{J}_c^j)^T \mathbf{D}^j$, \mathbf{B}^j , $\boldsymbol{\theta}^j$, \mathbf{G}^j and \mathbf{F}^j are assembled in block diagonal form for $j = 1 \dots g$ in the matrices $\tilde{\mathbf{J}}_c^T \tilde{\mathbf{D}}$, $\tilde{\mathbf{B}}$, $\tilde{\boldsymbol{\theta}}$, $\tilde{\mathbf{G}}$ and $\tilde{\mathbf{F}}$, respectively. The vectors $\boldsymbol{\beta}^j$ and $\boldsymbol{\alpha}^j$ are assembled in block columns in the vectors $\tilde{\boldsymbol{\beta}}$ and $\tilde{\boldsymbol{\alpha}}$. Finally, the joint equilibrium condition (6.14) assembled for all the grasps in the pool becomes the optimization objective:

$$\text{minimize } \left\| \left[\tilde{\mathbf{J}}_c^T \tilde{\mathbf{D}} \quad -\tilde{\mathbf{B}} \quad -\tilde{\boldsymbol{\theta}} \right] \begin{bmatrix} \tilde{\boldsymbol{\beta}} \\ \mathbf{p} \\ \mathbf{k} \end{bmatrix} - \tilde{\boldsymbol{\alpha}} \right\| \text{ subject to:}$$

$$\tilde{\mathbf{G}}\tilde{\boldsymbol{\beta}} = 0 \quad (6.17)$$

$$\tilde{\boldsymbol{\beta}}, \tilde{\mathbf{F}}\tilde{\boldsymbol{\beta}} \geq 0 \quad (6.18)$$

$$\mathbf{p}_{min} \leq \mathbf{p} \leq \mathbf{p}_{max} \quad (6.19)$$

$$\mathbf{k}_{min} \leq \mathbf{k} \leq \mathbf{k}_{max} \quad (6.20)$$

The minimum and maximum values for the construction parameters \mathbf{p} and \mathbf{k} can be set to reflect constraints in the physical construction of the gripper, as we will show in the applied example in the next section.

We note that the result is again a convex Quadratic Program. Furthermore, the program is always feasible by construction: constraints (6.17) and (6.18) are equivalent to each individual grasp having form-closure independently of the actuation mechanism, which we ensured by building our grasp pool accordingly. As a result, the problem can always be solved and a global optimum can be computed.

6.4.3 Application: Construction of an Optimized Gripper

The final step of using our framework was physical construction of a gripper according to the results of the optimization. This required setting limits for the optimized parameters that could be implemented in practice. In particular, we used a limit of $-5 \leq l_i \leq 5 \forall i$ to ensure that the tendon route was inside the physical volume of each link.

The joint stiffness levels require additional discussion. The first thing to note is that the deciding factors for the behavior of the hand are the relative ratios of individual joint stiffnesses, not their absolute values. Indeed, increasing all stiffness values by a constant factor only scales all the forces in the system accordingly, without a qualitative change in the result. In practice, this would suggest using the lowest absolute values that yield the desired ratios, as this would have the effect of scaling down the level of unbalanced forces applied to the object. However, when using fast construction methods and inexpensive materials, very low, yet reliable stiffness values

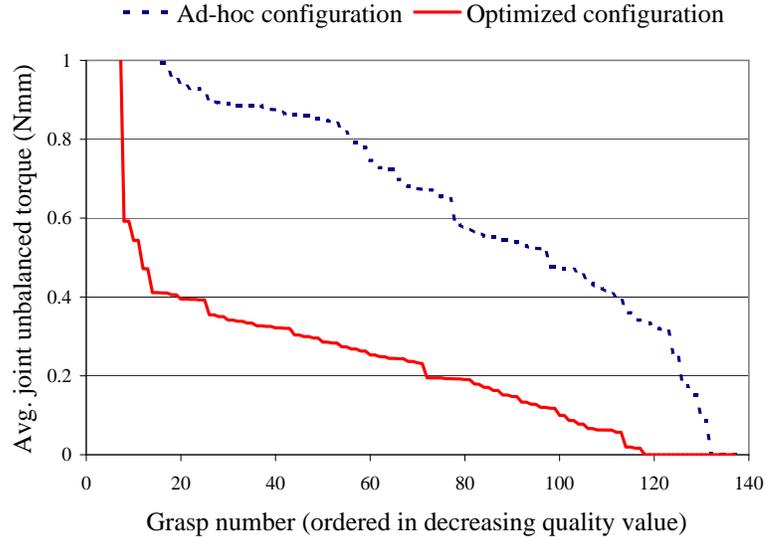


Figure 6.7: Comparison of unbalanced joint forces as a measure of grasp stability between optimized and ad-hoc configuration

Parameter	l_o	l_1	l_2	k_o	k_1	k_2
Optimized value	5.0	5.0	1.72	1.0	1.0	2.0

Table 6.2: Results of gripper design optimization

are hard to implement; so are very large relative ratios. In practice, we used as limits $1.0 \leq k_i \leq 2.0$. However, these limits can always be adapted based on the available materials and construction methods.

The results of the optimization are shown in Table 6.2. We only show the values for one of the fingers, since, as mentioned before, the results for the other finger are symmetrical. For a quantitative analysis of the computed optimal configuration, we compared it against a gripper configuration using an ad-hoc parameter set, with $l_i = 5$ and $k_i = 1 \forall i$. The comparison criterion was the level of unbalanced joint forces for each grasp. The results are shown in Figure 6.7. We notice that the optimized configuration provides significantly more stable grasps across the optimization pool. The total time spent formulating and solving the optimization problem was less

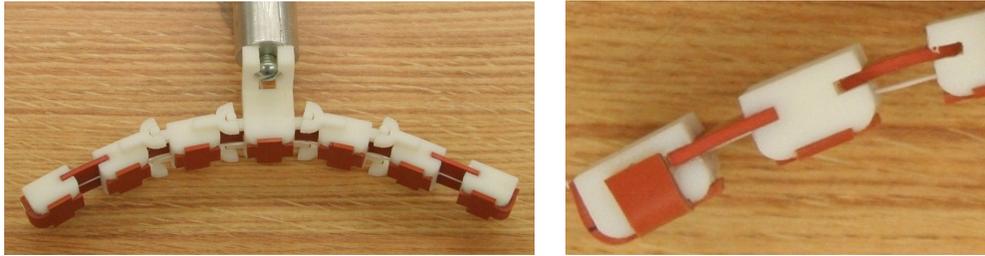


Figure 6.8: Prototype gripper constructed according to optimization results. Notice the different tendon route and rubber joint dimensions between the two distal joints.

than a minute, using a commodity desktop computer equipped with a 2.4GHz Intel Core2 Duo CPU. This suggests the future possibility of scaling to much larger grasp optimization pools.

We constructed a prototype gripper using the results of the optimization. The links were built using a Stratasys FDM rapid prototyping machine, and assembled using elastic joints cut from a sheet of hard rubber. Each link contained a tendon route with the entry and exit points set according to the optimization results. The width of the strip of rubber was varied for each joint to provide the specified stiffness ratios. For the tendon we used kite wire, which provided the desired combination of strength, flexibility and low friction. As this prototype is intended as a proof-of-concept for the kinematic configuration and design parameters, no motor or sensors were installed. Instead, actuation was performed manually. The final result is shown in Figure 6.8.

We found that the prototype gripper is capable of a wide range of grasping tasks and does not require precise positioning relative to the target object. Its passive adaptation ability is exemplified in Figure 6.9, which shows the execution of two grasps. The first one starts from a centered position and leads to relatively similar joint values for both fingers. In contrast, the second grasp requires the joints to conform to an asymmetrical, irregular shape. Both grasps were executed successfully.

Figure 6.10 attempts to provide an illustration of the spectrum of grasps that can be carried out with this gripper. All of the presented grasps were executed successfully

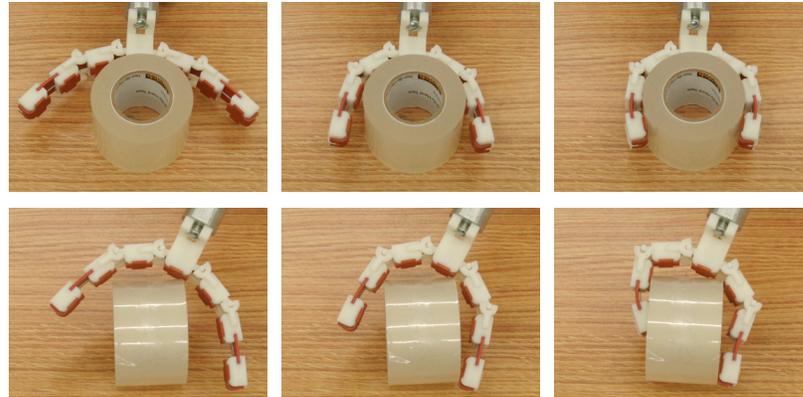


Figure 6.9: Two grasps executed with the prototype gripper. Top row: centered starting position and symmetrical grasp. Bottom row: off-centered starting position requiring passive adaptation to an irregular shape.



Figure 6.10: Examples of grasps successfully executed using the prototype gripper.

and the object was securely lifted off the table, with very little time or effort spent positioning the gripper relative to the target. In particular, we note that the gripper is capable of executing both fingertip grasps (of varying finger spans) and enveloping grasps (of both regular and irregular shapes).

Starting from this observation, we can attempt to perform a qualitative analysis of the optimization results. Intuitively, fingertip grasps require relatively low torques on the distal joints, so that fingertip forces are in opposition, rather than oriented towards the palm. Conversely, larger torques on the distal joints benefit enveloping grasps; as a result, the optimization process was required to combine two somewhat opposing goals. The results indicate that the solution does indeed enable both kinds of grasps, however the distal joint is both stiffer and less powerful than the proximal ones. It is interesting to note that, in this sense, the design comes close to a model with two links per finger.

We believe that this is precisely the type of analysis that our framework is natively well suited for. In future iterations, we can directly compare two- versus three-link models, and compute a numerical measure of the benefit provided by the additional link. The relatively simple two-fingered design that we used here allows an intuitive understanding of the design choices (which makes it well suited for initial testing and proof-of-concept implementations). However, for more complex models with multiple fingers, such qualitative analysis quickly becomes intractable, and quantitative tools, such as the one presented here, can prove extremely valuable.

6.5 Towards the Construction of an Eigenhand

In this chapter, we have focused on the problem of optimizing underactuated and passively adaptive robotic hands. For designs belonging to this class, the ability to apply forces to a grasped object is affected by co-actuation constraints. We have integrated these constraints in a quasistatic equilibrium formulation, also taking into

account contact friction models. Using this framework, we have identified two possible cases for framing hand design decisions as optimization problems. On one hand, adding a number of design constraints (as in the case of our gripper) enables the direct computation of a global optimum. On the other hand, for the more general problem, a number of non-linearities in the formulation prevent a similar solving strategy. Instead, more expensive, numerical algorithms can be employed.

We believe that future optimization studies will be a combination of both of these approaches. The space of possible hand designs, and implicitly the domain of parameters to be optimized, is practically limitless. Virtually any hand design ever proposed involves some compromise of ad-hoc decisions vs. informed, optimized parameter choices. In our case, we have discussed aspects such as tendon routes and joint stiffness. However, by moving up in the scale at which we are analyzing the hand, we can uncover many more design decisions, which we assumed as given: number and configuration of links, kinematic chains, *etc.* Some of these will likely prove impossible to encapsulate in a solvable optimization problem, thus some contribution from numerical approaches will be unavoidable.

From this standpoint, in order to make progress towards a fully optimized hand design, capable of reliable operation in a wide range of scenarios, we must focus on three key aspects:

- design more efficient algorithms to decrease the computational effort needed for analysis and optimization. As an applied example, consider the quasistatic optimization framework and the algorithms based on it that we have discussed here;
- take full advantage of the most powerful computing architectures for those computations that we can not (yet) avoid. For example, individual grasp analysis computations performed during the numerical optimization of Section 6.4.1 could be executed in parallel, taking advantage of multi-core CPU's;

- completely remove the human from the design optimization loop. An applied example involves the set of grasps used as an optimization pool for our gripper study: in our current implementation, this set was generated manually, which prevented complete automation of the optimization process. This can be replaced by an automated search procedure, running in a simulated environment, which can generate and analyze the pool of desired grasps.

Another interesting aspect concerns the on-line algorithms that are used to control the hand during grasping tasks. Traditionally, these algorithms have been designed after the hand was constructed, carefully tuned to extract the best performance from a given mechanical design. Off-line hand optimization enables the opposite approach: the hand mechanism is designed to suit a particular algorithm. The same holds true for sensor arrays: we can build a hand that is optimized for the types of grasps that we can perform based on data from a certain sensor. In this way, the hand is intrinsically equipped to handle the shortcomings of the input data. Overall, it seems natural to ask ourselves: what comes first, the hand or the algorithm?

We have started this chapter by discussing underactuated compliant hands as a natural complement to the eigengrasp concept. We have then presented a number of analysis tools, but did not come back to eigengrasps. In a sense, eigengrasps are a *top-down* view of low-dimensional hands: the subspaces that we can observe in the human hand, or those that we would like to use if the hand design afforded multiple choices. In contrast, under-actuated hands provide a *bottom-up* approach: the best subspaces that we can physically implement with as few actuators as possible. We believe that these are two ways of engaging the same problem, from opposite ends. At their meeting point, which has yet to be reached, we hope to find the next generation of robotic hands.

Chapter 7

Contact Models for Compliant Fingertips

Soft robotic fingertips can be thought of as another form of the passive adaptation phenomenon, which we have studied in the previous chapter at the kinematic chain level. Unlike adaptive kinematic chains however, compliant fingertips appear extremely easy to build and use. They do not require complex machinery, motors or transmission mechanisms. Instead, all that is needed is a layer of compliant material (such as rubber) applied on the surface of the fingertip. In use, they passively comply to the shape of the grasped object, giving rise to contact areas, rather than points. This phenomenon increases the space of forces and torques that can be transmitted between the two bodies in contact. As a result, soft fingers have often been used in practice, even for hands that do not display adaptation capabilities at other levels.

While the construction and application of compliant fingertips appear straightforward, the analysis of their behavior, along with a precise quantification of how they affect the grasping process, presents more difficulties. For example, we know that a fingertip coated in a given material will deform under contact, and support additional frictional wrenches. However, we would also like to compute contact properties such as the shape and pressure distribution of the contact area, the exact level of friction

that is supported, and their implications on the outcome of the grasp.

The analysis of compliant finger contacts is not specific to eigengrasp-based algorithms, or underactuated hands. However, it is as important to these particular sub-areas as it is to robotic grasping in general. By using soft finger contact models, all the analysis methods presented in this thesis can take advantage of the benefits provided by the use of compliant fingertip materials, while accounting for the subtle frictional effects that they give rise to. Our interactive grasping system from Chapter 5 used robot hands with rubber-coated fingertips; it was therefore necessary to integrate their effect in the planning algorithm. The underactuated gripper presented in Chapter 6 was also optimized taking into account the benefits of soft fingers. Both of these tasks required computationally efficient, yet realistic contact models.

7.1 The Soft Finger Model and its Linearized Version

In Section 2.1.2, we have reviewed a large body of work focusing on generating computational contact models. Armed with such a model, we can efficiently analyze the behavior of a contact, and design hands or grasping algorithms that take into account its nature. As we have already seen, the most common way of modeling a contact is by describing the space of wrenches that it can transmit. The result is known as the Contact Wrench Space, or CWS. Theoretically, this model can consist of any six-dimensional volume: any point inside this volume defines a wrench that can be applied through the contact.

In order to make this problem tractable, a common practice is to use analytical approximations for a contact model. We will use the contact reference frame that we defined earlier, with the z axis aligned with the contact normal and any tangential force decomposed in two components, along the x and y axes. For rigid bodies, the most commonly used model is Point Contact with Friction, which implements

Coulomb friction constraints:

$$f_x^2 + f_y^2 \leq \mu f_z^2 \quad (7.1)$$

Note that such a contact can only apply normal and tangential frictional forces, and no torques.

In the case of soft fingers, the contact occurs over some area that increases as the normal force increases. As a result, it is also possible to apply a frictional moment of magnitude τ_z about the contact normal. The constraint relating the magnitudes of frictional force and moment depends on the pressure distribution inside the contact, and can only be derived explicitly for a limited number of special cases. However, Howe and Cutkosky [1996] have shown that, for the general case, we can use an approximation of the following analytical form:

$$f_x^2 + f_y^2 + \frac{\tau_z^2}{e_z^2} \leq \mu f_z^2 \quad (7.2)$$

This model is characterized by two parameters: the friction coefficient μ and the eccentricity parameter e_z . Their values describe what is commonly referred to as the friction ellipsoid: according to this model, for a normal force f_z of unit magnitude, the frictional component $[f_x \ f_y \ \tau_z]$ of the contact wrench is constrained to lie inside a three-dimensional ellipsoid, of radius μ and height e_z . The value of e_z can be obtained experimentally as shown by Howe and Cutkosky [1996], but the result is only accurate for the particular combination of normal force, object geometry and material parameters used in the experiment. In the following sections we will present two methods for computing an appropriate value for this parameter; in addition we will discuss the possibility of adding contact wrenches that are not captured by the soft finger model.

We also note that, in their exact formulation, both of these models are difficult to use in practice due to the quadratic nature of the constraints. A common solution is to use a linearized version, where the frictional component of the contact wrench is expressed as a linear combination of a finite number of reference wrenches, that sample the boundary of the CWS. The analytical boundary of the CWS is thus replaced by the

convex hull of the sample wrenches. This is also the approach that we have used in the previous chapter to integrate contact constraints into the quasistatic grasp equilibrium formulation. More details about constructing and using linearized versions of the contact models presented here were presented, among others, by Anitescu and Potra [1997], Miller and Christensen [2003], Prattichizzo and Trinkle [2008], *etc.*

7.2 Computation using Finite Element Analysis

The first method that we propose for building a soft finger friction ellipsoid relies on Finite Element Analysis (FEA) to compute the fingertip response to contact. The finger is modeled as a 3D mesh, comprising a number of vertices and 3D elements. FEA is used to compute the deformation of a fingertip of arbitrary geometry in contact with a planar rigid surface (thus making the assumption that the grasped object is locally planar). For a given value of *total normal force* applied at the contact, we used FEA to compute the contact forces applied *at each vertex* in order to prevent interpenetration. We also compute the deformation of the fingerpad (the displacement of each vertex) in response to contact.

A key aspect of the FEA-based simulation is that it can take into account point-wise frictional forces applied at each vertex. If a direction of relative motion at the contact is specified, frictional forces that result during sliding can be computed, as well as the deformation of the mesh due to those forces. Specific details regarding the derivation and implementation of the finite element method we have used can be found in our previous study [Ciocarlie et al., 2005].

Using the FEA simulation we can compute the total frictional force and moment applied at the contact for any relative motion between the bodies in contact. Theoretically, the contact model could be built in its entirety by computing such force-moment combinations for any possible direction of relative contact motion. In order to avoid this computation, we approximate it using the soft finger contact model with an ap-

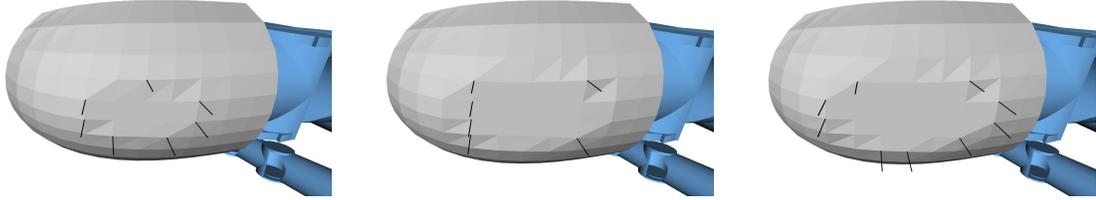


Figure 7.1: View of the contact area for normal forces of 5N (left), 20N (middle) and 25N (right) from within a transparent cube.

appropriate value of the eccentricity parameter. For a given value of f_z , we compute the value of the frictional torque τ_z applied when contact motion is a *rotation* around the *pressure-weighted center of contact*. According to Howe and Cutkosky [1996], this equals $\max(\tau_z)$, the maximum amount of friction torque that the contact can sustain. According to equation (7.2) we recover the eccentricity parameter as

$$e_z = \frac{\max(\tau_z)}{f_z} \quad (7.3)$$

After building the soft finger contact model we further augment it by considering moments that lie within the tangent plane of the contact. Traditional approaches assume that the contact force is concentrated into a single point, and can not apply any such moments. However, the FEA approach enables us to compute the shape of the contact area, as seen in figure 7.1. We therefore add to the model the range of moments obtained by considering the total contact normal force applied at any of the vertices comprised in the contact area. We note that moments within the tangent plane will generally deform the finger and allow some motion at the contact, requiring a re-computation of the contact area shape for an exact analysis of the fingertip response. However, we feel that our approach provides a conservative approximation for the space of moments in the tangent plane that the contact can apply.

One of the major drawbacks of using FEA is the computational effort that it requires. In general, building a single contact model takes between 1 and 3 minutes on a standard desktop computer. However, the method also has a number of advantages.

Most importantly, while we assume that the rigid object is locally planar, we make *no assumptions* regarding the geometry of the fingerpad. As such, this method is suited for applications that use accurate contact wrench spaces for complex or layered fingertip geometry, but do not require interactive computation speed. In the next section we will present an alternative to this approach, making different assumptions on the nature of the contact and requiring no computationally expensive steps during the model building process.

7.3 Computation using Analytical Surface Approximations

For applications requiring fast computation of the contact model, traditional approaches use analytical approximations not only for the contact wrench space but also for the geometry of one, or both of the objects in contact. For example, the finger is often modeled as a hemisphere [Barbagli et al., 2004], while the grasped object is assumed to be locally planar. Tada and Pai [2008] use a fingertip model that integrates geometry information specific to the subject, and use it to study finger deformation under line loads. In this section we present a different approach, using analytical surface models for both the finger and the external object, in order to account for contacts between general, non-planar surfaces.

We propose approximating the objects at the point of contact as smooth surfaces characterized by their *principal radii of curvature*. We apply the formulation of Johnson [1985], using the same contact coordinate system as before, with the origin at the center of the contact and the z axis aligned with the contact normal. For two contacting bodies identified by the subscript i , we locally approximate their surfaces using an expression of the form

$$z_i = A_i x^2 + B_i y^2 + C_i xy, \quad i \in \{1, 2\} \quad (7.4)$$

making the assumptions that the objects are locally smooth. The separation h between the two surfaces is:

$$h = z_1 - z_2 = (A_1 - A_2)x^2 + (B_1 - B_2)y^2 + (C_1 - C_2)xy \quad (7.5)$$

By choosing the orientation of the x and y axes so that the term in xy vanishes, equation (7.5) may be re-written as:

$$h = \frac{1}{2R'}x^2 + \frac{1}{2R''}y^2 \quad (7.6)$$

where R' and R'' are the *relative radii of curvature* of the objects in contact, depending only on their local geometry.

The relative radii of curvature R' and R'' form a simple yet flexible description of contact geometry. This form encompasses the simple case of a hemisphere touching a plane, but also allows for a good approximation for a much larger range of fingertip and object shapes. Our simulation environment considers objects as three-dimensional meshes, which enables the use of numerous existing triangle mesh models, as well as efficient collision detection algorithms. When initial contact between two objects is detected, we approximate the shape of each object in a small region around the contact, using an analytical surface as described above. Since the resulting surface is expected to fit the original mesh only in a small area close to the contact region, we can obtain very close approximations using the relatively simple surface form of equation (7.4).

Figure 7.2 exemplifies the result of our fitting method on robot fingers as well as the grasped object. We have used the fingertip model of the Shadow Robot [Shadow Hand] anthropomorphic hand, currently in commercial production. The quality of the approximation was computed as the standard deviation of the distance between the fitted surface and the original mesh, measured over a local region twice as large as the contact area predicted by the contact model. Considering all the fitting results shown in figure 7.2, the largest value was reported for the thumb contact, with a standard deviation of 0.94mm over an area of 82mm².

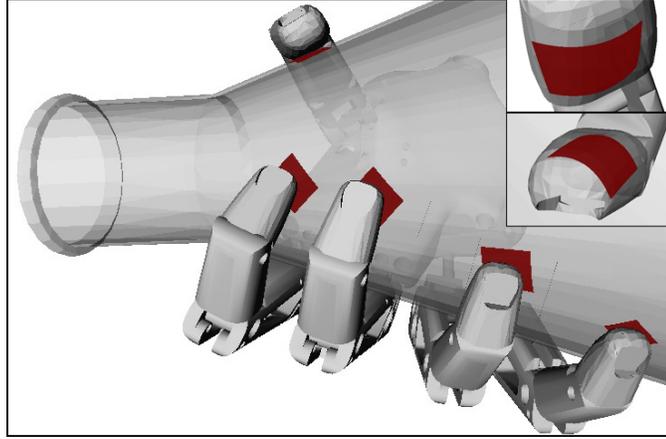


Figure 7.2: Analytical surfaces approximating the local geometry of a grasped object. Magnifications show local approximations for the thumb and little finger.

Having computed the relative radii of curvature R' and R'' , we can express the pressure distribution inside the contact using non-planar models that take into account the local geometry of the objects involved. We have performed this analysis for two pressure distribution models. The first is the Hertzian model, previously used in the literature to simulate both human and robotic soft fingertips [Barbagli et al., 2004, Brock, 1988]. The second is the Winkler elastic foundation model, which can be used in the case of an elastic layer resting on a rigid base, such as a robotic fingertip coated in a thin layer of soft material. It is important to note that these models share a number of assumptions regarding the bodies in contact: linear relationship between stress and strain, homogeneous inner structure of the objects and small deformations due to contact (relative to the total size of the objects).

For both of these models, the contact area is known to be elliptical in shape [Johnson, 1985], with semi-axes a and b depending, among other factors, on the relative radii of curvature at the contact. In the case of the Winkler foundation model, the ratio of frictional torque to contact load can be computed as:

$$\frac{\max(\tau_z)}{P} = \frac{8}{15}\mu\sqrt{ab} \quad (7.7)$$

Alternatively, in the case of a Hertzian pressure model the relationship is:

$$\frac{\max(\tau_z)}{P} = \frac{3\pi}{16}\mu\sqrt{ab} \quad (7.8)$$

The derivation of equation (7.7), applied in the case of the Winkler foundation model, is sketched in Appendix C. For more details, we refer the reader to the analysis presented by Johnson [1985]. The case of the Hertzian model can be handled in a similar manner, based on the results of Johnson [1985] and Barbagli et al. [2004]; the Boussinesq approximation could also be used, following the example of Pauly et al. [2004].

As described in the previous section, the ratio of maximum frictional torque to total contact load can then be used to compute an appropriate eccentricity parameter e_z for a soft finger model using equation (7.3). We note that this method can also be applied for other pressure distribution models, as long as the value of the maximum frictional moment can be derived from the contact force and the local geometry of the colliding objects.

We can now summarize the algorithm used for setting up soft contacts. Starting when initial contact between two bodies is detected, we perform the following steps:

- use a least squares method to fit a surface of the form of equation (7.4) to each of the bodies involved;
- compute the relative radii of curvature at the contact;
- choose a model for the pressure distribution inside the contact that best fits the objects involved;
- use the pressure distribution model and the relative radii of curvature, *i.e.* equations (7.7) and (7.8), to compute the dependency between contact normal force and maximum frictional torque;
- build a soft finger model using an eccentricity parameter computed as in equation (7.3).

7.4 Applications

We have integrated both approaches for building Soft Finger Models into the *GraspIt!* simulation and analysis engine. The FEA-based method can be used in conjunction with *GraspIt!*'s quality metric computation tool, which uses the total Grasp Wrench Space (GWS) built from individual wrench spaces of each contact. While this approach can potentially be applied to perform grasp analysis for complex fingertip structures, its computational requirements prevented further integration with the tools and algorithms presented in this thesis. For complete details regarding this method, as well as examples of quality values obtained for soft finger grasps, we refer the reader to our separate study [Ciocarlie et al., 2005]. We note that the FEA-based soft finger model enables the study of grasps that would otherwise be classified incorrectly, such as two-fingered pinch grasps which rely on both frictional force and torque applied in the contact tangent plane for stability.

On the other hand, the main advantage of the second contact model we have introduced is its computational efficiency. It can be explained by the fact that the algorithm uses both an analytical wrench space model and analytical expressions for the geometry of the contact. The model is adapted to the local geometry of the objects during the least squares fit (first step in the algorithm) which is a very fast procedure. The analytical form that is used to model object surfaces, relying on principal radii of curvature, is only an approximation of the real fingertip; however, it allows greater flexibility than point contact models, or soft finger models with a fixed relationship between frictional forces and torques.

In this thesis, we have already presented two applications of this model. The online grasp planning algorithm of Chapter 5 uses soft finger wrench spaces to assess the quality of a pre-grasp (Section 5.3.2). Our fast model construction method enables it to capture the frictional effects of soft fingers while still analyzing approximately 1000 hand postures per second. In turn, this allowed us to use a hand equipped with rubber-coated finger in the experiments presented in Section 5.4, increasing our

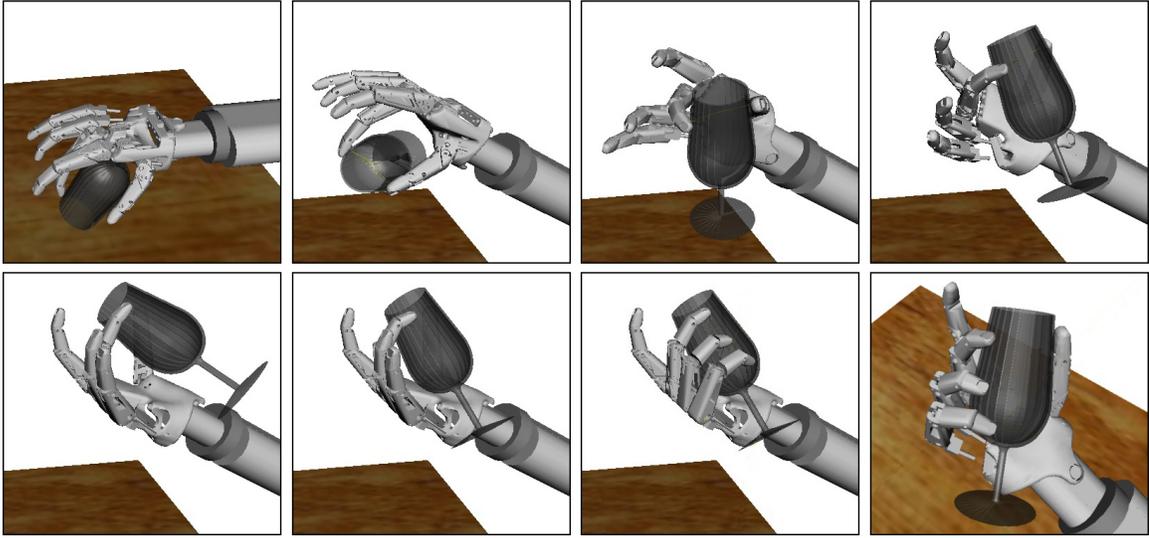


Figure 7.3: Simulation of a manipulation task using an anthropomorphic robotic hand. Top row: object is picked up and rotated above the palm. Bottom row: by controlling fingertip force, rotational sliding is allowed at the contacts until object rests on palm. By closing the fingers a stable grasp is obtained.

success rate. We have also integrated this model into our optimization framework for compliant hands presented in Chapter 6, by using the linearized version of the soft finger friction ellipsoid to set up the contact constraints from Section 6.2.1. Both the Harvard Hand and the single-tendon gripper that we analyze in Section 6.4.3 use soft fingers to help achieve their passive compliance ability.

The computational efficiency of the analytical soft finger model also makes it suitable for interactive dynamic simulations. We have integrated this model with the dynamic engine of the *GraspIt!* simulator, which uses linearized contact constraints framed as a Linear Complementarity Problem [Miller and Christensen, 2003]. An example of a complex manipulation task, requiring the use of compliant fingertips, simulated using the *GraspIt!* engine, is presented in Figure 7.3. The goal in this example is to obtain an enveloping grasp of the glass by using finger contacts around its circumference. However, in the initial position, the surface of the table prevents

the execution of such a grasp. We have simulated a control algorithm that uses two fingers to pick up the object and rotate it above the palm. This is possible only in the presence of frictional torque applied by soft finger contacts. With the palm facing up, the force applied by the fingers is decreased, allowing the glass to rotate while maintaining contact with the fingertips until it hits the rigid palm of the robot. The ring and little finger can now be closed around the glass, creating a stable grasp. For the sequence shown in figure 7.3, the simulation was carried out at a rate of 10.7 time steps per second on a standard desktop computer, with the entire sequence consisting of 1800 time steps.

Finally, a promising application for grasp analysis methods originally developed for robotic hands is the study of the human hand itself. However, this can only be achieved by using realistic computational models of the complex features of the human hand, including compliant fingertips. As an example, we have used our soft finger model, together with the *GraspIt!* analysis tools based on the Grasp Wrench Space, to study whether human choice of contact locations during grasping and manipulation can partly be explained by understanding the nature of the task and the expected disturbances; the results that we obtained are presented and discussed in a separate study [Ciocarlie et al., 2009]. While robotic hands are still a long way from matching the capabilities of their human counterparts, we believe that computer modeling and simulation may help us understand this disparity and, in the long run, also enable new insights into human grasping and manipulation.

Chapter 8

Conclusions

With the chapter dedicated to soft finger models, we have now completed the presentation of the new approaches, algorithms and tools that this thesis is dedicated to. The topics that we have covered include low-dimensional grasp planning for dexterous and anthropomorphic hands, interactive execution of grasping tasks using input from a human or non-human primate operator, optimization of passively adaptive transmission mechanisms for underactuated hands, and efficient computational models for soft finger contacts. It is time to take a step back and recall the common motivation behind all of these research directions, then point out some of their current limitations and opportunities for future improvements.

The principal motivation for the work described in this thesis is rooted in the following open research problem. Even though robotic hand design has produced a number of well-engineered, advanced hand models, with many anthropomorphic characteristics, and presumably able to match many of the abilities of the human hand, the field is currently without a proven solution for performing object acquisition (grasping) of a satisfactorily wide range of objects, in unstructured environments. Human-like robotic hands, equipped with many degrees of freedom, require equally complex grasping algorithms to handle the high dimensionality of the posture space. Simple, more intuitive hand designs, such as parallel grippers, are easier to manufac-

ture and use, but lack the versatility needed in typical human environments. Finally, until recently, few hand models have been proposed in the design space between these two extremes.

A newly emerging field, which shares many of the challenges in designing effective artificial hands, is that of hand neuroprosthetics. A neurally controlled prostheses is designed to interact in real time with a human operator. Unlike a fully autonomous robot, such a system does not need to integrate tools for elaborate semantic perception, environment modeling or high-level task planning. On the other hand, it is the interaction paradigm itself that has proven challenging to implement in practice: current results in recording and interpreting neural information from the motor cortex have shown success in decoding a relatively small number of channels of information.

At the intersection of these problems, we identify the need to reduce the complexity of robotic hands, without compromising their versatility. By achieving this reduction in complexity, we can potentially uncover a new control space for fully dexterous hands, a design space for less complex, yet highly reliable underactuated models, and an interaction space for the next generation of hand prostheses. While we have not yet reached this level of performance, it is our directional goal, and we believe that the algorithms and tools presented in this thesis are a number of steps along this path.

8.1 Thesis Summary

The starting point for our approach involves using a low-dimensional subspace of a dexterous hand's DOF space for finding postures appropriate for a given task. Linear dimensionality reduction techniques, applied to a large set of grasp posture data obtained from human user studies, have shown that such a linear subspace, spanning only two dimensions, encapsulates most of the posture variance. We have introduced the concept of *eigengrasps* as the set of basis that define this subspace.

An eigengrasp defines a linear relationship between multiple joints of a hand, in a way that is relevant for performing grasping tasks. This concept has multiple applications: when discussing human grasping, such a relationship is often called a "joint synergy", for underactuated hands, it can become a "co-actuation constraint", *etc.*

As long as the eigengrasp space provides a good approximation of the hand postures required for a given task, algorithms can be designed to operate in this space and take advantage of the dimensionality reduction. We have introduced a low-dimensional hand posture optimization method applied for stable grasp synthesis. Our algorithm uses a stochastic optimization approach (simulated annealing) to find the hand posture that maximizes a potential grasp quality function. The posture optimization domain includes only eigengrasp amplitudes, as opposed to individual joint values, greatly reducing the dimensionality of the search. The results show that, while not containing exact grasping postures, a low dimensional eigengrasp space can serve as an effective pre-grasp or planning space even for highly dexterous hand models.

Turning to the case of hand prosthetics, we proposed using a grasp planning algorithm as an interface between incomplete or noisy operator input and the full specification of hand posture and position needed for dexterous grasping. In order to achieve the computational rates needed for operator interaction, we again turn to an eigengrasp based synthesis algorithm. Grasp planning running at an interactive rate enables the use of operator input during the search, which in turn helps speed up the optimization. To this end, we extend our algorithm to define a target value, based on operator input, along any dimension of the search. The set of target values is then used to bias the automated synthesis towards the behavior specified by the operator. While the grasp planning process takes place in a simulated environment, which enables it to analyze thousands of postures per second, the results can be applied in the real world, enabling the interactive execution of real-life grasping tasks.

A grasp synthesis method operating in eigengrasp space is based on the idea

that effective pre-grasp postures can be achieved by searching along relatively few dimensions. An open-loop finger closing process then provides exact adaptation to the shape of the target object. The applications discussed so far are computational in nature, aiming to reduce the complexity of using dexterous or anthropomorphic robotic or prosthetic hands. Naturally, a hand that enables individual control for each DOF provides an ideal testing ground for such an approach, allowing any eigengrasp joint relationship to be implemented in practice. Furthermore, a larger number of hand DOFs enable better adaptation to the target object during the final stage. However, the ability to control each DOF comes at a high cost, in terms of mechanical complexity and production costs. If the goal that we focus on is stable grasping ability, the eigengrasp-based approach raises the question of whether we can achieve comparable performance without the traditional costs associated with fully actuated dexterous hands.

The alternative that we discuss in this thesis taps into recent advances in the complementary areas of underactuated and passively adaptive hands. These two concepts mirror the computational approach that we have already discussed: an underactuated hand can be thought of as the hardware equivalent of an eigengrasp algorithm, while passive adaptation allows the object itself to implicitly fine-tune the grasp without explicit computation. However, they also require further analysis of the hand design: the choice of an eigengrasp subspace is affected by the transmission mechanism used to implement it in practice, and by passive compliance constraints.

Along these lines, we have presented a method for analysis and optimization of tendon-based underactuated adaptive hands. We have integrated the co-actuation and compliance constraints, together with contact friction constraints, into a quasistatic equilibrium formulation. Using this model, we can build a solvable optimization problem to compute the hand design parameters that provide the best performance over a large set of grasping tasks. We believe that, for the class of adaptive underactuated hands, the on-line grasp planning effort, traditionally carried out un-

der tight time constraints and requiring extensive sensing capabilities, can be replaced by off-line optimization increasing hand performance over many grasping scenarios. As a concrete example, we have analyzed a simplified single-tendon gripper model, and have used the results of the optimization to construct a prototype capable of a wide range of grasps.

Finally, many of the grasp synthesis and hand design tools that we have proposed in this thesis rely on the use of soft fingertips. The understanding of touch is paramount to grasp analysis, as shown by the advantages afforded by compliant fingertips during grasping: by making contact over an area, as opposed to a single point, soft fingers are able to sustain a range of contact torques that are not supported by their rigid counterparts. Along these lines, we have introduced a fast method for constructing an analytical model of a soft finger Contact Wrench Space. We used analytical approximations for the local geometry of the bodies in contact based on their relative radii of curvature at the contact. This component is an intrinsic part for both the design of passively compliant hands and the on-line application of grasp analysis results when using any hand model equipped with soft fingertips.

8.2 Lessons Learned

We have applied the methods above in a variety of scenarios, in both simulation and real environments. Detailed discussions for each case are presented in the respective chapters of the thesis; here we summarize some of the most representative results, as well as the implications that we derive from them. We believe that these lessons can prove useful moving forward towards new artificial hands and grasping algorithms. They include the following:

- for a wide spectrum of hand models, with the number of DOFs ranging from 4 to 20, hand posture can be optimized inside a 2-dimensional eigengrasp space to provide pre-grasp postures with a high chance of resulting in stable grasps.

By using the eigengrasp space, the dimensionality of the posture optimization domain is reduced by as much as a factor of 10 (in the case of the most dexterous model).

- the refinement of a pre-grasp shape into a stable grasp can be performed using an open-loop process where the fingers are closed until contact with the object prevents further motion. During this step, in which the hand leaves eigengrasp space, the shape of the object implicitly determined the final adjustments, without any explicit computation.
- the complete grasp synthesis algorithm can be used to find form-closure grasps on all the hand models in the test set. No adaptation to a particular hand model is needed beyond the definition of the hand-specific eigengrasp subspace.
- by using an eigengrasp subspace to optimize hand posture, along with operator input for hand position and orientation, grasp planning can be performed at the computational rates needed for on-line interaction with the operator.
- a shared control paradigm, where operator input is complemented by an automated grasp planner operating in eigengrasp space, can enable the operator to perform grasping tasks using an artificial hand even if the communication between the biological and automated components is incomplete and/or noisy.
- moving to the analysis of underactuated hands based on tendons and compliant joints, the quasistatic formulation that we propose can be used to analyze hundreds of postures in approximately one minute, separating the stable and unstable grasps. When applying this tool to a particular hand design, a one hour analysis carried out over thousands of possible grasps can identify a subset of the hand construction parameters that provide the largest number of stable grasps.

- alternatively, for a class of two-fingered, single-tendon grippers, we can formulate hand design choices as a solvable optimization problem, reducing the computational effort to less than a minute. The prototype built according to the results of this optimization is able to use a wide range of grasps, including fingertip and enveloping grasps.

8.3 Directions for Future Research

A low-dimensional eigengrasp space lies at the core of many of the methods presented in this thesis. As such, one of the most clear paths for further improvement of the algorithms is to focus on the derivation of the subspace itself. We have based our choices on synergies observed through numerical analysis of human grasping data, and have translated this results empirically to a number of robot hands. An important question that we would like to address in the future is the computation of an optimal eigengrasp subspace for a given robotic hand model and task. One possible approach is to perform a dense sampling of the high-dimensional control space of the robotic hand, then find the low-dimensional decomposition that contains most of the desirable hand postures. The sampling process for the hand configuration space can be performed off-line, therefore computational restrictions can be somewhat relaxed. However, in the case of very complex hands with 20 or more intrinsic DOFs, this task is intractable even with an off-line assumption, and such cases will require further study.

An intriguing alternative is to perform this analysis not in the space of hand kinematics, but rather in the space of grasped object geometry. If we can identify a subspace of objects, or object features, that a given hand must be able to grasp, we can use it as a goal for hand co-actuation schemes. Furthermore, since object space is independent of hand kinematics, it could potentially enable the translation of one hand's eigengrasp subspace to another kinematic model. This is also a key problem for the design of hand prostheses, where low-dimensional input from the operator

must be translated to the eigengrasp space of the prosthetic.

Our eigengrasp planner can be applied to a wide range of hands, objects and grasping task scenarios. However, run-time application during task execution requires extensive sensing capabilities to provide information on the grasped object. An alternative is to use the same algorithm strictly inside a simulated environment, in order to generate large amounts of labeled grasp information using a given hand model. Very large amounts of data, generated off-line, can help us learn about the intrinsic nature of the grasping task, and reduce the need for on-line sensing information at execution time.

A number of steps have already been taken in this direction: in recent work, the eigengrasp planner presented here was used as the data generation component of a complete data-driven pipeline. The result of this process was the Columbia Grasp Database introduced by Goldfeder et al. [2009a], a collection of labeled grasp information that is larger than similar datasets by several orders of magnitude. This dataset provides the input for a database-backed grasp planning algorithm [Goldfeder et al., 2009b] developed in parallel with the work described in this thesis, which was shown to be effective even when working with incomplete sensor data of the grasped object.

An eigengrasp-based planning algorithm operating directly on sensor data (rather than complete object models) is also a promising direction. The most commonly used sensing methods that we rely on (laser scanners, monocular and stereo vision, *etc.*) share a number of limitations, such as occlusion and noise. As such, they usually provide a representation of the object that is deprived of fine detail, just as a low-dimensional representation of the hand posture space misses the high frequencies. Eigengrasp planning in the space on sensor data seems natural in this context. The gap between the low-dimensional hand / sensed data representation and the final grasp execution can be filled through passive adaptation. We envision a comprehensive hand design and optimization approach, taking into account the eigengrasp

subspace, the planning algorithm and the nature of sensor data that will be used at run-time.

8.4 Robotic Grasping: The Road Ahead

When will we see robots capable of grasping with human-like confidence in unstructured environments? The robotics community at large is investing increasing resources in finding an answer to this question. Based on the body of work presented here, we can only extrapolate what we believe to be some of the key principles that will take us there.

The next generation of robotic hands will be designed by gradually eliminating decisions made ad-hoc, or based only on resemblance with the human hand, in favor of informed choices based on analysis and optimization. Both mechanical and algorithmic complexity should be increased based strictly on provable performance gains, and also reflect the level on meaningful sensed data that we can rely on. In this sense, the nature of the sensors that we can use must inform the design decisions, rather than becoming a liability at run-time. If we can achieve all these results, we will have realized the hand design equivalent of a constructive proof in mathematics: to show that a hand exists that *can* perform a given set of grasping tasks, and in the process to also show *how* to perform these tasks, in a repeatable and robust manner.

Robotic grasping is much more than just hand design and operation. It also requires close interoperation with components such as semantic perception, motion planning, high-level task planning, *etc.* In this sense, the field of hand prosthetics seems poised to take advantage of the human operator's knowledge and input and make the first quantum leap. However, this is predicated on the ability to provide an effective interaction method between the operator and the artificial hand. For fully autonomous robots, the ability to interact with, and effect change on the environment through grasping and manipulation, must become one component of a

complete platform for which other major components are already in place. These are highly challenging tasks, but the goal is equally compelling: to deliver robotic applications with very high social impact, applications that have long been envisioned in popular culture, and that we are getting closer to every day.

Appendix A

Biased Neighbor Generating Functions for Simulated Annealing

A number of neighbor generating functions have been proposed in the simulated annealing literature; in our study we have used the version presented by Ingber [1989], which allows for faster exploration of the input domain. Its exact form is:

$$y(u) = \text{sgn}(u)T(k) \left[\left(1 + \frac{1}{T(k)} \right)^{|u|} - 1 \right] \quad (\text{A.1})$$

where T_k is the annealing temperature at step k , decreasing as the annealing algorithm progresses. If the generating variable u follows a uniform distribution, the density function of its probability distribution can be computed as

$$PDF_y(x) = \frac{1}{2(|x| + T(k)) \ln(1 + 1/T(k))} \quad (\text{A.2})$$

In Figure A.1, the solid blue lines show the probability density function during early, middle and late stages of the annealing algorithm. As discussed, this neighbor generation method favors small neighborhoods of the current solution, especially for low temperatures during the late stages of the annealing schedule.

When a target value u_t is given, the generating variable u obeys a normal distribution of mean u_t and variance $1 - \sigma$. This changes the probability distribution of

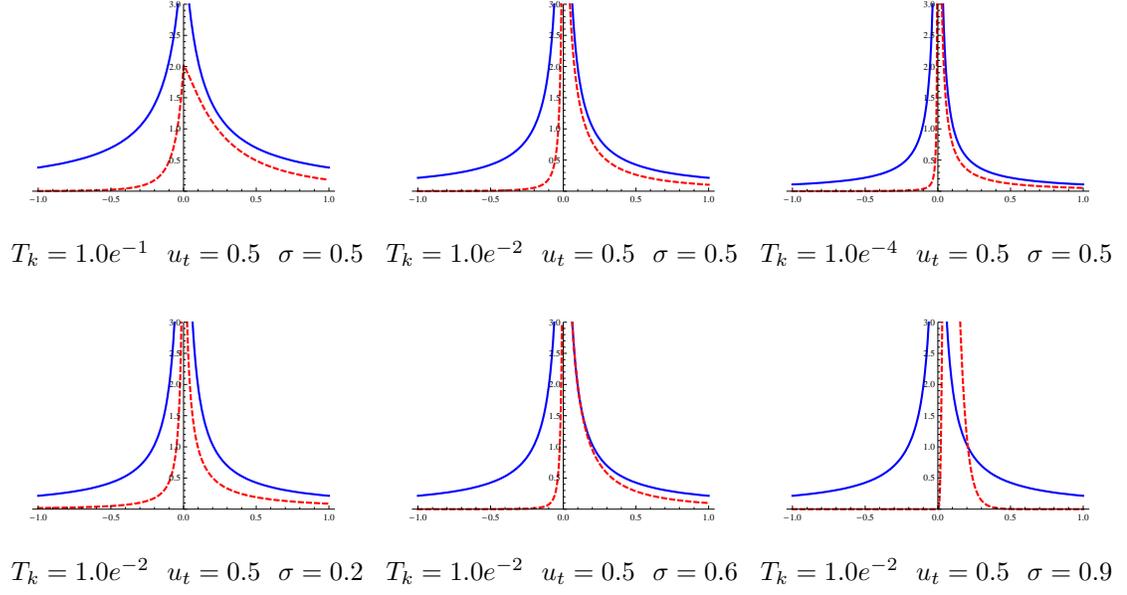


Figure A.1: Comparison of probability density functions for different annealing temperatures T_k . Solid blue line shows PDF of neighbor generation function without external input; dashed red line shows PDF of neighbor generation function when a jump with target u_t and confidence σ is specified.

$y(u)$, and the new density function can be computed as

$$PDF_{input}(x) = n[y^{-1}(x)] PDF_y(x) \quad (\text{A.3})$$

where $n(x)$ is the probability density function of a normal distribution of mean u_t and variance $1 - \sigma$.

The dashed red lines in Figure A.1 show the changed profile of the neighbor distribution for particular values of the target jump u_t and confidence levels σ . We notice that the presence of the input biases the neighbor generation towards the target value, without affecting its main characteristics.

Appendix B

Derivation of A Linear Tendon Route Model

In order to sketch the derivation for the relationship between the tendon route parameters and the resulting joint torques, we start by focusing on how tendon entry and exit points on link i affect the torque applied at joint j . Using the notation shown in Figure B.1, we use joint i as our reference coordinate frame, and assume that the translation from joint j to joint i is $\mathbf{t}^{ij} = [t_x^{ij} \ t_y^{ij}]^T$.

In general, for any point where a tendon changes direction, such as the link entry point in the figure, the force applied to the link is the resultant of the total tendon force applied in both the initial and the changed direction, or $\mathbf{f} = \mathbf{f}_{in} + \mathbf{f}_{out}$. We

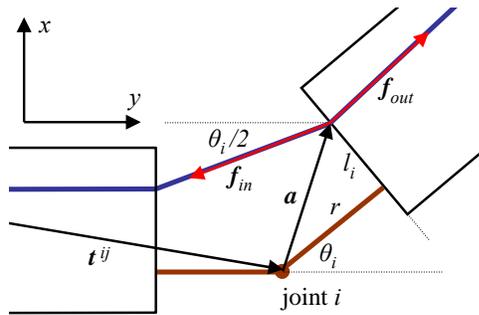


Figure B.1: Torque computation for tendon entry point

note that $\|\mathbf{f}_{in}\| = \|\mathbf{f}_{out}\| = \delta$. However, since we normalize tendon force to $\delta = 1$ we can omit it from the computations. We then obtain the torque applied around a given joint by cross-product with the joint moment arm. Using this notation, the torque around joint j applied at the tendon entry point in link i is:

$$\tau_{\text{entry}}^{ij} = (\mathbf{t}^{ij} + \mathbf{a}) \times (\mathbf{f}_{in} + \mathbf{f}_{out}) \quad (\text{B.1})$$

$$\begin{aligned} &= \left(\begin{bmatrix} t_x^{ij} \\ t_y^{ij} \end{bmatrix} + \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix} \begin{bmatrix} l_i \\ r \end{bmatrix} \right) \times \\ &\quad \times \left(\begin{bmatrix} -\sin(\theta_i/2) \\ -\cos(\theta_i/2) \end{bmatrix} + \begin{bmatrix} \sin\theta_i \\ \cos\theta_i \end{bmatrix} \right) \end{aligned} \quad (\text{B.2})$$

Through a similar computation, using the notation from Figures 6.5 and B.1, we can compute the torque applied at the tendon exit point from link i as:

$$\begin{aligned} \tau_{\text{exit}}^{ij} &= \left(\begin{bmatrix} t_x^{ij} \\ t_y^{ij} \end{bmatrix} + \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix} \begin{bmatrix} l_{i+1} \\ r + d \end{bmatrix} \right) \times \\ &\quad \times \left(\begin{bmatrix} \sin(\theta_i + \theta_{i+1}/2) \\ \cos(\theta_i + \theta_{i+1}/2) \end{bmatrix} + \begin{bmatrix} -\sin\theta_i \\ -\cos\theta_i \end{bmatrix} \right) \end{aligned} \quad (\text{B.3})$$

If $l_i \neq l_{i+1}$, the tendon must also change direction somewhere inside link i . The resulting torque is simply:

$$\tau_{\text{change}}^{ij} = l_{i+1} - l_i \quad (\text{B.4})$$

All of these contributions are added to obtain the total torque applied on joint j due to tendon routing points on link i . Finally, the computation above is repeated for all desired combinations of i and j . By explicitly computing cross products as $\mathbf{u} \times \mathbf{v} = [v_y \ -v_x][u_x \ u_y]^T$ we obtain the respective entries in the matrix \mathbf{B} and the vector \mathbf{a} , which are then assembled in the linear relationship

$$\boldsymbol{\tau}_{\text{tendon}} = \mathbf{B}(\boldsymbol{\theta})\mathbf{p} + \mathbf{a}(\boldsymbol{\theta}) \quad (\text{B.5})$$

which can then be integrated in the complete grasp formulation presented in Chapter 6.

Appendix C

Maximum Frictional Torque as a Function of Contact Load

Consider a Winkler elastic foundation, of depth h and elastic modulus K , resting on a rigid base and in contact with a rigid object applying a total load P . Using the results of Johnson [1985], the compression δ of the elastic layer caused by the indenter at the center of contact will be:

$$\delta = \sqrt{\frac{Ph}{K\pi(R'R'')^{\frac{1}{2}}}} \quad (\text{C.1})$$

where R' and R'' are the relative radii of curvature computed as described in Chapter 7. The contact area will be described by an ellipse of semi-axes $a = \sqrt{2\delta R'}$ and $b = \sqrt{2\delta R''}$. The pressure distribution inside the contact area is:

$$p(x, y) = \frac{K\delta}{h} \left(1 - \frac{x^2}{a^2} - \frac{y^2}{b^2} \right) \quad (\text{C.2})$$

By integrating over the contact area we obtain the formula for total contact load:

$$P = \frac{K\pi ab\delta}{2h} \quad (\text{C.3})$$

According to Howe and Cutkosky [1996], maximum frictional moment will be applied if relative contact motion is a rotation around the pressure-weighted center

of the contact. The contribution of each contact point to the total friction moment is therefore

$$m(x, y) = \sqrt{x^2 + y^2} \mu p(x, y) \quad (\text{C.4})$$

By integrating eq. (C.4) over the entire contact area we obtain the value of the maximum frictional moment than can be applied at the contact

$$\max(\tau_n) = \frac{K\delta}{h} \mu \frac{4\pi}{15} (ab)^{\frac{3}{2}} \quad (\text{C.5})$$

and from (C.3) and (C.5) we obtain

$$\frac{\max(\tau_n)}{P} = \frac{8\mu}{15} \sqrt{ab} \quad (\text{C.6})$$

Bibliography

- P. Afshar and Y. Matsuoka. Neural-based control of a robotic hand: Evidence for distinct muscle strategies. In *IEEE Intl. Conf. on Robotics and Automation*, pages 4633–4638, New Orleans, 2004.
- J. Aleotti and S. Caselli. Grasp recognition in virtual reality for robot pregrasp planning by demonstration. In *IEEE International Conference on Robotics and Automation*, pages 2801–2806, 2006.
- K. An, E. Chao, W. Cooney, and R. Linscheid. Normative model of human hand for biomechanical analysis. *J. of Biomechanics*, 12:775–788, 1979.
- M. Anitescu and F. A. Potra. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dynamics*, 14: 231–247, 1997.
- P. K. Artemiadis, G. Shakhnarovich, C. Vargas-Irwin, J. P. Donoghue, and M. J. Black. Decoding grasp aperture from motor-cortical population activity. In *IEEE EMBS Conference on Neural Engineering*, pages 518–521, 2007.
- F. Barbagli, A. Frisoli, K. Salisbury, and M. Bergamasco. Simulating human fingers: a soft finger proxy model and algorithm. In *International Symp. on Haptic Interfaces*, pages 9–17, 2004.
- A. Bicchi and V. Kumar. Robotic grasping and contact: a review. In *IEEE International Conference on Robotics and Automation*, pages 348–353, 2000.
- A. Bicchi and D. Prattichizzo. Analysis and optimization of tendinous actuation for biomorphically designed robotic systems. *Robotica*, 18:23–31, 2000.
- L. Birglen, T. Laliberte, and C. Gosselin. *Underactuated Robotic Hands*. Springer Tracts in Advanced Robotics, 2008.

- S. Bitzer and P. van der Smagt. Learning emg control of a robotic hand: Towards active prostheses. In *IEEE Intl. Conf. on Robotics and Automation*, pages 2819–2823, Orlando, 2006.
- C. Borst, M. Fischer, and G. Hirzinger. A fast and robust grasp planner for arbitrary 3D objects. In *IEEE International Conference on Robotics and Automation*, pages 1890–1896, Detroit, MI, May 1999.
- C. Borst, M. Fischer, and G. Hirzinger. Grasp planning: How to choose a suitable task wrench space. In *IEEE International Conference on Robotics and Automation*, pages 319–325, 2004.
- D. Brock. Enhancing the dexterity of a robot hand using controlled slip. In *IEEE International Conference on Robotics and Automation*, pages 249–251, 1988.
- C. Brown and H. Asada. Inter-finger coordination and postural synergies in robot hands via mechanical implementation of principal components analysis. In *IEEE-RAS International Conference on Intelligent Robots and Systems*, pages 2877–2882, 2007.
- M. Buss, H. Hashimoto, and J. Moore. Dextrous hand grasping force optimization. *IEEE Trans. on Robotics and Automation*, 12:406–418, 1996.
- J. Butterfass, G. Hirzinger, S. Knoch, and H. Liu. DLR’s articulated hand, part I: Hard- and software architecture. In *IEEE International Conference on Robotics and Automation*, pages 2081–2086, 1998.
- M. C. Carrozza, G. Cappiello, S. Micera, B. B. Edin, L. Beccai, and C. Cipriani. Design of a cybernetic hand for perception and action. *Biol. Cybern.*, 95(6):629–644, 2006.
- Census. U.S. Census Bureau. http://www.census.gov/Press-Release/www/releases/archives/aging_population/013988.html, 2009.
- D. C. Chang and M. R. Cutkosky. Rolling with deformable fingertips. In *IEEE International Conference on Intelligent Robots and Systems*, pages 2194–2199, 1995.
- V. C. K. Cheung, A. dAvella, M. C. Tresch, and E. Bizzi. Central and sensory contributions to the activation and organization of muscle synergies during natural motor behaviors. *Journal of Neuroscience*, 25(27):6419–6434, 2005.

- M. Ciocarlie, A. Miller, and P. Allen. Grasp analysis using deformable fingers. In *IEEE International Conference on Intelligent Robots and Systems*, pages 4122–4128, 2005.
- M. Ciocarlie, H. Dang, J. Lukos, M. Santello, and P. Allen. Functional analysis of finger contact locations during grasping. In *Joint Eurohaptics Conference and IEEE Symp. on Haptic Interfaces*, pages 401–405, 2009.
- C. Cipriani, F. Zaccone, G. Stellin, L. Beccai, G. Cappiello, M. Carrozza, and P. Dario. Closed-loop controller for a bio-inspired multi-fingered underactuated prosthesis. *IEEE International Conference on Robotics and Automation*, pages 2111–2116, 2006.
- M. R. Cutkosky. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Trans. on Robotics and Automation*, 5:269–279, 1989.
- K. Dandekar, B. Raju, and M. Srinivasan. 3-D finite-element models of human and monkey fingertips to investigate the mechanics of tactile sense. *J. Biomechanical Engineering*, 125:682–691, 2003.
- A. Dollar and R. Howe. Joint coupling design of underactuated grippers. In *30th Annual Mechanisms and Robotics Conference*, September 2006.
- A. Dollar and R. Howe. Simple, robust autonomous grasping in unstructured environments. In *IEEE International Conference on Robotics and Automation*, pages 4693–4700, 2007.
- Z. Doulergi and J. Fasoulas. Grasping control of rolling manipulations with deformable fingertips. *IEEE/ASME Transactions on Mechatronics*, 8(2):283–286, 2003.
- Z. Doulergi, A. Simeonidis, and S. Arimoto. A position/force control for a soft tip robotic finger under kinematic uncertainties. In *IEEE International Conference on Robotics and Automation*, pages 3867–3872, 2000.
- A. Edsinger and C. Kemp. Manipulation in human environments. In *IEEE-RSJ International Conference on Humanoid Robotics*, pages 102–109, 2006.
- C. Ferrari and J. Canny. Planning optimal grasps. In *IEEE International Conference on Robotics and Automation*, pages 2290–2295, 1992.

- A. Fod, M. Mataric, and O. Jenkins. Automated derivation of primitives for movement classification. *Autonomous Robots*, 12:39–54(16), 2002.
- A. Frisoli, F. Barbagli, E. Ruffaldi, K. Salisbury, and M. Bergamasco. A limit-curve based soft finger god-object algorithm. In *International Symp. on Haptic Interfaces*, pages 217–223, 2006.
- J. Fu and N. Pollard. On the importance of asymmetries in grasp quality metrics for tendon driven hands. In *IEEE-RAS Intl. Conf. on Intelligent Robots and Systems*, 2006.
- A. P. Georgopoulos, A. B. Schwartz, and R. E. Kettner. Neuronal population coding of movement direction. *Science*, 233(4771):1416–1419, Sep 1986.
- A. P. Georgopoulos, G. Pellizzer, A. V. Poliakov, and M. H. Schieber. Neural coding of finger and wrist movements. *Journal of Computational Neuroscience*, 6(3):279–288, 1999.
- C. Goldfeder, M. Ciocarlie, H. Dang, and P. Allen. The Columbia grasp database. In *IEEE International Conference on Robotics and Automation*, 2009a.
- C. Goldfeder, M. Ciocarlie, J. Peretzman, H. Dang, and P. Allen. Data-driven grasping with partial sensor data. In *IEEE International Conference on Intelligent Robots and Systems*, 2009b.
- C. Gosselin, T. Laliberte, and T. Degoulange. Underactuated robotic hand. In *Video Proceedings of the IEEE Intl. Conf. on Robotics and Automation*, 1998.
- C. Gosselin, F. Pelletier, and T. Laliberte. An anthropomorphic underactuated robotic hand with 15 Dofs and a single actuator. *IEEE Intl. Conf. on Robotics and Automation*, 2008.
- S. Goyal, A. Ruina, and J. Papadopoulos. Planar sliding with dry friction, part 1. *Wear*, 143:307–330, 1991.
- GPL. The GNU General Public License. <http://www.gnu.org/licenses>, 2009.
- H. Y. Han, S. Arimoto, K. Tahara, M. Yamaguchi, and P. Nguyen. Robotic pinching by means of a pair of soft fingers with sensory feedback. In *IEEE International Conference on Robotics and Automation*, pages 97–102, 2001.

- L. Han, J. Trinkle, and Z. Li. Grasp analysis as linear matrix inequality problems. *IEEE Trans. on Robotics and Automation*, 16:663–674, 2000.
- S. Hirose and Y. Umetani. The development of soft gripper for the versatile robot hand. *Mechanism and Machine Theory*, 13:351–358, 1978.
- R. Howe and M. Cutkosky. Practical force-motion models for sliding manipulation. *International Journal of Robotics Research*, 15(6):557–572, 1996.
- K. Hsiao, L. Kaelbling, and T. Lozano-Perez. Grasping POMDPs. In *IEEE International Conference on Robotics and Automation*, pages 4685–4692, 2007.
- K. H. Hunt. *Kinematic Geometry of Mechanisms*. Oxford University Press, 1978.
- T. Iberall. Human prehension and dexterous robot hands. *International Journal of Robotics Research*, 16:285–299, 1997.
- L. Ingber. Very fast simulated re-annealing. *J. Mathl. Comput. Modelling*, 12(8):967–973, December 1989.
- S. Jacobsen, J. Wood, D. Knutti, and K. Biggers. The Utah/M.I.T. dextrous hand: Work in progress. *International Journal of Robotics Research*, 3(21):21–50, 1984.
- K. Johnson. *Contact Mechanics*. Cambridge University Press, 1985.
- J. Kerr and B. Roth. Analysis of multifingered hands. *International Journal of Robotics Research*, 4(4):3–17, 1986.
- D. Kragic, A. Miller, and P. Allen. Real-time tracking meets online planning. In *IEEE International Conference on Robotics and Automation*, pages 2460–2465, 2001.
- P. G. Kry and D. K. Pai. Interaction capture and synthesis. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 25(3), 2006.
- R. Kurtz and V. Hayward. Dexterity measure for tendon actuated parallel mechanisms. In *IEEE Intl. Conf. on Advanced Robotics*, 1991.
- T. Laliberte, L. Birglen, and C. M. Gosselin. Underactuation in robotic grasping hands. *Machine Intelligence & Robotic Control*, 4(3):1–11, 2002.
- Y. Li, J. L. Fu, and N. S. Pollard. Data-driven grasp synthesis using shape matching and task-based pruning. *IEEE Trans. on Visualization and Computer Graphics*, 13(4):732–747, 2007.

- Z. Li and S. Sastry. Task-oriented optimal grasping by multifingered robot hands. *IEEE Journal of Robotics and Automation*, 4(1):32–44, 1988.
- Y.-H. Liu, M.-L. Lam, and D. Ding. A complete and efficient algorithm for searching 3-D form-closure grasps in the discrete domain. *IEEE Transactions on Robotics*, 20(5):805–816, 2004.
- C. S. Lovchik and M. A. Diftler. The Robonaut hand: A dextrous robot hand for space. In *IEEE International Conference on Robotics and Automation*, pages 907–912, 1998.
- C. S. Lovchik and M. A. Diftler. The Robonaut hand: A dextrous robot hand for space. In *IEEE International Conference on Robotics and Automation*, pages 907–912, 1999.
- C. R. Mason, J. E. Gomez, and T. J. Ebner. Hand synergies during reach-to-grasp. *Journal of Neurophysiology*, 86:2896–2910, 2001.
- M. Mason and K. Salisbury. *Robot hands and the mechanics of manipulation*. MIT Press, 1985.
- A. Miller and P. Allen. Examples of 3-D grasp quality computations. In *IEEE International Conference on Robotics and Automation*, pages 1240–1246, 1999.
- A. Miller and P. K. Allen. GraspIt!: a versatile simulator for robotic grasping. *IEEE Robotics and Automation Magazine*, 11(4):110–122, 2004.
- A. Miller and H. Christensen. Implementation of multi-rigid-body dynamics within a robotic grasping simulator. In *IEEE Intl. Conference on Robotics and Automation*, pages 2262–2268, 2003.
- A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen. Automatic grasp planning using shape primitives. In *IEEE International Conference on Robotics and Automation*, pages 1824–1829, 2003.
- Mosek. Mosek ApS Denmark. <http://www.mosek.com>.
- J. R. Napier. The prehensile movements of the human hand. *Journal of Bone and Joint Surgery*, 38:902–913, 1956.
- V.-D. Nguyen. Constructing force-closure grasps. In *IEEE International Conference on Robotics and Automation*, pages 2290–2295, 1986.

- M. Pauly, D. K. Pai, and L. Guibas. Quasi-rigid objects in contact. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2004.
- R. Platt, A. H. Fagg, and R. Grupen. Nullspace composition of control laws for grasping. In *IEEE International Conference on Robotics and Automation*, pages 1717–1723, 2002.
- R. Platt, A. H. Fagg, and R. Grupen. Manipulation gaits: Sequences of grasp control tasks. In *IEEE International Conference on Robotics and Automation*, pages 801–806, 2004.
- N. Pollard and R. Gilbert. Tendon arrangement and muscle force requirements for humanlike force capabilities in a robotic finger. *IEEE Intl. Conf. on Robotics and Automation*, pages 3755–3762, 2002.
- J. Ponce, S. Sullivan, J.-D. Boissonnat, and J.-P. Merlet. On characterizing and computing three- and four-finger force-closure grasps of polyhedral objects. In *IEEE International Conference on Robotics and Automation*, pages 821–827, 1993.
- D. Prattichizzo and J. C. Trinkle. Grasping. In *Springer Handbook of Robotics*, pages 671–700. Springer, 2008.
- N. Rezzoug and P. Gorce. A biocybernetic method to learn hand grasping posture. *Kybernetes*, 32(4):478–490, 2003.
- M. Roa and R. Suarez. Geometrical approach for grasp synthesis on discretized 3D objects. In *IEEE-RSJ International Conference on Intelligent Robots and Systems*, pages 3283–3288, 2007a.
- M. Roa and R. Suarez. Determination of independent contact regions on discretized 3d objects. In *IEEE International Symposium Conference on Assembly and Manufacturing*, pages 191–196, 2007b.
- M. Santello, M. Flanders, and J. F. Soechting. Postural hand synergies for tool use. *Journal of Neuroscience*, 18(23):10105–10115, 1998.
- M. Santello, M. Flanders, and J. F. Soechting. Patterns of hand motion during grasping and the influence of sensory guidance. *Journal of Neuroscience*, 22:1426–1435, 2002.
- A. Saxena, J. Driemeyer, and A. Ng. Robotic grasping of novel objects using vision. *International Journal of Robotics Research*, 27(2):157–173, 2008.

- Shadow Hand. The Shadow Hand. <http://www.shadowrobot.com/hand/>.
- K. B. Shimoga. Robot grasp synthesis algorithms: a survey. *International Journal of Robotics Research*, 15:230–266, 1996.
- K. B. Shimoga and A. A. Goldenberg. Soft materials for robotic fingers. In *IEEE International Conference on Robotics and Automation*, pages 1300–1305, 1992.
- S. Stansfield. Robotic grasping of unknown objects: A knowledge-based approach. *International Journal of Robotics Research*, 10(4):314–326, 1991.
- S. Sueda, A. Kaufman, and D. K. Pai. Musculotendon simulation for hand animation. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 27(3), 2008.
- M. Tada and D. K. Pai. Finger shell: Predicting finger pad deformation under line loading. In *Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2008.
- D. Taylor, S. H. Tillery, and A. Schwartz. Information conveyed through brain control: Cursor versus robot. *IEEE Trans. Neural Systems Rehab Eng.*, 1(2):195–199, 2003.
- D. M. Taylor, S. H. Tillery, and A. B. Schwartz. Direct cortical control of 3D neuroprosthetic devices. *Science*, 296(5574):1829–1832, 2002.
- P. H. Thakur, A. J. Bastian, and S. Hsiao. Multidigit movement synergies of the human hand in an unconstrained haptic exploration task. *Journal of Neuroscience*, 28(6):1271–1281, 2008.
- E. Todorov and Z. Ghahramani. Analysis of the synergies underlying complex hand manipulation. In *26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4637–4640, 2004.
- J. C. Trinkle. A quantitative test for form closure grasps. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1670–1676, 1992.
- L.-W. Tsai. *Robot Analysis*. John Wiley & Sons, 1999.
- W. Tsang, K. Singh, and E. Fiume. Helping hand: an anatomically accurate inverse dynamics solution for unconstrained hand motion. In *Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, 2005.
- A. Tsoli and O. C. Jenkins. Robotic grasping for prosthetic applications. In *International Symposium of Robotics Research*, 2007.

- M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
- N. Ulrich, R. Paul, and R. Bajcsy. A medium-complexity compliant end effector. In *IEEE International Conference on Robotics and Automation*, pages 434–439, 1988.
- F. Valero-Cuevas and H. Lipson. A computational environment to simulate complex tendinous topologies. In *26th Annual International Conference of the IEEE EMBS*, 2004.
- F. Valero-Cuevas, F. E. Zajac, and C. G. Burgar. Large index-fingertip forces are produced by subject-independent patterns of muscle excitation. *J. of Biomechanics*, 31:693, 1998.
- M. Vande Weghe, M. Rogers, M. Weissert, and Y. Matsuoka. The ACT hand: Design of the skeletal structure. In *IEEE International Conference on Robotics and Automation*, pages 3375–3379, 2004.
- M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford, and A. B. Schwartz. Cortical control of a prosthetic arm for self-feeding. *Nature*, 453(7198):1098–1101, Jun 2008.
- N. Xydas and I. Kao. Modeling of contact mechanics with experimental results for soft fingers. In *IEEE International Conference on Intelligent Robots and Systems*, pages 488–493, 1998.
- F. E. Zajac. Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Critical Reviews in Biomedical Engineering*, 17: 359–411, 1989.
- M. Zecca, S. Micera, M. C. Carrozza, and P. Dario. Control of multifunctional prosthetic hands by processing the electromyographic signal. *Critical Reviews in Biomedical Engineering*, 30:459–485, 2002.
- C. Zilles and J. Salisbury. A constraint-based god-object method for haptic display. In *IEEE International Conference on Intelligent Robots and Systems*, pages 146–151, 1995.