

Dynamic Grasp Adaptation -- From Humans To Robots

THÈSE N° 6908 (2016)

PRÉSENTÉE LE 8 FÉVRIER 2016

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE D'ALGORITHMES ET SYSTÈMES D'APPRENTISSAGE
PROGRAMME DOCTORAL EN SYSTÈMES DE PRODUCTION ET ROBOTIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Miao LI

acceptée sur proposition du jury:

Prof. J. Paik, présidente du jury
Prof. A. Billard, directrice de thèse
Prof. K. Tahara, rapporteur
Prof. A. Rodriguez, rapporteur
Prof. J. del R. Millan, rapporteur



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2016

ABSTRACT

THE human hand is an amazing tool, demonstrated by its incredible motor capability and remarkable sense of touch. To enable robots to work in a human-centric environment, it is desirable to endow robotic hands with human-like capabilities for grasping and object manipulation. However, due to its inherent complexity and inevitable model uncertainty, robotic grasping and manipulation remains a challenge. This thesis focuses on *grasp adaptation* in the face of model and sensing uncertainties: Given an object whose properties are not known with certainty (e.g., shape, weight and external perturbation), and a multifingered robotic hand, we aim at determining *where* to put the fingers and *how* the fingers should adaptively interact with the object using tactile sensing, in order to achieve either a stable grasp or a desired dynamic behaviour.

A central idea in this thesis is the *object-centric dynamics*: namely, that we express all control constraints into an object-centric representation. This simplifies computation and makes the control versatile to the type of hands. This is an essential feature that distinguishes our work from other robust grasping work in the literature, where generating a static stable grasp for a given hand is usually the primary goal. In this thesis, grasp adaptation is a dynamic process that flexibly adapts the grasp to fit some purpose from the object’s perspective, in the presence of a variety of uncertainties and/or perturbations. When building a grasp adaptation for a given situation, there are two key problems that must be addressed: 1) the problem of choosing an initial grasp that is suitable for future adaptation, and more importantly 2) the problem of designing an adaptation strategy that can react adequately to achieve desired behaviour of the grasped object.

To address challenge 1 (planning a grasp under shape uncertainty), we propose an approach to parameterizing the uncertainty in object shape using Gaussian Processes (GPs) and incorporate it as a constraint into contact-level grasp planning. To realize the planned contacts using different hands interchangeably, we further develop a probabilistic model to predict the feasible hand configurations, including hand pose and finger joints, given the desired contact points only. The model is built using the concept of *Virtual Frame*(VF), and it is independent from the choice of hand frame and object frame. The performance of the proposed approach is validated on two different robotic hands, an industrial gripper (4 DOF Barrett hand) and a humanoid hand (16 DOF Allegro hand) to manipulate objects of daily use with complex geometry and various texture (a spray bottle, a tea caddy, a jug and a bunny toy).

In the second part of this thesis, we propose an approach to the design of adaptation strategy to ensure grasp stability in the presence of physical uncertainties of objects (object weight, friction at contacts and external perturbation). Based on an object-level impedance controller, we first design a grasp stability estimator in the object frame using the grasp experience and tactile sensing. Once a grasp is predicted to be unstable during online execution, the grasp adaptation strategy is triggered to improve the grasp stability, by either changing the stiffness at finger level or relocating the position of one fingertip to a better area. We validate our approach using the Allegro hand equipped with the tactile sensor – BioTac on each fingertip, from which a variety of contact conditions are implicitly extracted (contact normals, positions, normal and tangential forces). Two different types of experiments are conducted: increasing the object weight gradually and shaking the object with different accelerations using a robotic arm (7 DOF KUKA LWR). The experimental results demonstrate the robustness of our method on objects with different physical properties (a cola can, a juice bottle, a plastic cup, a milk box and a jug).

In the last part of this thesis, we further extend the grasp adaptation approach for Constrained Object Manipulation (COM), namely move the object to a desired configuration. We first propose a manifold learning approach to encode the task requirement. Built on the learned task manifold and an object-level impedance controller, an adaptation strategy is designed to adapt for potential disturbance by exploiting the local geometry on the manifold. The effectiveness of the approach is demonstrated using the Allegro hand on several typical COM tasks.

Throughout this thesis, we adopt the approach of Learning by Demonstration (LbD), by investigating how humans adapt their grasps under different perturbations. Using demonstrations of skilled in-hand manipulation, we learn how to characterize the dynamics of the object’s motion. We then use this knowledge to plan and perform corrective actions for robotic grasping and manipulation. We show that this can be achieved by encoding the correlations among grasp configurations, tactile readings and external perturbations into a single statistic model, from which the optimal adaptation strategies can be built and used in real time. Throughout this thesis we show that the object-centric dynamic approach to robotic grasping and manipulation can accommodate a large variety of uncertainties and perturbations, and moreover it is applicable to different robotic hands with various kinematic structures.

KEYWORDS: Grasp Adaptation, Object-level Impedance Control, Dexterous Manipulation, Tactile Sensing, Learning by Demonstration

RÉSUMÉ

La main humaine est un outil extraordinaire, la preuve en est par ses impressionnantes capacités motrices et son remarquable sens du toucher. Pour permettre aux robots de travailler dans un environnement centré sur les hommes, il est souhaitable de doter les mains robotiques de capacités humaines pour la préhension et la manipulation d'objets. Toutefois, en raison de leur complexité et de l'inexactitude inhérente à tout modèle possible, la préhension et la manipulation d'objets restent un défi.

Cette thèse porte sur l'adaptation de la préhension robotique face aux incertitudes de modèle et de perception. Étant donné qu'un objet dont les propriétés ne sont pas connues avec certitude (par exemple, la forme, le poids et les perturbations externes), et une main robotisée à plusieurs doigts, nous visons à déterminer où placer les doigts et comment les doigts doivent interagir de manière adaptative à l'objet en utilisant les sensations tactiles, afin de parvenir à une préhension stable ou bien à un comportement dynamique souhaité.

Une idée centrale de cette thèse est celle de la dynamique centrée sur l'objet: à savoir, que nous exprimons toutes les contraintes de contrôle en une représentation centrée sur l'objet. Cela simplifie le calcul et rend le contrôle adaptable à tout type de main. Ceci est une caractéristique essentielle qui distingue notre travail d'autres travaux de préhension robuste dans la littérature, où la génération d'une préhension statique stable pour une main donnée est habituellement le principal objectif. Dans cette thèse, l'adaptation de la préhension est un processus dynamique qui modifie la préhension pour atteindre un but exprimé dans le référentiel de l'objet, en présence d'une variété d'incertitudes et/ou de perturbations. Lors du design de l'adaptation d'une préhension pour une situation donnée, il y a deux problèmes clés qui doivent être abordés: 1) le problème du choix d'une préhension initiale qui est appropriée pour l'adaptation future, et plus important encore 2) le problème de la conception d'une stratégie d'adaptation qui peut réagir de manière adéquate pour atteindre le comportement souhaité de l'objet saisi.

Pour répondre au premier défi (design d'une préhension sous contraintes d'incertitudes), nous proposons une approche pour paramétrer l'incertitude sur la forme de l'objet en utilisant des Gaussian Process (GPs) et de l'intégrer comme une contrainte dans la planification de la préhension. Pour réaliser les contacts prévus avec différentes mains interchangeables, nous développons un modèle probabiliste pour prédire les configurations possibles de la main, y compris la position de la main et celles des articulations

des doigts, étant donné les points de contact désirés. Le modèle est construit en utilisant le concept de Virtual Frame (VF), et il est indépendant du choix de référentiel de la main et de celui de l'objet. La performance de l'approche proposée est validée sur deux mains robotiques différentes, un préhenseur industriel (4 DDL main Barrett) et une main humanoïde (16 DDL main Allegro) pour manipuler des objets d'usage quotidien avec une géométrie complexe et diverses textures (un spray, une boîte à thé, une cruche et un jouet en forme de lapin).

Dans la deuxième partie de cette thèse, nous proposons une approche pour la conception de la stratégie d'adaptation pour assurer la stabilité de la préhension en présence d'incertitudes sur les propriétés physiques des objets (poids de l'objet, frottement des contacts et perturbation externe). Basé sur un contrôleur d'impédance au niveau de l'objet, nous concevons d'abord un estimateur de la stabilité de préhension dans le référentiel de l'objet en utilisant l'expérience de préhension et de sensation tactile. Lorsque qu'il est calculé qu'une manière de saisir est instable durant l'exécution, la stratégie d'adaptation de la préhension est déclenchée pour améliorer la stabilité de la saisie, soit en changeant la rigidité au niveau du doigt ou en déplaçant la position d'un doigt vers une meilleure région de l'objet.

Nous validons notre approche en utilisant la main Allegro équipée du capteur tactile - BioTac sur chaque doigt, à partir de laquelle une variété de conditions de contact sont implicitement extraites (normales de contact, les positions normales et forces tangentielles). Deux types d'expériences différentes sont menées: en augmentant progressivement le poids de l'objet et en secouant l'objet avec différentes accélérations en utilisant un bras robotisé (7 DDL KUKA LWR). Les résultats expérimentaux démontrent la robustesse de notre méthode avec des objets aux propriétés physiques différentes (une canette de soda, une bouteille de jus de fruits, une tasse en plastique, un pack de lait et une cruche).

Dans la dernière partie de cette thèse, nous étendons encore l'approche de l'adaptation de la préhension pour la manipulation contrainte d'objets (MCO), à savoir déplacer l'objet dans une configuration souhaitée. Nous proposons tout d'abord une approche de "manifold learning" pour l'encodage des exigences de la tâche. Construite sur la tâche apprise et un contrôleur en impédance au niveau de l'objet, une stratégie d'adaptation est conçue pour s'adapter aux perturbations potentielles en exploitant la géométrie locale du modèle. L'efficacité de l'approche est démontrée en utilisant la main Allegro sur plusieurs tâches typiques de MCO.

Tout au long de cette thèse, nous adoptons l'approche de l'apprentissage par démonstration (APD), en étudiant comment les humains adaptent leur préhension sous différentes perturbations. En utilisant des démonstrations expertes de manipulation dans la main, nous apprenons comment caractériser la dynamique du mouvement de l'objet. Nous utilisons ensuite cette connaissance pour planifier et exécuter des actions correctives pour préhension et la manipulation robotique. Nous montrons que ceci peut être réalisé en codant les corrélations entre les configurations de préhension, des capteurs tactiles et des perturbations externes dans un seul modèle statistique, à partir duquel des stratégies d'adaptation optimales peuvent être construites et utilisées

en temps réel. Tout au long de cette thèse, nous montrons que l’approche dynamique centrée sur l’objet de la préhension et de la manipulation robotique peuvent accueillir une grande variété d’incertitudes et de perturbations, et que de plus elle est applicable à différentes mains robotiques avec diverses structures cinématiques.

MOTS CLÉ: Portée l’adaptation, le niveau-objet de contrôle Impédance, Manipulation Agile, Tactile Sensing, Apprendre en démonstration

*To my beloved Hong and our son Noah,
who are the joy of my life*

*To my parents,
for their endless support for my study*

ACKNOWLEDGMENTS

I am grateful to my advisor Aude Billard for giving me the opportunity to pursue my PhD under her supervision in such an amazing lab – LASA. I really appreciate her for teaching me so many invaluable research and life experience, and this thesis would not have been possible without her continuous support, patience, insights and availability. Thanks to her for all the inspiring discussions, for all the time she spent to read and correct my papers, for her insights of great researches, and for all her valuable advice during the past four years.

I would like to acknowledge all my thesis examiners, Prof. Jose del R. Millan (EPFL), Prof. Kenji Tahara (Kyushu University) and Prof. Alberto Rodriguez (MIT) for taking time to provide valuable comments and suggestions on the earlier version of this manuscript. Many thanks also to Prof. Jamie Paik (EPFL) who served as president of my thesis committee.

I would like to thank all my coauthors during my PhD studies, from whom I have learned various skills and insights about research. I am hugely indebted to Prof. Kenji Tahara for his guidance and discussions with me on robotic hand control, during his one year visiting at LASA and the tele-meetings after his return to Japan. I am greatly thankful to Dr. Sahar El-Khoury, with whom I started the first research topic on grasp planning. It is a very memorable experience to work with her and I was really lucky to have her support during the first year. I would like to thank Prof. Danica Kragic at KTH for her efforts and supports on several joint works on robotic grasping. I am grateful to have the chance to work closely with Kaiyu Hang and Dr. Yasemin Bekiroglu from her lab, on the integration of grasp planning and control. I would also like to thank Prof. Joanna J. Bryson at University of Bath and Dr. Bidan Huang from her lab to have all the interesting discussions. Thanks to Nicolas Sommer, Hang Yin and Ravin Luis De Souza in the LASA lab, with whom we have explored various research topics together. Specially, I would like to thank Aude again to allow me and provide the possibilities to have all these different collaborations.

I felt really lucky to have worked with my colleagues in the LASA lab, Basilio Noris, Sahar El Khoury, Seungsu Kim, Ashwini Shukla, Seyed Mohammad Khansari-Zadeh, Lucia Ureche, Nicolas Sommer, Klas Kronander, Guillaume de Chambrier, Bidan Huang, Mustafa Suphi Erden, Ravin Luis De Souza, Hang Yin, Nadia Figueroa, Seyed Sina Mirrazavi Salehian, Mahdi Khoramshahi, AJung Moon, Nili Eliana Krausz, Denys Lamotte, Luka Lukic, Silvia Magrelli, Sylvain Calinon, Laura Cohen, Felix

Duvallet, Iason Batzianoulis, Joao Pedro Isaias Abrantes. I truly thank them for all the interesting conversations on various topics and for the inspiring working environment they have built. Many thanks to Basilio Noris and also Marion for all the interesting discussions, ranging from scientific topics to different national cultures. Special thanks to Lucia Ureche for all her efforts and help during the RoboHow project. Thanks to Ashwini Shukla and Guillaume de Chambrier for having the relaxing after-working chat in the swimming pool. Thanks to Ashwini Shukla, Hang Yin, Klas Kronander, Guillaume de Chambrier and Ravin Luis De Souza, for all the coding and debugging skills that help me to apply my methods on real robots.

I would like to also thank our lab secretary Joanna Charlotte Erfani and the secretary of EDPR Corinne Lebet, for taking care of the administrative aspects of my PhD. Many thanks also for the EU Project RoboHow for supporting my research.

I would have never reached the place where I am now without the infinite mental support from my parents and also my parents-in-law. Many thanks to my younger sister and brothers in law for all their back up, and the nice moments we have spent together.

Last but not least, I would like to deeply thank my lovely wife Hong, for being an endless source of support, encouragement, and happiness. I am greatly indebted for all her efforts to our family and her infinite support for our life journeys. Many thanks to her and also our son Noah for bringing me so much joyful time together.

TABLE OF CONTENTS

1	Introduction	1
1.1	Motivation	1
1.2	Contributions	2
1.3	Thesis Outline	4
2	Background	7
2.1	Dynamics of Robotic Grasping	7
2.2	Grasp Planning	10
2.3	Grasp Control	12
2.4	Grasp Learning	13
2.4.1	Learning in grasp planning	14
2.4.2	Learning in grasp control	15
2.5	Our Approach	16
3	Dexterous Grasping Under Shape Uncertainty	19
3.1	Introduction	19
3.2	Problem Statement	20
3.2.1	Grasp planning as optimization	20
3.2.2	Challenge	21
3.2.3	Our approach	24
3.3	Grasp Planning under Shape Uncertainty	25
3.3.1	Object modeling for grasping	25
3.3.2	Grasp planning with shape uncertainty constraints	27
3.4	Hand Configuration in Virtual Frame	29
3.4.1	Probabilistic model for hand inverse kinematics	29
3.4.2	On-line hand configuration query	30
3.5	Grasp Control under Shape Uncertainty	32
3.6	Implementation and Experimental Results	33
3.6.1	Results for object surface modeling	34
3.6.2	Results for grasp planning	34
3.6.3	Results for hand configuration query	35
3.6.4	Results for grasp realization	38
3.6.5	Implementation on real robotic hand	39
3.7	Discussion and Summary	42
4	Dynamic Grasp Adaptation	43
4.1	Introduction	43
4.2	Problem Statement	44
4.2.1	Grasp stability prediction	44
4.2.2	Dynamic grasp adaptation	47
4.2.3	Pipeline of our approach	48
4.3	Grasp Stability Estimation	49

4.4	Grasp Adaptation	52
4.4.1	Impedance adaptation	52
4.4.2	Local exploration	54
4.5	Experimental Evaluation	55
4.5.1	Setup and data collection	55
4.5.2	Results for grasp stability estimation	57
4.5.3	Results for grasp adaptation	59
4.5.4	Discussion	61
4.6	Integration with Grasp Planning	62
4.6.1	System setup	62
4.6.2	Experimental results	64
4.7	Discussion and Summary	67
5	Task Manifold for Constrained Object Manipulation	69
5.1	Introduction	69
5.2	Problem Statement	70
5.2.1	Constrained manipulation planning	71
5.2.2	Constrained manipulation control	72
5.3	Task Manifold Learning	73
5.4	Planning on Task Manifold	75
5.4.1	Path planning on manifold	75
5.4.2	Path replanning on Manifold	76
5.4.3	Learning projection onto manifold	77
5.5	Object-level Impedance Control on Task Manifold	78
5.5.1	Object-level impedance control	79
5.5.2	Impedance learning on task manifold	79
5.5.3	Impedance selection for projection	82
5.6	Simulation and Experimental Results	82
5.6.1	Simulation results for task manifold learning	83
5.6.2	Experimental results for dexterous manipulation	86
5.6.3	Experimental results for constrained manipulation	92
5.7	Discussion and Summary	97
6	Conclusion and Summary	99
6.1	Main Contributions	99
6.2	Limitations and Future Work	100
6.3	Final Words	102
	Appendices	105
	Appendix A Object-level Impedance Control	107
	Appendix B Stability of Variable Grasp Stiffness Control	109
B.1	Dynamics	109
B.1.1	Contact model of soft fingertip	109
B.1.2	Rolling constraints	110
B.1.3	Overall dynamics	110
B.2	Variable Grasp Stiffness Control	111
B.3	Stability Proof-1	111
B.4	Stability Proof-2	112
B.5	Identities	114
	References	117
	Curriculum Vitae	129

INTRODUCTION

1.1 MOTIVATION

The human hand is an amazing tool that centrally involved in most of our daily activities. We use our hands to feel, explore, grasp and manipulate objects, and further to perform some cognitive skills like writing and playing music instruments, as shown in Fig. 1.1. Such variety of skills is achievable in part thanks to the incredible dexterity of the human hand and in part thanks to our marvelous control strategy.

Even though we use the hand everyday, it is still difficult to explain how the Central Nervous System (CNS) control the the large number of degrees of freedom of the human hand. Evidence from neuroscience indicates that our brain relies on a rich repertoire of sensory feedback mainly from the vision and the touch, to achieve robust grasping and dexterous manipulation (Castiello, 2005; Johansson and Flanagan, 2009). During the process of hand actions, once the actual sensory information deviates from the expected one, a set of corrective strategies is triggered so that we can accomplish most of our daily activities reliably, despite of the variety and complexity of processes that are involved in these activities (Wolpert et al., 2011; Krakauer and Mazzoni, 2011).

When such strategies are applied to robotic grasping, a large number of assumptions are set up in order to facilitate the analysis of the grasping process. For instance, all contacts are often simplified as point contacts with friction, while the coefficient of friction, geometry and weight of the object are assumed to be precisely known. Under these assumptions, a given grasp can be analyzed in a static manner, and the desired grasp can be planned geometrically using the concept of force(or form) closure which requires a grasp to be able to resist arbitrarily disturbance (Bicchi, 2000). However, in real-world scenarios, these assumptions are either invalid (*e.g.*, the fingertip can provide arbitrarily grasping forces to resist external disturbance) or unnecessarily detailed (*e.g.*, accurate contact positions and contact normals). Moreover, such information may be unavailable or very difficult to obtain with certainty, especially in unstructured environments. As a consequence, the planned grasps may end up being unstable or infeasible even though they can possess a very high grasp quality in terms of force closure (J. Bohg and Kragic, 2014; Kim et al., 2013).

More importantly, the dynamics of grasping has not been taken in account and therefore the control of robotic grasping is still in a kinematic manner that can not adequately adapt for the variation of the environment or the unprepared perturbations. When considering dexterous manipulation (DM), where multiple fingers are required



Figure 1.1: Examples of human hand activities in everyday life. Three pictures (playing piano, unbuttoning, tuning watch) are adapted from video “BBC Dissected The Incredible Human Hand”, link: <https://www.youtube.com/watch?v=bxxU2G1MJYk>. More interesting and amazing facts about human hands can be found therein.

to cooperatively move the object to a desired configuration, it becomes more evident that the grasp needs to be adapted dynamically in terms of grasping forces and finger motions, in order to accomplish the task requirement and compensate for all the uncertain effects during the process.

To sum up, how to cope with the inherent uncertainties during grasping (object shape, weight, friction, elasticity and external perturbations etc.) and increase robot’s adaptation capabilities, has emerged as a fundamental and challenging problem in robotic grasping and manipulation. The major part of this thesis attempts to address this issue by integrating the efficient planning algorithm with the dynamic, robust and adaptive control technique, leveraging the power of different learning techniques. Both the planning algorithm and control technique are derived from data-driven models in an object-centric representation, which can essentially capture the uncertainties and thus give the robotic hands necessary capability to understand, predict and react to the uncertainties.

1.2 CONTRIBUTIONS

This thesis proposes a new approach that incorporates the model of uncertainty for planning and controlling of a robotic grasp, with the goal to improve the robustness and adaptability of robotic hands. It makes original contributions in the field of robotic grasping and manipulation from the following three aspects:

- **Dexterous grasping under shape uncertainty**

Most of the state-of-the-art dexterous grasping approach are built on the knowledge of accurate object geometry, which may be difficult or impossible to obtain in real-world scenarios. This thesis proposes a contact-level grasp plan-

ning approach that incorporates the uncertainty of object shape that is characterized using Gaussian Processes. To realize the planned grasping points with a given hand, a probabilistic hand inverse kinematics model is learned off-line and used on-line to predict the most likely hand configuration (finger joints and hand pose). Moreover, an uncertainty-aware complaint control scheme is devised to move the finger adaptively to the desired positions, which takes both the shape uncertainty and force uncertainty from the tactile sensing on fingertips into account.

- **Dynamic grasp adaptation**

We propose a framework for grasp adaptation, which is a dynamic process that flexibly adapts the grasp to fit some purpose from the object’s perspective. The main objective of the framework is to address the problem of grasp instability due to problems such as uncertain or varying object weights, slippage and external disturbances from collision or human perturbation. The framework implements both grasp stiffness adaptation through object-level impedance control and re-grasping/finger gaiting when the former is not sufficient. We demonstrate that this framework can significantly increase the robustness of robotic grasping by closing the loop between grasp planning and control through stability estimation and dynamic grasp adaptation. To the best of our knowledge, this is so far the only system that accomplishes grasp planning, stability estimation, on-line replanning and in-hand adaptation in a unified framework.

- **Task manifold for constrained object manipulation**

We propose a task-consistent adaptation strategy for post-grasp manipulation, where the grasped object is either manipulated in-hand or interacted with external environments (e.g. wipe a table, insert a bulb, polishing etc.). For these constrained object manipulation (COM) tasks, one of the main sources of uncertainties comes from the imprecise contact dynamics (contact between fingertips and object) and other possible external perturbations (external forces from the environment acting on the object). Also, due to the task constraints, the relevant control variables usually reside on a low-dimensional space which may not be known in advance. How to react to the uncertainties to maximize the task performance remains a challenge (Paolini et al., 2014; Dogar and Srinivasa, 2011). This thesis presents a learning approach to encode the task as a differential manifold, on which the task relevant motion is restricted to the tangent space. Moreover, the projection direction that can bring any off-manifold deviation onto the manifold is naturally defined and efficiently computed. This learned manifold is further integrated with an impedance controller that allows a fast task-consistent adaptation during on-line execution, where the proper control parameters are learned from human demonstration with the local geometric information of the manifold. Our experimental results demonstrate that a significant improvement of the task performance for COM can be achieved with the task manifold approach, both in simulation and on a real robot platform.

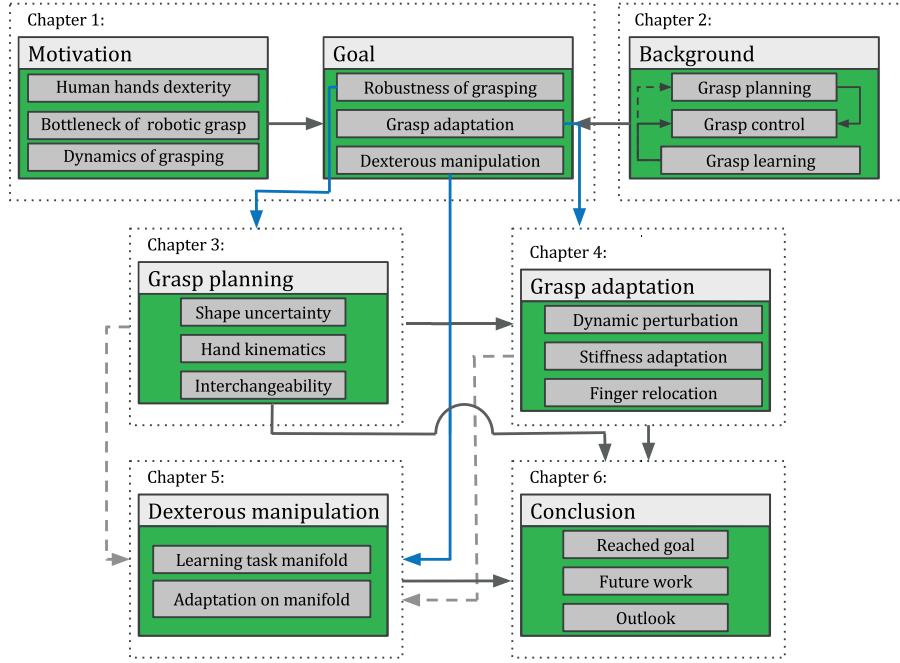


Figure 1.2: Thesis structure with key points of each chapter.

1.3 THESIS OUTLINE

Most of the results presented in this thesis have appeared previously or are still under review in peer-reviewed conference and robotic journals. In this section, we briefly summarize the content of each chapter and the structure of the thesis is described in Fig. 1.2.

Chapter 2 - Background and related work

This chapter presents a review of related work and preliminaries on the planning, control and learning of robotic grasping. The novelty of our approach are discussed in detail in relation to the state-of-the-art work in the area.

Chapter 3 - Dexterous grasping under shape uncertainty

In this chapter we present an approach for addressing the performance of dexterous grasping under object shape uncertainty. First, the uncertainty in object shape is explicitly parameterized using Gaussian Processes (GPs) and incorporated as a constraint into a contact level grasp planning. Then a probabilistic hand inverse kinematics model is proposed to predict the feasible hand configurations that can realize the planned grasping points using different robotic hands. Finally, a compliant finger closing scheme is devised by exploiting both the object shape uncertainty and tactile sensing at fingertips. Experimental evaluations on a 16 DOF Allegro Hand demonstrate that our method improves the performance of dexterous grasping under shape uncertainty. Major part of this chapter has been presented in (Li et al., 2015a) and in joint work with Sahar El-Khoury,

a former colleague at EPFL ([El Khoury et al., 2013, 2012](#)).

Chapter 4 - Dynamic grasp adaptation

In this chapter, a dynamic grasp adaptation strategy is presented, in order to stabilize the grasp under physical uncertainties including change in object weight, friction at the contact points and external perturbations. A grasp stability estimator is learned in the object frame through the grasp experience that is supervised by a human demonstrator. During grasp execution, once a grasp is predicted to be unstable by the stability estimator, a dynamic grasp adaptation is triggered, which activates either grasp stiffness adaptation through object-level impedance control or regrasping/finger gaiting when the former is not sufficient. The effectiveness of our approach is demonstrated by extensive experiments on Allegro hand mounted on KUKA LWR arm. This work has been initially presented in ([Li et al., 2014a](#)). We will also present part of the experimental results in the joint work with Kaiyu Hang and Danica Kragic at KTH ([Hang et al., 2015, 2014a](#)), which integrates our grasp adaptation strategy with their fast grasp planning algorithm ([Hang et al., 2014b](#)) to quantify the performance of the whole system with more systematic experiments.

Chapter 5 - Task manifold for COM

In this chapter, we integrate the task requirement into the adaptation model to further improve the robustness for constrained object manipulation (COM). First the task is encoded as a manifold, on which the feasible path for the task can be efficiently planned. Then, for each off-manifold point, the projection direction can be computed on-line in an iterative way. In addition, to improve the computation efficiency, a Gaussian Processes (GPs) is adopted to learn and predict the projection directions for new points. The task manifold along with the efficient projection thus induce an intuitive adaptation strategy that is consistent with the task requirements. Combined with an object-level impedance controller, the learned task manifold further facilitates the learning of the proper impedance parameters from human demonstration for a given task. Several simulations and experiments on Allegro hand demonstrate the validity of our approach. Part of this work has been published in ([Li et al., 2014b](#)).

Chapter 6 - Conclusions

In the final chapter, we conclude, by first providing a summary of the work achieved and its relations to our stated objectives. Then, we point out directions for extending this work along the research line of dynamic grasp adaptation, as well as an outlook for robotic grasping.

BACKGROUND

Robotic grasping has been an active research topic for decades and a considerable progress has been achieved, as witnessed by the various mechanical hands, efficient grasp planning algorithms and robust grasp control techniques (Bicchi, 2000). These achievements have laid the cornerstone for this thesis, and reviewing their strengths and weaknesses is helpful to delineate the contribution of this thesis. Due to the diversity of researches in this domain (e.g. planning, control, perception, mechanical design, etc.), this overview is not aimed to be exhaustive, but to provide necessary information to place this work among the state-of-the-art approaches.

The chapter unfolds as follows: In Section 2.1, we introduce the dynamics of robotic grasping, with the emphasis upon the assumptions which are generally required to allow precise analysis. Exploring the dynamics not only gives rise to the two fundamental problems in grasping: planning and control, but also underpins the theme of the whole thesis that a grasp should be able to adapt to uncertainty dynamically. In Section 2.2, we describe the techniques that aim at planning the optimal grasps in terms of contact points and feasible hand configuration. In Section 2.3, we present methods to control the robotic hands for finger closing and robust grasping. In Section 2.4, we review learning-based approaches and discuss how they can bridge the gap between planning and control. Finally in Section 2.5, we summarize our approach in relation to the state-of-the-art techniques.

2.1 DYNAMICS OF ROBOTIC GRASPING

The *dynamics* of a hand-object system describes how the finger moves in response to the joint actuator forces, and how the object moves in response to the grasping forces applied by the fingers. Being able to shape the dynamics is pivotal for a general-purpose grasping system to either keep the grasped object stable or move it to a desired configuration. In this section, the dynamics of robotic grasping is formulated and we follow notations in (Murray et al., 1994).

Given a hand-object system, as shown in Fig. 2.1, the object is grasped by a multifingered hand with N fingers and n_f degrees of freedom, the dynamics of the whole system is obtained by combining the dynamics of the hand (Eq. (2.2)) and the object

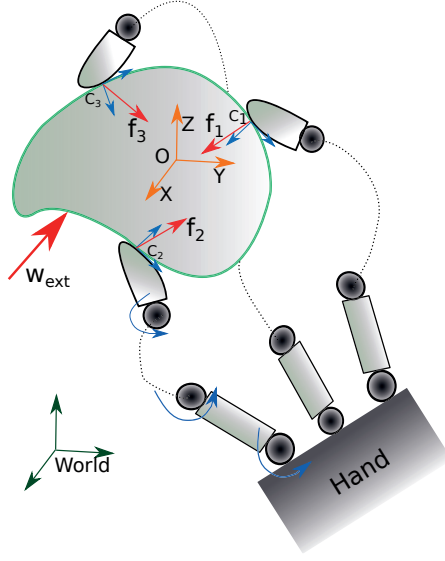


Figure 2.1: A typical hand-object system under external perturbation \mathbf{W}_{ext} . The contact forces \mathbf{f}_i are represented at i -th local contact coordinate frame C_i , which are considered as the control inputs from the object's perspective.

(Eq. (2.1)) via the grasp constraint (Eq. (2.3)) as follows,

$$\mathbf{M}_r(\mathbf{x}_r)\ddot{\mathbf{x}}_r + \mathbf{C}_r(\mathbf{x}_r, \dot{\mathbf{x}}_r)\dot{\mathbf{x}}_r + \mathbf{g}_r(\mathbf{x}_r) = \mathbf{w}_f + \mathbf{w}_{\text{ext}} \quad (2.1)$$

$$\mathbf{M}_h(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{C}_h(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \mathbf{g}_h(\boldsymbol{\theta}) = \boldsymbol{\tau} - \boldsymbol{\tau}_f + \boldsymbol{\tau}_{\text{ext}} \quad (2.2)$$

$$\text{subject to constraint: } \mathbf{J}\dot{\boldsymbol{\theta}} = \mathbf{G}^T\dot{\mathbf{x}}_r \quad (2.3)$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1^T, \dots, \boldsymbol{\theta}_N^T)^T$ is the vector of the generalized joint positions for N fingers and $\boldsymbol{\tau} \in \mathbb{R}^{n_f}$ contains the corresponding generalized actuator forces, which are considered as the control inputs for the hand. The vector $\mathbf{x}_r \in \mathbb{R}^{n_x}$ stands for the position and orientation for the object. The vector $\mathbf{w}_f \in \mathbb{R}^{n_x}$ describes the resultant object wrench from contact forces at fingertips and $\boldsymbol{\tau}_f \in \mathbb{R}^{n_f}$ represents the generalized joint torques due to contact forces at fingertips. The vector $\mathbf{w}_{\text{ext}} \in \mathbb{R}^{n_x}$ contains the generalized forces acting on the object and $\boldsymbol{\tau}_{\text{ext}} \in \mathbb{R}^{n_f}$ contains the generalized joint torques due to external forces acting on the finger links. The matrices $\mathbf{M}_r(\mathbf{x}_r) \in \mathbb{R}^{n_x \times n_x}$ and $\mathbf{M}_h(\boldsymbol{\theta}) \in \mathbb{R}^{n_f \times n_f}$ are the symmetric and positive definite inertia matrices of the object and hand, respectively. $\mathbf{C}_r(\mathbf{x}_r, \dot{\mathbf{x}}_r)\dot{\mathbf{x}}_r \in \mathbb{R}^{n_x}$ and $\mathbf{C}_h(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ contain the centrifugal and Coriolis terms, and $\mathbf{g}_r(\mathbf{x}_r) \in \mathbb{R}^{n_x}$ and $\mathbf{g}_h(\boldsymbol{\theta}) \in \mathbb{R}^{n_f}$ are the vectors of the generalized gravity forces, for the object and hand respectively. The notations are also given in Table 2.1. A detailed derivation of the dynamics can be found in (Siciliano and Khatib, 2008; Murray et al., 1994).

Note that in Eq. (2.3), the *grasp matrix* $\mathbf{G} \in \mathbb{R}^{n_x \times 3n_c}$ and the *hand Jacobian* $\mathbf{J} \in \mathbb{R}^{3n_c \times n_f}$ are of utmost importance. \mathbf{G} can be thought of as a mapping to transform contact forces to object wrench, i.e., $\mathbf{w}_f = \mathbf{G}\mathbf{f}$, while \mathbf{J} can be thought of as a mapping to transform contact forces to finger joint torques, namely $\boldsymbol{\tau}_f = \mathbf{J}^T\mathbf{f}$. In general, \mathbf{G} is related to the contact positions and normals, while \mathbf{J} is related to the configuration of

Table 2.1: Primary notations for grasp dynamics analysis

Notation	Definition
n_f	number of finger joints of hand
n_x	number of DOFs of object
n_c	number of contacts
$\boldsymbol{\theta} \in \mathbb{R}^{n_f}$	generalized positions for finger joints
$\mathbf{x}_r \in \mathbb{R}^{n_x}$	position and orientation of object
$\boldsymbol{\tau} \in \mathbb{R}^{n_f}$	generalized actuator forces of finger joints
$\boldsymbol{\tau}_f \in \mathbb{R}^{n_f}$	generalized joint torques due to contact forces at fingertips
$\boldsymbol{\tau}_{\text{ext}} \in \mathbb{R}^{n_f}$	generalized joint torques due to external forces on finger links
$\mathbf{w}_f \in \mathbb{R}^{n_x}$	resultant object wrench from contact forces at fingertips
$\mathbf{w}_{\text{ext}} \in \mathbb{R}^{n_x}$	external generalized forces acting on object
$\mathbf{M}_r(\mathbf{x}_r) \in \mathbb{R}^{n_x \times n_x}$	inertia matrix of object
$\mathbf{M}_h(\boldsymbol{\theta}) \in \mathbb{R}^{n_f \times n_f}$	inertia matrix of hand
$\mathbf{C}_r(\mathbf{x}_r, \dot{\mathbf{x}}_r) \dot{\mathbf{x}}_r \in \mathbb{R}^{n_x}$	centrifugal and Coriolis terms of object
$\mathbf{C}_h(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \dot{\boldsymbol{\theta}} \in \mathbb{R}^{n_f}$	centrifugal and Coriolis terms of hand
$\mathbf{g}_r(\mathbf{x}_r) \in \mathbb{R}^{n_x}$	generalized gravity forces for object
$\mathbf{g}_h(\boldsymbol{\theta}) \in \mathbb{R}^{n_f}$	generalized gravity forces for hand
$\mathbf{J} \in \mathbb{R}^{3n_c \times n_f}$	hand Jacobian
$\mathbf{G} \in \mathbb{R}^{n_x \times 3n_c}$	grasp matrix
$\mathbf{f} \in \mathbb{R}^{3n_c}$	vector of all contact forces at fingertips

the hand, i.e., $\boldsymbol{\theta}$. The computation of \mathbf{G} and \mathbf{J} requires accurate models of the fingers and the grasped object (Siciliano and Khatib, 2008).

In this thesis, only precision grasp and fine manipulation are treated, thus the contacts between object and hand are restricted to the fingertips and idealized as Point Contact with Friction (PCwF). To allow fine manipulation, usually at least three fingers are used, i.e., $N \geq 3$ and the object is grasped and manipulated in 3D-space, i.e., $n_x = 6$. Moreover, the constraint in Eq. (2.3) implies that no sliding or rolling occurs between fingertips and object. Note that the modeling of sliding and rolling are studied in the work of Kao and Cutkosky (1992), Montana (1988) and Trinkle (1989) to relax the grasp constraint. Nonetheless, accurate (geometric and dynamic) information of the fingers and object are still required in these models, making them unrealistic for practical use, especially when considering the variety of objects to be grasped and manipulated. For this reason, one of the main objective in this thesis is to relax the demand on precise object models by exploiting the uncertainties of the object.

To sum up, to allow precise analysis of the dynamics of grasping, the following assumptions are made in conventional approaches:

1. Accurate models of the fingers and the grasped object are known.
2. The contacts are restricted to the fingertips and are idealized as Point Contact with Friction (PCwF).
3. The relative contact positions between the object and fingertips do not change (ignoring the sliding and rolling effect).

With the dynamics of the grasping system, one of the most important objectives is to design the control inputs ($\boldsymbol{\tau}$ for the hand and \mathbf{f} for the object) to attain at least one

of the two desirable properties:

1. The ability to resist external forces \mathbf{w}_{ext} .
2. The ability to manipulate object, namely be able to control \mathbf{x}_r .

To achieve this, recalling Eq. (2.1), it is required to apply a proper object wrench $\mathbf{w}_f = \mathbf{G}\mathbf{f}$ to either stabilize or move the object, while the contact constraint Eq. (2.3) holds. However, we should notice that designing a controller that satisfies these conditions already goes beyond what current classic control approaches can provide: while the classic control theory relies on the controllability for given situations, here we need to first create the controllability by choosing proper locations for fingertips to construct an appropriate grasp matrix \mathbf{G} . As a consequence, in the study of robotic grasping, this problem is usually divided into two phases: *grasp planning* (selecting \mathbf{G}) and *grasp control* (regulating \mathbf{f}).

In the next three subsections, we will first briefly review the related work on grasp planning and control, with an emphasis on how they tackle the uncertainties. Then we describe related work on grasp learning and present how the learning approach can bridge the gap left by grasp planning and control, to improve the robustness of the whole grasping system. We conclude this chapter by a description of the pipeline of our approach.

2.2 GRASP PLANNING

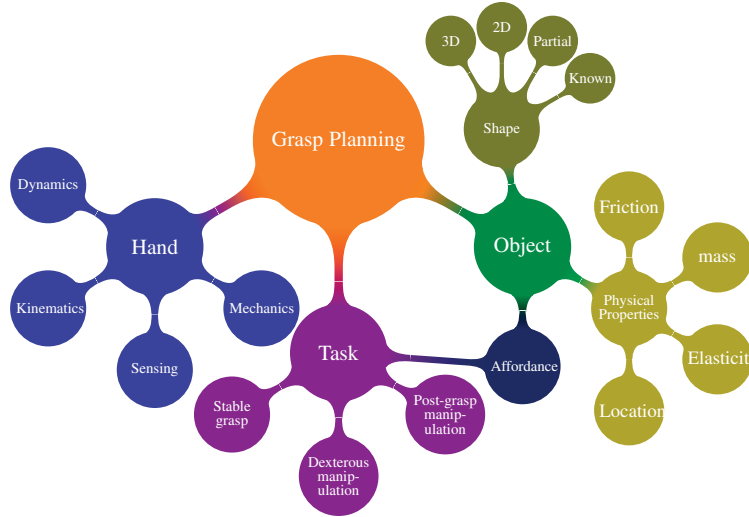


Figure 2.2: A number of aspects that influence how a grasp is planned for a given object. The most important one is the apriori object knowledge as will be discussed in Chapter 3, where in particular the shape of the object and the kinematics of the hand are addressed.

Napier, in his paper (Napier, 1956), distinguishes between power grasps and precision grasps. Power grasps lead to large areas of contact between the palm, the fingers,

and the object providing means of holding an object robustly into one’s palm. Precision grasps are particularly useful during manipulation since they offer more dexterity. In robotics, when a robot is provided with a hand mimicking the human hand dexterity, precision grasps become particularly relevant. Usually, these hands have a much larger degrees of freedom number than more classical robot grippers. Using such complex hands to generate a precision grasp requires determining *a position and an orientation for the hand and fingers* that are feasible, and that guarantee the grasp stability and task constraints. As shown in Fig. 2.2, a number of aspects are essentially involved in this process, mainly surrounded by the apriori knowledge of the object, task and hand. Obviously, for a grasp to be optimal, these aspects needs to be taken into account together. However, this is difficult to achieve due to the high-dimensionality of the grasping space and the non-linearity of the constraints, and therefore research in this area has often addressed only part of these factors and to deal with other aspects as known or unimportant information for planning.

Early work on grasp planning focused on finding the optimal contact points considering force closure as a grasp quality measure (Nguyen, 1987; Ferrari and Canny, 1992; Mirtich and Canny, 1994; Zhu and Ding, 2004). More recently, hand kinematics has been taken into account when estimating the feasible hand configuration for realizing the grasping points (Borst et al., 2002; Rosales et al., 2011). A drawback of this approach is that the valid hand configuration to realize the contacts may not be found. To address this issue, some studies first sample the hand pose around the object, and then apply different variants of the close-until-contact strategy to find the contact points (Miller et al., 2003; Berenson and Srinivasa, 2008; Li et al., 2013c; Miller and Allen, 1999; Goldfeder and Allen, 2011). However, these works require a large amount of heuristics to select the sampling strategy for the hand pose as well as the hand’s pre-shape before finger closing. Another alternative approach is to optimize the contact locations and the hand configurations simultaneously. Due to the high dimensionality of the problem, the optimization is conducted in a projected space of lower dimensionality using hand synergies (Santello et al., 1998) or eigen-grasps (Ciocarlie and Allen, 2009). In collaboration with S. El Khoury, we also developed an one-shot grasp synthesis approach that formulates the problem in the original hand configuration space and finds optimal finger joints and hand pose simultaneously if it exists (El Khoury et al., 2012, 2013). However, this is still computationally expensive and the obtained grasps are hand-dependent.

When considering uncertainty in grasp planning, one way is to incorporate robustness into the planning, preferring grasps that are somewhat insensitive to the uncertainty or search for stable graspable regions on the object (Zheng and Qian, 2005; Kim et al., 2013; Brost, 1985; Roa and Suarez, 2009; Christopoulos and Schrater, 2007). For instance, the concept of independent contact regions (ICRs) is introduced to provide robustness to finger placement error (Roa and Suarez, 2009) where any grasp with fingertip’s positions inside the ICR will achieve a force-closure grasp. These studies are usually conducted in the contact level and an additional step is still required to find a feasible hand configuration for realizing the grasping points. The uncertainty in-

formation can also be updated on-line using vision (Saut et al., 2014; Laaksonen et al., 2012) or tactile exploration (Dragiev et al., 2013; Sommer et al., 2014; Bjorkman et al., 2013). However, these approaches usually use a set of predefined grasps and require several rounds of grasp trials.

Even though planning with uncertainty can potentially improve the robustness of the final executed grasps, it is virtually an open-loop system as the planned grasps will never be corrected during execution, due to the computational complexity of these approaches. Thus, a planned stable grasp may end up to be unstable if the condition used for planning is different from the one in real-world scenarios. More importantly, grasps are also affected by factors that are difficult to be considered in the planning stages (e.g. the object weight is changed, the friction at contacts is different, the external perturbation occurs, etc.). Therefore, grasp planning alone is not sufficient to achieve a robust grasping system.

2.3 GRASP CONTROL

Grasp control refers to the problem of controlling the motion of an object by constraining its dynamics through contact with the hand (Prattichizzo, 2014). Research on grasp control has mainly focused on the design of control algorithms for achieving the desired grasping force that can either resist the external disturbance (robust grasping) or exert particular forces on the object to control its motion (dexterous manipulation), as mentioned in Section 2.1. The approaches can be classified into two groups depending on whether the contact force is explicitly controlled or not (Yoshikawa, 2010). One is the hybrid position and force control (including the grasping force control) (Li et al., 1989; Yoshikawa and Zheng, 1993; Buss et al., 1996; Zheng et al., 2012) which controls the positions in some directions and the force in other directions simultaneously. The other is impedance control which regulates the contact forces implicitly by specifying the desired impedance (Schneider and Cannon, 1992; Wimbock et al., 2008; Tahara et al., 2012; Li et al., 2014b). In general, all these control techniques require an accurate model of the system and a full state estimation, both of which are difficult to obtain in real hand-object system, as explained in Section 2.1. A more detailed review on these control techniques and their limitations has been discussed in Yoshikawa (2010) and Wimböck et al. (2012).

While the above mentioned grasp control focuses more on the stage when the fingers are already in contact with the object, another control stage that received less attention is the process for finger approaching and closing. Previous researches usually assume that a position controller is first used for approaching and then it is switched to a force controller once contact or collision is detected (Takahashi et al., 2008; Romano et al., 2011). However, the switching may easily lead to overshooting of the contact force due to the limitation in sensing and low control rate. This problem becomes more serious if there are geometric uncertainties of the object (location, shape, etc.), as the finger may very likely move or even knock over the object during finger closing.

To deal with uncertainty in grasp control, one typical solution is to use sensory

feedback to perform grasp adjustment locally so as to find stable grasps near the original planned grasp (Dang and Allen, 2013; Hsiao et al., 2010; Felip and Morales, 2009). For instance, Hsiao et al. (2010) proposes a set of simple and efficient heuristics to reactively correct the alignment error of the PR2 gripper. In Felip and Morales (2009), a sensor-based grasp primitive of Barrett hand is developed to adapt to the variation of the task conditions. These methods are usually reactive using actual sensing data from force or tactile sensors. The reactive correction strategy is designed to alleviate the need for precise hand-object pose information and hence can be more robust to pose and location uncertainty. The main problem of these methods is that, to design the corrective strategy, the grasp is usually limited to a predefined set of grasp primitives (Felip and Morales, 2009) or only simple hand kinematics is considered (Hsiao et al., 2010). For a more complex dexterous hand with a possibility to execute a large variety of grasps, it becomes more difficult to design such a corrective strategy.

Moreover, the strategy of performing local grasp adjustment can only be effective when the uncertainties mainly come from object’s pose or location. Another more common and important uncertainties in grasp control are uncertainties from the object physical properties, such as coefficient of friction, object weight and center of mass, or even external disturbance, that is, the parameters of object dynamics in Eq. (2.1). For these uncertainties, it will be more efficient to first adapt the grasping forces or impedance (\mathbf{f} in Eq. (2.1) rather than to change the grasp configuration (\mathbf{G} in Eq. (2.1) directly. Even though there are a body of studies on the control the grasping forces as mentioned above, in general it is still difficult to precisely control the grasping forces on the fingertips due to the uncertainty from finger dynamics and object geometry (Nguyen and Perdereau, 2013), even with the recent advanced tactile sensors such as BioTac (Wettels et al., 2008). As an alternative, the grasping forces can be regulated implicitly by varying the grasp impedance. However, it is considered as a difficult problem to select a proper impedance for a given task (Siciliano et al., 2008), let alone to adapt the impedance dynamically to stabilize the grasp or manipulate the object.

2.4 GRASP LEARNING

Grasp learning studies techniques allowing a robot to acquire grasp strategies in terms of planning and control through learning algorithms. As discussed above, grasp planning is still computational expensive, impeding its power in uncertain environments where fast re-planning is required, and therefore mostly it is implemented and executed as an open-loop. On the other hand, grasp control can leverage the rich sensory feedback to close the loop, but it requires a detailed model of the system and a full state estimation, which are considered hard to obtain for a grasping system in real-world scenarios. Also, it is a challenge to design an adaptation strategy for a dexterous hand under various uncertainties. To address these issues, one of the promising solutions is to take advantage of learning approach, which can essentially contribute to both planning and control. More importantly, it can unify grasp planning and control as a single framework, resulting in a closed-loop system.

2.4.1 LEARNING IN GRASP PLANNING

Learning-based approaches have been studied extensively in grasp planning and most of these use data-driven model to learn “rules” between object features and feasible hand configurations (J. Bohg and Kragic, 2014). In (Miller et al., 2003; Ekvall and Kragic, 2007; Huebner and Kragic, 2008), objects are represented as basic shape primitives and then associated with predefined grasp primitives. Pelossof et al. (2004) use a support vector machine (SVM) to learn the grasp quality manifold for a specific hand and simple object shapes. The manifold represents the mapping from grasp parameters and object shape to the grasp quality and new optimal grasps are found through interpolation on the manifold. Most of these methods are either limited to basic shapes or simple grasp primitives and cannot be used to execute a set of specific contact locations. Along this direction, Huang et al. (2013) learns the joint density function of hand pose and finger joints from a large set of grasps. This density function is used later to retrieve finger joints on-line given any query hand pose. However, learning is conducted in the object frame and with specific hand-object combination. As a result, a new learning model is required for each new pair of hand-object combination. Kopicki et al. (2014) learn two separate models, i.e., the contact model to express the relationship between fingers and object local features, and the hand configuration model to represent whole hand configuration during approach to grasp. They show this approach can generalize to new objects for given grasp types.

Instead of learning the rules directly with robotic hands, another pipeline attempts to transfer the grasps from human hands to robotic hands, by taking advantage of the human knowledge of grasping (Cutkosky and Howe, 1990; Kang, 1994). One of the main issue for this pipeline is known as “correspondence problem” in the imitation learning literature (Billard et al., 2008; Alissandrakis et al., 2007), which does not permit a direct correlation in the joint level due to the difference between human and robotic hand in terms of kinematic structure. To address this, one solution is to extract the grasp primitives or intentions from human demonstration (De Souza et al., 2015; Ekvall and Kragic, 2004; Cutkosky and Howe, 1990; Kang, 1994), assuming that the grasps of human and robot share the same intention. If some key grasping points can be specified beforehand, the grasp can be also transferred at the contact level (Lin and Sun, 2014; Salvietti et al., 2014; Rosales et al., 2011). Considering the goal of grasps, recent studies attempt to learn task requirements from human grasp demonstration, which are then encoded as grasp quality metrics for robotic grasp planning (Lin and Sun, 2015; El-Khoury et al., 2015; Song et al., 2015).

Despite all these progresses, the main bottleneck still exists when applying learning-based approaches to grasp planning, that is, how to generalize the learned grasps across different hands and objects, under various task constraints. In our approach, we follow a similar principle as Kopicki et al. (2014). With the help of Virtual Frame (Tahara et al., 2010), planning of grasping points and learning of hand inverse kinematics are conducted independently, thus allowing for different hands to be used and making it also possible for different contact level grasp planners to be used in the system.

2.4.2 LEARNING IN GRASP CONTROL

When integrating learning with grasp control, not only can it alleviate the need for accurate system model and full state estimation, but also help to devise more intelligent grasp adaptation strategy. The learning approaches usually can be directly applied to learn the dynamics of the finger (Yun and Deshpande, 2014), which is similar as the case for a single manipulator (Nguyen-Tuong et al., 2009), if we consider each finger as a small arm. A more peculiar issue in grasp learning is to estimate the states, especially the states of the object. Numerous studies have been conducted in this area, in order to use different perception techniques to extract the object state from sensory information. For instance, the in-hand object can be modeled on-line using depth camera (Krainin et al., 2011), or using tactile exploration (Sommer et al., 2014; Strub et al., 2014; Dragiev et al., 2013; Bimbo et al., 2015), or by combining both (Bjorkman et al., 2013; Rosales et al., 2014). Given initial object pose and the geometry of the object, the pose of the object can be locally estimated on-line using proprioception and tactile information of the hand (Chalon et al., 2013; Haidacher and Hirzinger, 2003). The frictional property of the object surface can be also obtained from tactile sensing at the fingertips (Liu et al., 2015; Rosales et al., 2014; Liu et al., 2011). In spite of these progresses, it is still a tough problem to obtain robust object states estimation that can be used directly for grasp control in real-world scenarios (Ritter and Haschke, 2015; Yousef et al., 2011; Yoshikawa, 2010).

Rather than use the parameters of object model explicitly in grasp control, recent researches attempt to seek for holistic solution in which the object parameters are used implicitly. This is usually implemented in the form of reactive control that couples the sensory outputs with the actuator inputs directly, to allow the hand to response quickly to changing and uncertain environments. One of the conventional approaches is to detect the “incipient slippage” and adapt the grasping forces accordingly to prevent from complete object slipping (Takahashi et al., 2008; Kanno et al., 2013; Romano et al., 2011). However, this solution actually only detects the stability at contact level. Another more important meaning for “grasp stability” refers to the tendency of the grasped object to return to an equilibrium location when the object position is perturbed (Montana, 1991; Nakamura et al., 1989; Xiong et al., 1999). To this end, Bekiroglu et al. (2011, 2013) adopt a statistic learning approach to assess the grasp stability from tactile sensing and joint configuration of the hand. Based on a similar stability estimator, Dang and Allen (2013) present a hand adjustment algorithm to optimize the hand pose locally to achieve a stable grasp. However, only object pose uncertainty has been considered in this work and it requires to re-open all the fingers to adjust the hand pose, which greatly limits its generality to dynamic uncertainties. In (Sauser et al., 2012), the authors learn a probabilistic model of the relationship between the finger joints and the tactile responses on the fingertips, which allows a fast adaptation of the finger joints given the change of the tactile sensing due to external perturbation. More recently, based on an object-level impedance controller (Tahara et al., 2012; Li et al., 2014b), Li et al. (2014a) propose a dynamic grasp adaptation strategy to maintain grasp stability

by varying the grasping force and fingertips locations in-hand under various physical uncertainties. Besides the grasp stability, [Paolini et al. \(2014\)](#) present a statistical model to encode post-grasp (task) requirements, which allows to choose actions to maximize the likelihood of successful execution of tasks, rather than to keep the grasp stability.

Notwithstanding these recent achievements, it is still, to a large extent, an open question to learn a grasp control strategy to stabilize a grasp under various uncertainties. We should also note that most of the learning-based grasp control techniques mentioned above aim for robust grasping. Relative few work has been done to consider that how uncertainty affects post-grasp manipulation (e.g. constrained object manipulation), and how to design a control policy to react to these uncertainties while maximizing the task achievements. In this thesis, we propose a dynamic grasp adaptation strategy based on a statistical model of the previous grasp experience that is supervised by a human demonstrator, which can deal with a large variety of uncertainties, such as changing object weight, unprepared perturbation, etc. For constrained object manipulation, in addition to the grasp stability requirement, we further define an adaptation strategy that is consistent with the task requirement, in order to improve the task performance.

2.5 OUR APPROACH

As discussed above, both grasp planning and control can greatly benefit from the integration with learning-based approach, especially in uncertain environments. However, it is still difficult to make the grasp planning efficient and generalizable, and to design a grasp control strategy that is robust and adaptive. In addition, the gap between grasp planning and control remains a key challenge. To become part of the solution, this thesis develops a unified framework, taking advantage of the learning-based approach encoded in an object-centric frame, to bridge the gap left by grasp planning and control, and thus enables robot hand to adapt adequately to various uncertainties.

To be more specific, during the entire process of grasping, as shown in Fig. 2.3, we focus on the dominant uncertainties of each stage, which are cooperatively addressed by grasp planning and control. This requires a kind of “uncertainty distribution” between planning and control. To this end, the apriori knowledge of the uncertainties, if available, will be encoded in the form of statistical models, which can be later incorporated into the planning algorithm; Moreover, this apriori knowledge together with the planned grasps, are propagated to the grasp control, to allow the controller to react more robustly and adaptively to the uncertainties. Below we explain how we address different types of uncertainties during the process of robotic grasping.

- **Grasp Planning:** The main uncertainty in this stage (i.e., shape uncertainty) is parametrized by Gaussian Process Implicit Surface, which is formulated as a constraint in a contact level grasp planning algorithm. Moreover, a probabilistic hand inverse kinematic model is proposed to estimate feasible hand configurations that can realize the planned grasp.

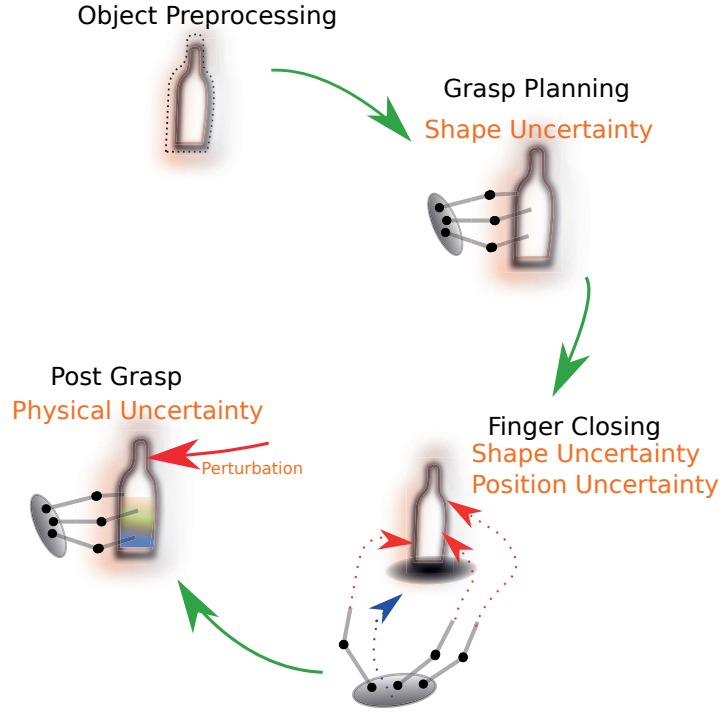


Figure 2.3: During the process of robotic grasping, different types of uncertainties of the object to be grasped (manipulated) become dominant for each stage, i.e., grasp planning, finger closing and post-grasp. Note that in the second stage, the reaching motion is not addressed in this paper, which is mainly influenced by position uncertainty.

- **Finger Closing:** This stage actually consists of two steps: the hand approaching and finger closing. The former one is a large research area in motion planning studies, which is not included in this work. During the finger closing, the uncertainties from object shape and desired contact force are dominant, and thus are used to devise an uncertainty-aware control strategy to close the finger compliantly. Note that this step is of great importance, as it settles down the pre-conditions for the post-grasp stage.
- **Post-Grasp:** Depending on the task, the goal in the post-grasp stage can be either robust grasping (i.e. keep object stable in-hand) or constrained object manipulation (i.e. exert force on object to move it or interact with environment). For both tasks, the physical uncertainties (including external perturbations) can greatly affect the task performance. For robust grasping, a dynamic grasp adaptation strategy is proposed based on a statical model of the grasp experience in terms of tactile sensing and control parameters, which can react to the physical uncertainties by either changing the grasping forces at fingertips or relocating the position of fingertips in-hand. For constrained object manipulation, the task requirement is encoded as a manifold in the planning stage, which allows a fast task-consistent adaptation in terms of varying the desired reference trajectory of object, to deal with uncertainties from imprecise contact dynamics and other

external perturbations acting on the object.

Following the grasping process, this thesis is organized as follows. In Chapter 3, we will address the shape uncertainty during grasp planning and finger closing. The physical uncertainties (change in object's weight and external perturbations) during post-grasping will be addressed in Chapter 4 to achieve robust grasping. Chapter 5 further extends the grasp adaptation approach to constrained object manipulation, in order to deal with uncertainties from imprecise contact dynamics and other external perturbations acting on the object.

DEXTEROUS GRASPING UNDER SHAPE UNCERTAINTY

3.1 INTRODUCTION

In this chapter, we focus on achieving dexterous grasping under object shape uncertainty. More precisely, we propose a method that takes explicitly the amount of shape uncertainty into account during both grasp planning and grasp execution to obtain desired grasp configuration. The approach consists of:

1. The object shape and its uncertainty is parametrized using Gaussian Processes Implicit Surface (GPIS), and explicitly taken into account during a contact-level grasp planning.
2. A probabilistic hand inverse kinematics model, which is learned off-line in the object frame, is proposed to find the feasible hand configurations to realize the planned grasping points.
3. During the grasp execution, a compliant finger closing controller is devised to adaptively place the fingertips on the desired locations.

The novelty of our approach will be detailed during problem formulation in the following section. In general, the above-mentioned item 1 and 3 mainly aim at taking the uncertainty of object shape into account, while the item 2 is mainly to alleviate the need for solving non-linear constraints of hand inverse kinematics. Note that the work presented here are published as joint papers with Kaiyu Hang and Danica Kragic at KTH ([Li et al., 2015a](#)), and with Sahar El-Khoury ([El Khoury et al., 2013, 2012](#); [Li et al., 2013a](#)), a former colleague at EPFL. This chapter reports only on the parts that were developed by myself.

Related Publications:

- ([Li et al., 2015a](#)) Li, M., Hang, K., Kragic, D., and Billard, A. (2015a). Dexterous grasping under shape uncertainty. *Robotics and Autonomous Systems*, 75:352–364
- ([El Khoury et al., 2013](#)) El Khoury, S., Li, M., and Billard, A. (2013). On the generation of a variety of grasps. *Robotics and Autonomous Systems*, 61(12):1335–1349

- (El Khoury et al., 2012) El Khoury, S., Li, M., and Billard, A. (2012). Bridging the gap: One shot grasp synthesis approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2027–2034
- (Li et al., 2013a) Li, M., El Khoury, S., and Billard, A. (2013a). Synergy-level grasp synthesis learning. In *ICRA 2013 Workshop: Hand synergy-how to tame the complexity of grasping*

3.2 PROBLEM STATEMENT

Given an object, grasp planning refers to the problem of finding a grasp configuration that satisfies a set of criteria that is relevant for the grasping task. The term *grasp configuration* has two different meanings in the literature, which entails defining the configuration of the fingers either from the standpoint of the hand (determining the finger joints and hand pose) or from the standpoint of the object (determining the position of the fingers on the object). As mentioned in Section 2.2 and shown in Fig. 2.2, a large number of aspects are involved in grasp planning (e.g., object, hand, task), making the problem of grasp planning difficult to solve in general. In the remainder of this section, we will first give a general formulation of grasp planning as an optimization problem in Section 3.2.1. Then we point out the challenges to solve this problem and summarize the related work in Section 3.2.2. Our approach is depicted in Section 3.2.3.

3.2.1 GRASP PLANNING AS OPTIMIZATION

As mentioned in Section 2.2, grasp planning requires to take the constraints from the hand, object and task into account simultaneously, which ultimately can be formulated as an optimization problem in general, as shown Fig. 3.1. In this framework we assume the objective function as well as all the constraints can be all represented analytically. The objective function $f(\boldsymbol{\theta}, \mathbf{H}, \mathbf{p}, \mathbf{n})$ can be any scalar function of the hand pose and finger joints, such as the quality of force closure (Roa and Suárez, 2014; Ferrari and Canny, 1992) and the sum of joint torques (El Khoury et al., 2013; Wenyu et al., 2010; Li et al., 2013b). The hand constraints consist of the hand forward kinematics to compute the positions and normal directions of fingertips. These constraints are generally non-linear, and a set of such constraints need to be considered due to the large number of degrees of freedom of the dexterous hand, as shown in Fig. 3.2. The object constraints mainly restrict the location of the fingertips to be consistent with the local object surface. Different object representations (see Fig. 3.2) can be used for this purpose, such as point cloud (Hang et al., 2014b; Roa and Suarez, 2009), polyhedron (Ponce et al., 1997; Ding and Wang, 2001), superquadrics (El Khoury et al., 2012) and GPIS (Li et al., 2015a; El Khoury et al., 2013; Dragiev et al., 2011). Finding a proper representation of the task constraints, i.e., $Q_{\text{task}}(\boldsymbol{\theta}, \mathbf{H}, \mathbf{p}, \mathbf{n}) \in \mathcal{G}_{\text{task}}$ in grasp planning is still, to a large extent, an open question and many ongoing work are conducted in this area (Song et al., 2015; El-Khoury et al., 2015; De Souza et al., 2015). Given this grasp planning framework, the difference of previous studies in grasp planning largely

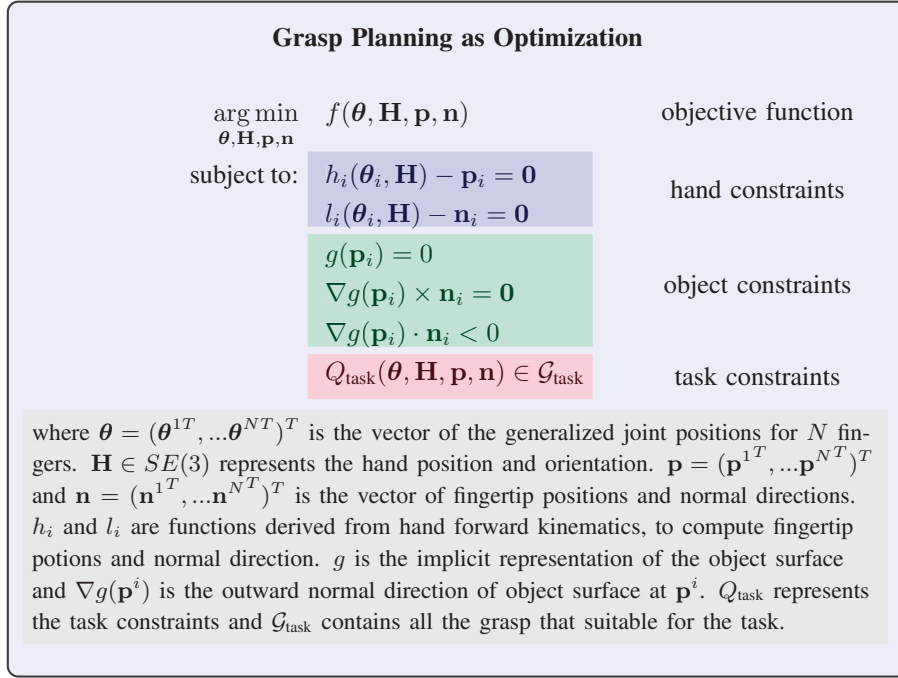


Figure 3.1: A general framework of grasp planning as optimization.

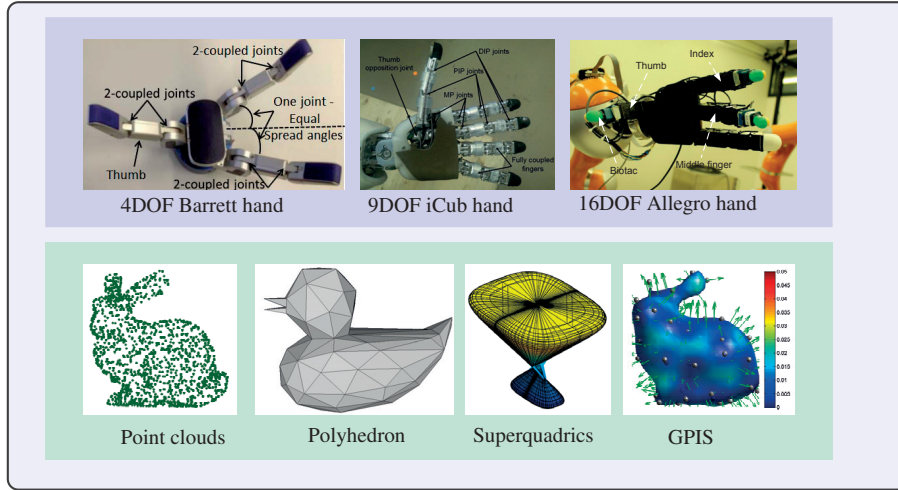


Figure 3.2: Typical examples of dexterous hands (Barrett, iCub and Allegro hand) and object representations (point clouds, polyhedra, superquadrics and GPIS) used for grasp planning.

boils down to the different choices of the constraints and objective function. A thorough survey of grasp planning can refer to (Bicchi, 2000; Sahbani et al., 2012; J. Bohg and Kragic, 2014; Goldfeder and Allen, 2011).

3.2.2 CHALLENGE

In this chapter, we only target the hand and object constraints and their corresponding

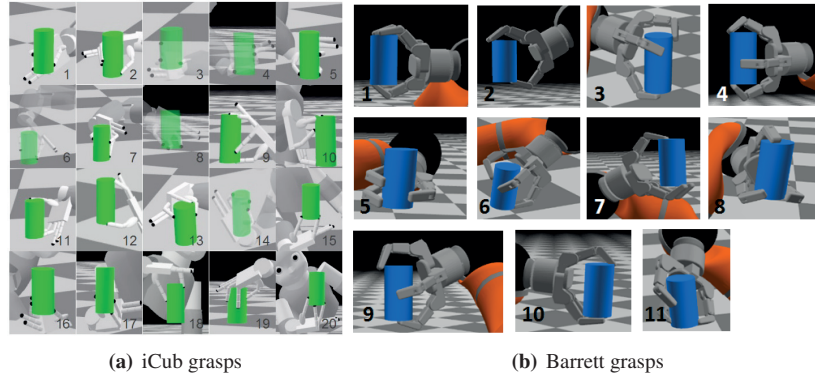


Figure 3.3: The different hand configurations of iCub hand and Barrett hand to grasp a cylinder.

difficulties. In Section 2.2, we presented a number of techniques to handle the hand constraints and object constraints. As we have highlighted, because of the highly non-linearity of the hand constraints, the optimization can be computationally expensive or even infeasible. Moreover, the obtained grasps are hand-dependent, which means the obtained grasp can not be transferred among different robotic hands. Figure 3.3 shows several different hand configurations of iCub hand and Barrett hand to grasp a cylinder, as reported in (El Khoury et al., 2013). It is obvious to notice that some of the grasps share the same grasping points, for example, iCub grasp 1 and Barrett grasp 5. It will be desirable if the same grasping points can be realized by different robotic hands, which enables the possibility to share grasps among different hands. To achieve this, an efficiently solver for the hand inverse kinematics is required to check whether the given grasping points are feasible for the adopted hand or not. In our work, we propose a probabilistic hand inverse kinematics model to quickly find the most likely hand configurations to realize the grasp if it is feasible.

So far, all the constraints in the grasp planning framework are assumed known precisely. Another more challenging issue comes from the uncertainties associated with these constraints, especially the uncertainties from the object side (e.g., object pose, location and shape, etc.). Because of these uncertainties, a planned stable grasp can end up being unstable (Goldfeder and Allen, 2011; Ciocarlie et al., 2010; Kim et al., 2013). As discussed in Section 2.2, considering uncertainty in robotic grasping has become increasingly important and a large number of techniques have been proposed. A comparative study of these works is summarized in Table 3.1. Few works have considered object shape uncertainty by integrating planning and control. However, in real robotic grasping tasks due to the, for example, occlusion problems (Bohg et al., 2011; Gori et al., 2014; Bjorkman et al., 2013) or non-reachability from tactile exploration (Sommer et al., 2014), object shape uncertainty is inevitable. Therefore, the second novelty of the work in this chapter is to explicitly reason about the object shape uncertainty by integrating grasp planning and control.

Table 3.1: A brief summary of grasping under uncertainty. **Uncertainty type:** The type of uncertainty has been studied; **Rep.:** The uncertainty is represented implicitly (not parametrized) or explicitly (parametrized); **Hand:** The type of robotic hand has been considered; **Planning:** The uncertainty has been considered during the planning; **Control:** The uncertainty has been considered during the control stage and the corresponding strategy used.

Paper	Uncertainty type	Rep.	Hand	Planning	Control
(Dang and Allen, 2013; Felip and Morales, 2009)	Object pose	Implicit	Barrett hand	No	Hand adjustment
(Hsiao et al., 2010)	Object pose and shape	Implicit	PR2 gripper	Grasp primitives	Heuristic
(Dragiev et al., 2013; Bjorkman et al., 2013)	Object shape	Explicit	Dexterous hand	Grasp primitives	iterative exploration
(Saut et al., 2014)	Object pose	Explicit	Dexterous hand	Grasp Primitives	No
(Roa and Suarez, 2009)	Contact location	Explicit	No	Yes	No
(Rosales et al., 2011)	Contact location	Explicit	Dexterous Hand	Yes	No
(Christopoulos and Schrater, 2007)	Contact location and shape	Explicit	No	Yes	No
(Takahashi et al., 2008; Li et al., 2014a)	Object mass and friction	Implicit	Dexterous hand	No	Finger adaptation
Our work	Object shape	Explicit	Dexterous hand	Yes	Finger adaptation

3.2.3 OUR APPROACH

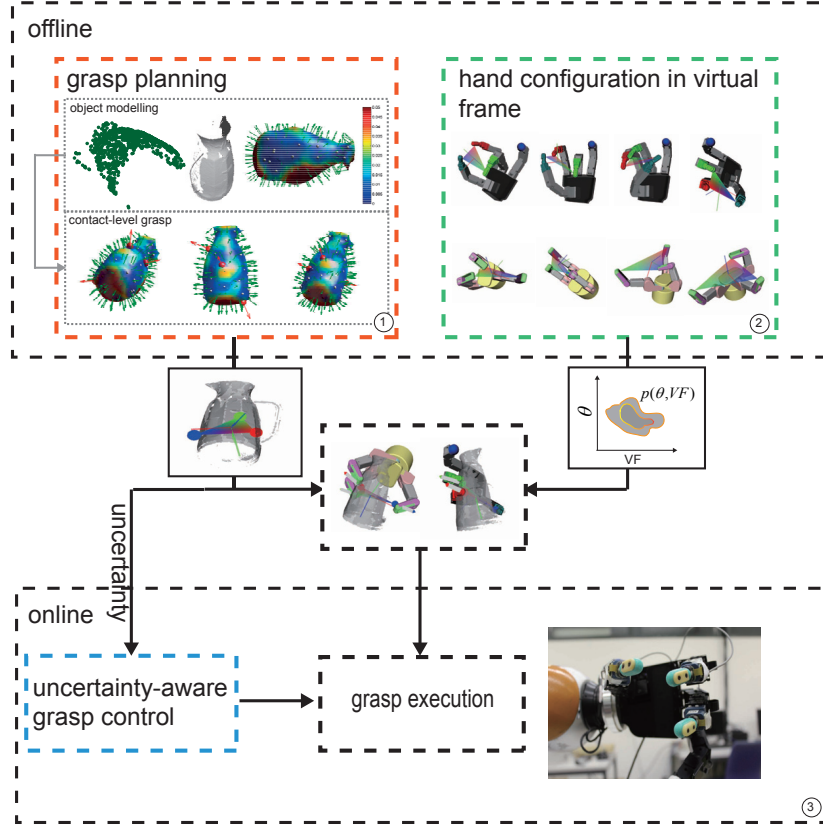


Figure 3.4: The overview of the proposed approach.

- ①: The contact-level grasp planning with shape uncertainty. The output is a set of contacts defining a grasp with associated shape uncertainty these can withstand.
- ②: The probabilistic model for the hand inverse kinematics is learned offline and is frame invariant.
- ③: Given the desired grasping points and the employed hand, the corresponding hand configuration is obtained in real time.
- ④: The obtained hand configuration and the uncertainty information are passed to the controller for compliant grasp execution.

In this chapter, We present a system dealing with object shape uncertainty which may originate from occlusion, partial view or issue with sensor calibration. Shape uncertainty is parametrized using Gaussian Processes (GP) and it is incorporated as a constraint into a contact-level grasp synthesis algorithm. The output of the algorithm is a set of contacts defining a grasp with an associated shape uncertainty that determines the maximum uncertainty a grasp can withstand, as shown in the left upper part of Fig. 3.4(1). Given the desired grasping contacts, the feasible hand configuration (hand pose and finger joint configuration), is computed using a learned probabilistic model. The probabilistic model is learned off-line for each hand and is frame invariant due to the use of a *Virtual Frame* (VF) approach. The learned model is hence, independent of the choice of hand and object frame. VF relies on a set of parameters defined to encode

grasps as shown in the right upper part of Fig. 3.4(2). Since a grasp is first planned in the object frame by generating a set of contact locations, it is not dependent on a specific hand design. Similarly, the learned probabilistic model for the hand inverse kinematics is not constrained by object shape. Therefore, given a new hand with its learned probabilistic model, the corresponding hand configuration that matches the generated grasping contacts can be obtained in real time. During grasp execution, a compliant finger closing scheme is devised. A parallel position/force (tactile) controller is implemented by exploiting the uncertainty in object shape and contact force based on our previous work (Li et al., 2014a). An overview of the system is shown in Fig. 3.4.

3.3 GRASP PLANNING UNDER SHAPE UNCERTAINTY

In this section, we begin by presenting the GP for object shape modeling as well as the parametrization of associated shape uncertainty. Thereafter, a shape uncertainty constrained grasp planning approach is presented. For clarity of the presentation, the notations adopted in this chapter are summarized in Table 3.2.

Notation	Definition
$\mathbf{x} \in \mathbb{R}^3$	a point in \mathbb{R}^3
$d \in \mathbb{R}$	signed distance
$\boldsymbol{\omega} \in \mathbb{R}^3$	normal direction
$\mathbf{y} = \{d, \boldsymbol{\omega}\} \in \mathbb{R}^4$	output of GP
$cov(\mathbf{y}^i, \mathbf{y}^j) \in \mathbb{R}^4$	covariance between \mathbf{y}^i and \mathbf{y}^j
$E(\mathbf{y}_*) \in \mathbb{R}^{4 \times 1}$	predicted output of GP
$cov(\mathbf{y}_*) \in \mathbb{R}^{4 \times 4}$	the covariance with the prediction
$f_{cov}(\mathbf{x}_*)$	shape uncertainty, i.e., $[cov(\mathbf{y}_*)]_{11}$
$\mathbf{p}^i \in \mathbb{R}^3$	position of contact point
$\mathbf{n}^i \in \mathbb{R}^3$	normal direction at contact point
S_{thresh}	threshold on the shape uncertainty
$\mathbf{w}_i^j \in \mathbb{R}^6$	contact wrench
ϕ_j^i	coefficient of contact wrench
$\mathbf{p}^o \in \mathbb{R}^3$	origin of virtual frame
$R^o \in \text{SO}(3)$	orientation of virtual frame
$\Theta \in \mathbb{R}^h$	finger joints
$L \in \mathbb{R}^3$	distance between each fingertip and \mathbf{p}^o
$N \in \mathbb{R}^3$	pairwise inner product of \mathbf{n}^i
π_i	prior of the i th Gaussian component
$\mathcal{N}(\mu_i, \Sigma_i)$	Gaussian distribution
$T_{Hand}^{Obj} \in \mathbb{R}^{4 \times 4}$	hand pose in object frame
$\mathbf{x}_d \in \mathbb{R}^3$	desired fingertip position
$\mathbf{f}_d \in \mathbb{R}^3$	desired contact force
$\mathbf{K}_P \in \mathbb{R}^{3 \times 3}$	position control gain
$\mathbf{C}_F \in \mathbb{R}^{3 \times 3}$	force control gain

Table 3.2: List of notations.

3.3.1 OBJECT MODELING FOR GRASPING

For an unknown object, due to the limited viewing angle or occlusion, it may be difficult for a robot to observe its complete shape. The unseen parts of the object may be important from a grasping perspective as these may provide better contact locations. To synthesize contacts on unseen parts, we propose to model the whole object surface using GPs. We denote by $\mathbf{x} \in \mathbb{R}^3$ an arbitrary point with normal direction $\boldsymbol{\omega} \in \mathbb{R}^3$, and by $d \in \mathbb{R}$ the relative position of \mathbf{x} with respect to the object surface. We define a function

$$g(\mathbf{x}) : \mathbb{R}^3 \rightarrow \mathbb{R}^4 \quad (3.1)$$

that maps the position of a point to its relative position and its outward normal direction as the basis for estimation of the object shape. In particular, the relative position $d = 0$ when the point is on the object, and $d \in \mathbb{R}^-$ or $d \in \mathbb{R}^+$ when the point is inside or outside of the object respectively.

For training the GP model of the object, the input is a training dataset denoted $\mathbf{X} = \{\mathbf{x}^i \in \mathbb{R}^3\}_{i=1 \dots n_t}$, composed of points on the object surface. It originates from point clouds of partially viewed object, consisting also of points inside and outside the object surface. The latter two kinds of points are included to increase the accuracy of the GP estimation (Rasmussen and Williams, 2005). In practice, we first normalize the points on the object surface to range $[-1, 1]$. The origin is then selected as the interior point for training with $d = -1$. For the outside points, 20 points are randomly sampled from a sphere with radius 1.2 with $d = 1$. The output of our training dataset is $\mathbf{Y} = \{\mathbf{y}^i = (d^i, \boldsymbol{\omega}^i) \in \mathbb{R}^4\}_{i=1 \dots n_t}$. The covariance between two outputs $cov(\mathbf{y}^i, \mathbf{y}^j) \in \mathbb{R}^4$ is defined as:

$$cov(\mathbf{y}^i, \mathbf{y}^j) = \begin{bmatrix} cov(d^i, d^j) & cov(d^i, \boldsymbol{\omega}_1^j) & cov(d^i, \boldsymbol{\omega}_2^j) & cov(d^i, \boldsymbol{\omega}_3^j) \\ cov(\boldsymbol{\omega}_1^i, d^j) & cov(\boldsymbol{\omega}_1^i, \boldsymbol{\omega}_1^j) & cov(\boldsymbol{\omega}_1^i, \boldsymbol{\omega}_2^j) & cov(\boldsymbol{\omega}_1^i, \boldsymbol{\omega}_3^j) \\ cov(\boldsymbol{\omega}_2^i, d^j) & cov(\boldsymbol{\omega}_2^i, \boldsymbol{\omega}_1^j) & cov(\boldsymbol{\omega}_2^i, \boldsymbol{\omega}_2^j) & cov(\boldsymbol{\omega}_2^i, \boldsymbol{\omega}_3^j) \\ cov(\boldsymbol{\omega}_3^i, d^j) & cov(\boldsymbol{\omega}_3^i, \boldsymbol{\omega}_1^j) & cov(\boldsymbol{\omega}_3^i, \boldsymbol{\omega}_2^j) & cov(\boldsymbol{\omega}_3^i, \boldsymbol{\omega}_3^j) \end{bmatrix} \quad (3.2)$$

To compute all the entries in the covariance matrix, the following identities are required, which can be obtained from the selected kernel function $k(\cdot, \cdot)$ and its derivatives (Solak et al., 2003):

$$cov(d^i, d^j) = k(\mathbf{x}^i, \mathbf{x}^j) \quad (3.3)$$

$$cov(\boldsymbol{\omega}_m^i, d^j) = \frac{\partial}{\partial x_m} cov(d^i, d^j) \quad (3.4)$$

$$cov(\boldsymbol{\omega}_m^i, \boldsymbol{\omega}_n^j) = \frac{\partial^2}{\partial x_m \partial x_n} cov(d^i, d^j) \quad (3.5)$$

$$m = 1, 2, 3, n = 1, 2, 3;$$

For the kernel function, a thin plate kernel (Bookstein, 1989) is adopted to regularize the first order continuity, assuming that the normal direction on the object surface

is continuous. Given two inputs, the thin plate kernel function is computed by

$$k(\mathbf{x}^i, \mathbf{x}^j) = 2\|\mathbf{x}^i - \mathbf{x}^j\|^3 - 3\psi\|\mathbf{x}^i - \mathbf{x}^j\|^2 + \psi^3 \quad (3.6)$$

where $\psi \in \mathbb{R}^+$ is the longest pairwise distance among all the training inputs. Note that ψ is the only parameter in the kernel function and it can be easily determined from the training inputs without any optimization procedure involved, such as maximizing the likelihood. This is one of the main reasons for adopting it here in comparison to other kernel functions such as Radial Basis Function (Dragiev et al., 2011, 2013).

Given the computed covariance matrix and a new data point $\mathbf{x}_* \in \mathbb{R}^3$, we can use GP to predict the function mean value $E(\mathbf{y}_*) \in \mathbb{R}^{4 \times 1}$ and its corresponding variance $cov(\mathbf{y}_*) \in \mathbb{R}^{4 \times 4}$, (Rasmussen and Williams, 2005) by

$$E(\mathbf{y}_*) = E([d_*, \boldsymbol{\omega}_*]^T)^T = K_*[K(\mathbf{X}, \mathbf{X}) + \sigma^2 I]^{-1} \mathbf{Y} \quad (3.7)$$

$$cov(\mathbf{y}_*) = K(\mathbf{x}_*, \mathbf{x}_*) - K_*[K(\mathbf{X}, \mathbf{X}) + \sigma^2 I]^{-1} K_*^T \quad (3.8)$$

where $K_* = K(\mathbf{x}_*, \mathbf{X})$ denotes the $4 \times 4n_t$ covariance matrix evaluated at all pairs of the testing and training inputs, and similarly for $K(\mathbf{X}, \mathbf{X}) \in \mathbb{R}^{4n_t \times 4n_t}$ and $K(\mathbf{x}_*, \mathbf{x}_*) \in \mathbb{R}^{4 \times 4}$. The parameter σ^2 reflects the variance of noise in the output.

Given Eq. (3.7), we can estimate if the point \mathbf{x}_* is on the object surface or not using $E([d_*])$ as well as predicting the normal direction at this point using $E([\boldsymbol{\omega}_*])$. From Eq. (3.8), we can compute the uncertainty of our prediction. In our grasp planning method presented in next section, we only consider the shape uncertainty, which is the first entry of $cov(\mathbf{y}_*)$, i.e., $[cov(\mathbf{y}_*)]_{11}$. As these three quantities only depend on the input \mathbf{x}_* , we use the following notations to represent each of them: $f_d(\mathbf{x}_*) = E([d_*])$, $f_\omega(\mathbf{x}_*) = E([\boldsymbol{\omega}_*])$ and $f_{cov}(\mathbf{x}_*) = [cov(\mathbf{y}_*)]_{11}$. Therefore, the final object shape is expressed as the function:

$$f_d(\mathbf{x}) = 0, \mathbf{x} \in \mathbb{R}^3 \quad (3.9)$$

3.3.2 GRASP PLANNING WITH SHAPE UNCERTAINTY CONSTRAINTS

In this section, we first describe how to explicitly incorporate shape uncertainties as constraints in grasp planning. We then proceed by explaining how to plan dexterous, three-fingered grasps by formulating planning as an optimization problem, as mentioned in Section 3.2.1. We denote a contact point and its corresponding normal direction as $\mathbf{p}^i \in \mathbb{R}^3$ and $\mathbf{n}^i \in \mathbb{R}^3$, $i = 1, 2, 3$.

Points on surface and normal alignment: To execute the desired grasp, a basic constraint is that the planned contact locations are on the object surface. With the GP representation of the object surface, Eq. (3.9), this can be expressed as:

$$f_d(\mathbf{p}^i) = 0, \quad i = 1, 2, 3; \quad (3.10)$$

Besides this constraint, the normal direction of the contact points should be aligned

with the object's surface normal which is represented as

$$\mathbf{n}^i = f_{\omega}(\mathbf{p}^i), \quad i = 1, 2, 3; \quad (3.11)$$

Shape uncertainty constraint: GP based object surface representation can be used to predict the shape uncertainty and it is taken into account in the grasp planning procedure as

$$f_{cov}(\mathbf{p}^i) < S_{\text{thresh}}, \quad i = 1, 2, 3; \quad (3.12)$$

where S_{thresh} is a threshold that defines how much uncertainty a certain grasp can withstand.

Frictional form closure constraint: A grasp will be said to have frictional form closure if the origin of the wrench space lies inside the convex hull of the contact wrenches (Siciliano and Khatib, 2008). In this work, we choose to use frictional form closure as the task constraints since we don't intend to plan grasps for any particular task. This constraint is formulated as follows,

$$\begin{aligned} & \exists \phi_j^i \in \mathbb{R}, \phi_j^i > 0, \sum_{i,j} \phi_j^i = 1, i = 1 \cdots m, j = 1 \cdots 3; \\ \text{s.t. } & \sum_{i,j} \phi_j^i \mathbf{w}_i^j = \mathbf{0}; \end{aligned} \quad (3.13)$$

where $\mathbf{w}_i^j \in \mathbb{R}^6, i = 1 \cdots m, j = 1 \cdots 3$ is the i -th primitive contact wrench of the j -th contact point (El Khoury et al., 2013), which depends on the contact points, contact normal directions and friction coefficient. Note that in (Murray et al., 1994) this property is also defined as force closure, here we adopt the definition in (Siciliano and Khatib, 2008) where force closure additionally requires that the internal forces are controllable by the hand.

Objective function: Many different grasp quality metrics can be selected as the objective function for the contact level grasp planning (Roa and Suárez, 2014). Here, we adopt a simple objective function that minimizes the distance between the center of the contact points and the origin of the object frame as follows ¹,

$$\min: \left\| \frac{1}{3} \sum_{i=1}^3 \mathbf{p}^i \right\| \quad (3.14)$$

where $\| \cdot \|$ represents the 2-norm.

To generate grasping points, we formulate the grasp planning as a constrained optimization problem subject to the above constraints while minimizing the objective. By using AMPL (A Mathematical Programming Language) and adopting IPOPT (Interior Point OPTimizer) as the optimization solver (Wachter and Biegler, 2006), our method can generate multiple feasible solutions for each object by varying the initial points for IPOPT. More details and examples of grasp optimization are provided in Section 3.6.

¹Note that if the center of gravity is chosen as the origin of object frame, then this objective function actually attempts to minimize the effect of gravity on the grasp. In this work, we use the geometric center of

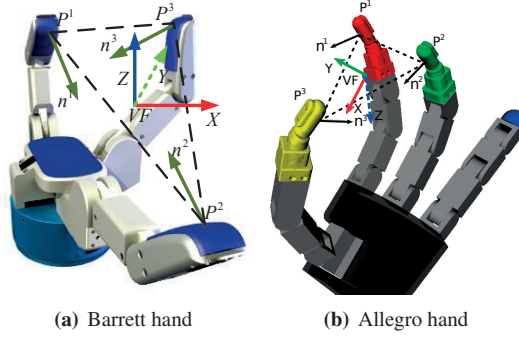


Figure 3.5: The examples of Virtual Frame for Barrett hand and Allegro hand.

3.4 HAND CONFIGURATION IN VIRTUAL FRAME

In this section, we present our learning-based approach for computing hand configurations needed to realize the planned grasping points. We first describe our frame invariant encoding of hand configurations in terms of a virtual frame. Thereafter, we introduce a probabilistic model for representing the mapping between contact positions and hand configurations, which is used to compute finger joints given the desired contact points.

3.4.1 PROBABILISTIC MODEL FOR HAND INVERSE KINEMATICS

We want the mapping between contact positions and hand configurations to be frame invariant. To this end, we follow the idea presented in our previous work (Li et al., 2014b) that learns a probabilistic representation using the concept of *Virtual Frame* (VF) (Tahara et al., 2010). A VF defined in the hand frame can be expressed as:

$$T_{VF}^{Hand} = \begin{bmatrix} R^o & \mathbf{p}^o \\ [0, 0, 0] & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \quad (3.15)$$

where \mathbf{p}^o is the origin of the VF with

$$\mathbf{p}^o = \frac{1}{3} \sum_{i=1}^3 \mathbf{p}^i \quad (3.16)$$

and $\mathbf{p}^i \in \mathbb{R}^3, i = 1, 2, 3$, is the position of the i -th fingertip. The orientation of the frame is defined by

$$\begin{aligned} R^o &= [\mathbf{r}_x, \mathbf{r}_y, \mathbf{r}_z] \in \text{SO}(3) \\ \mathbf{r}_x &= \frac{\mathbf{p}^3 - \mathbf{p}^1}{\|\mathbf{p}^3 - \mathbf{p}^1\|} \\ \mathbf{r}_y &= \mathbf{r}_z \times \mathbf{r}_x \\ \mathbf{r}_z &= \frac{(\mathbf{p}^2 - \mathbf{p}^1) \times \mathbf{r}_x}{\|(\mathbf{p}^2 - \mathbf{p}^1) \times \mathbf{r}_x\|} \end{aligned}$$

the object point cloud as the origin of the object frame

Two examples of the VF for Barrett hand and Allegro hand are shown in Fig. 3.5. With the definition of the VF, we encode a hand configuration \mathcal{G} as

$$\mathcal{G} = \{\Theta, L, N\} \quad (3.17)$$

where $\Theta \in \mathbb{R}^h$ is the finger joint. $L = [L_1, L_2, L_3] \in \mathbb{R}^3$ is the distance between each fingertip and the origin of the VF, i.e., $L_i = \|\mathbf{p}^i - \mathbf{p}^o\|$. $N = [N_1, N_2, N_3] \in \mathbb{R}^3$ is the pairwise difference of normal direction in the sense of inner product, $N_1 = \mathbf{n}^1 \cdot \mathbf{n}^2$, $N_2 = \mathbf{n}^1 \cdot \mathbf{n}^3$, $N_3 = \mathbf{n}^2 \cdot \mathbf{n}^3$. Given this encoding, all the variables Θ, L, N are frame invariant, and so is \mathcal{G} .

Given a set of hand configurations $\{\mathcal{G}^i, i = 1 \cdots N_g\}$ we can learn a probabilistic model – Gaussian Mixture Model (GMM) to represent the joint density of $\{\Theta, L, N\}$. This set of hand configurations can be obtained from simulation by sampling in the joint space, or from human demonstration through kinesthetic teaching. The likelihood of a grasp $\mathcal{G}_* = (\Theta_*, L_*, N_*)$ under a GMM model, denoted by Ω with m Gaussian components is given by

$$p(\mathcal{G}_*|\Omega) = \sum_{i=1}^m \pi_i \mathcal{N}(\mathcal{G}_*|\mu_i, \Sigma_i) \quad (3.18)$$

where π_i is the prior of the i th Gaussian component and $\mathcal{N}(\mu_i, \Sigma_i)$ is the Gaussian distribution with mean μ_i and covariance Σ_i as:

$$\mu_i = \begin{bmatrix} \mu_{\Theta,i} \\ \mu_{L,i} \\ \mu_{N,i} \end{bmatrix}, \Sigma_i = \begin{bmatrix} \Sigma_{\Theta\Theta,i} & \Sigma_{\Theta L,i} & \Sigma_{\Theta N,i} \\ \Sigma_{L\Theta,i} & \Sigma_{LL,i} & \Sigma_{LN,i} \\ \Sigma_{N\Theta,i} & \Sigma_{NL,i} & \Sigma_{NN,i} \end{bmatrix} \quad (3.19)$$

The number of Gaussian components, i.e., m is determined using Bayesian information criterion (BIC) and the parameters of π_i, μ_i, Σ_i in the model are trained using expectation-maximization (EM) algorithm to maximize the likelihood of all the trained grasps. More details for training and testing the probabilistic model for a specific hand will be given in Section 3.6.3.

3.4.2 ON-LINE HAND CONFIGURATION QUERY

Given a set of desired grasping points, we can now compute feasible hand configurations for realizing the grasp using the learned probabilistic model. The key idea here is to define two VFs using the fingertips and the grasping points respectively. Thus, the goal of on-line query is to search for a match of these two VFs. An example of on-line hand configuration query is shown in Fig. 3.6. We now explain the details of the query process in detail.

Query hand feasibility: Given a set of desired grasping points, we first construct a VF, $T_{VF}^{Obj} \in \mathbb{R}^{4 \times 4}$, similarly constructing the VF for hand configurations in Eq. (3.15). We then compute the corresponding L and N for this VF and determine if the current query point, i.e., $q = (L, N)$ is likely enough with respect to the learned model Ω . This

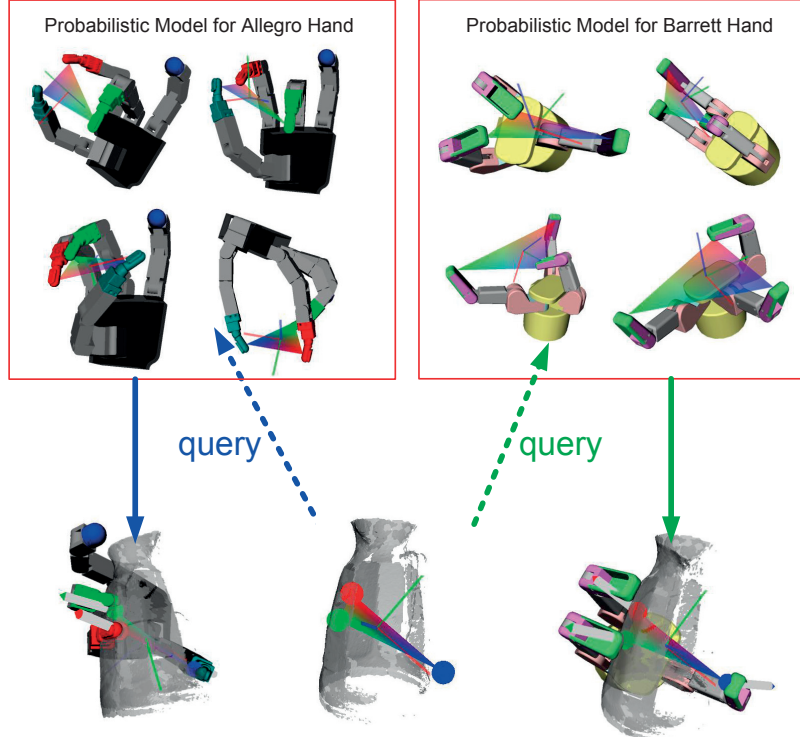


Figure 3.6: Given a set of grasping points on the object, we first construct a VF and use it as a key to compute corresponding hand configurations given the learned probabilistic model. In this figure, we also show a set of grasping points that can be reached by two different hands, which also explains the idea of being able to use different hands to execute grasps.

step computes the reachability of the hand given the grasping points. For instance, if two points are too far away and the distance between them is larger than the maximal spread length of the finger, these grasping points cannot be realized by the given hand. For this purpose, we use the Mahalanobis distance from q to the center of each Gaussian component. The distance to the i -th component is defined as:

$$f_i(q, \Omega) = \frac{1}{2}(q - \mu_{q,i})^T \Sigma_{q,i}^{-1} (q - \mu_{q,i}) \quad (3.20)$$

where $i = 1, \dots, m$ is the index of Gaussian components, $\mu_{q,i}$ and $\Sigma_{q,i}$ are the corresponding components in Eq. (3.18) as follows:

$$\mu_{q,i} = \begin{bmatrix} \mu_{L,i} \\ \mu_{N,i} \end{bmatrix}, \Sigma_{q,i} = \begin{bmatrix} \Sigma_{LL,i} & \Sigma_{LN,i} \\ \Sigma_{NL,i} & \Sigma_{NN,i} \end{bmatrix} \quad (3.21)$$

We consider that the likelihood that a query point q belong to the learned model is high enough if $\exists i, i = 1, \dots, m, f_i(q, \Omega) < 2$. In other words, if the query point is within two standard deviations of any Gaussian components of the model, it is considered to be close enough to the learned model. Otherwise, the grasping points are considered as infeasible for the given hand. Note here when we query the hand feasibility, the possible collisions between the hand and object are not taken into account. However,

the collision is checked in simulation (*OpenRave* (Diankov, 2010)) before the final selected grasp is executed.

Query Finger joints: When the current query point q is likely enough under the model, the desired finger joints Θ are obtained by taking the expectation over the conditional distribution, $p(\Theta|L, S, \Omega)$ (Mclachlan and Peel, 2000), which can be computed as follows:

$$E\{p(\Theta|L, N, \Omega)\} = \sum_{i=1}^m h_i (\mu_{\Theta,i} + \Sigma_{\Theta q,i} \Sigma_{q,i}^{-1} (q - \mu_{q,i})) \quad (3.22)$$

where $\Sigma_{\Theta q,i} = \begin{bmatrix} \Sigma_{\Theta L,i} \\ \Sigma_{\Theta S,i} \end{bmatrix}$, and $h_i = \frac{\pi_i \mathcal{N}(q|\mu_{q,i}, \Sigma_{q,i})}{\sum_{j=1}^m \pi_j \mathcal{N}(q|\mu_{q,j}, \Sigma_{q,j})}$.

Note that for three grasping points, we can have six different query points by permuting the correspondence between grasping points and the finger index. Therefore, given three grasping points, we may find several different hand configurations that can realize the grasping points. However, it is also possible that none of the six query points is likely enough under the model (e.g., object too big or too small), which implies that the given grasping points cannot be realized by the considered hand.

Query hand pose: After obtaining joint angles for each finger, we can again construct a VF $T_{VF}^{Hand} \in \mathbb{R}^{4 \times 4}$ using the fingertips position. This can be obtained using the hand's forward kinematics. Note that this VF is represented in the hand frame. Thus, the desired hand pose in the object frame, i.e., $T_{Hand}^{Obj} \in \mathbb{R}^{4 \times 4}$, can be obtained as follows:

$$T_{Hand}^{Obj} = T_{VF}^{Obj} T_{VF}^{Hand}{}^{-1} \quad (3.23)$$

Due to the probabilistic model we use to compute hand configurations, there can be errors between the realized fingertip positions and the desired grasping points. This position error is taken care of by our compliant grasp controller presented in the next section.

Remark: The proposed probabilistic hand inverse kinematics can be applicable to both under-actuated hand (e.g., Barrett hand) and fully-actuated hand (e.g., Allegro hand), given enough sampled hand configurations in the simulation. Moreover, in our more recent work (Hang et al., 2016), we demonstrate that the precision of the probabilistic hand inverse kinematics can be incrementally improved, when more realistic data of robot hand configurations are collected during the execution of the task.

3.5 GRASP CONTROL UNDER SHAPE UNCERTAINTY

Our approach superimposes position and force control, taking both the shape and contact force uncertainty into account. The control scheme for a given finger is represented as

$$\Delta \theta = J^{-1} [(1 - \lambda) K_P (x_d - x_c) + \lambda C_F (f_d - f_c)] \quad (3.24)$$

where J is the Jacobian of a finger. $x_d (f_d) \in \mathbb{R}^3$ and $x_c (f_c \in \mathbb{R}^3)$ are respectively

the desired and current fingertip positions (contact normal force), both of which are expressed in the hand frame. \mathbf{K}_P and \mathbf{C}_F are controller gains. $\lambda \in [0, 1]$ is a positional error measure to estimate how close the finger is to its desired position and weighted by the inverse of shape uncertainty as

$$\lambda = \exp\left(-\frac{1}{2}(\mathbf{x}_d - \mathbf{x}_c)^T \Sigma_{cov}^{-1}(\mathbf{x}_d - \mathbf{x}_c)\right) \quad (3.25)$$

By changing λ , the position controller is designed to first dominate and then smoothly switch to the force controller. Hence, a position error will be tolerated along the contact normal direction in order to regulate the contact force. Note that the desired fingertip position \mathbf{x}_d is the grasping point, i.e., $\mathbf{p}^i, i = 1, 2, 3$ in Section 3.3.2, but represented in the hand frame. The desired force \mathbf{f}_d is estimated as in our previous work (Li et al., 2014a). Moreover, the estimation can also provide the variance of the expected force value in the hand frame, i.e., $\Sigma_{\mathbf{f}_d} \in \mathbb{R}^{3 \times 3}$.

The diagonal matrices $\mathbf{K}_P \in \mathbb{R}^{3 \times 3}$ and $\mathbf{C}_F \in \mathbb{R}^{3 \times 3}$ are respectively the gain for position control and force control, which are usually selected heuristically. Here, we use the information from the variance of the desired position and desired force to choose proper parameters. For \mathbf{K}_P , we set it inversely proportional to the variance of the \mathbf{p}^i , i.e., $f_{cov}(\mathbf{p}^i)$ in Eq. (3.12). The variance of desired position is along the normal direction $\mathbf{f}_w(\mathbf{p}^i)$ (Eq. (3.11)) in the object frame, which can be transformed to the hand frame, denoted as Σ_{cov} . We have:

$$\mathbf{K}_P = \alpha_p \underbrace{|[diag(R_{Obj}^{Hand} f_{cov}(\mathbf{p}^i) \mathbf{f}_w(\mathbf{p}^i))]^{-1}]|}_{\Sigma_{cov}^{-1}} \quad (3.26)$$

where R_{Obj}^{Hand} is the rotation from hand frame to object frame, which can be obtained from Eq. (3.23). $diag(\cdot)$ means the diagonal entries of a matrix and $|\cdot|$ is the absolute value for each entry of a matrix. $\alpha_p \in \mathbb{R}^+$ is a scaling parameter. Similar for \mathbf{C}_F , we have,

$$\mathbf{C}_F = \alpha_f |[diag(\Sigma_{\mathbf{f}_d})]^{-1}| \quad (3.27)$$

The intuition behind the selection \mathbf{K}_P and \mathbf{C}_F is that when there is a large uncertainty on the desired contact point location, the finger will move more slowly by choosing a smaller \mathbf{K}_P . When there is a large variance in the desired contact normal force, we make the contribution of force controller smaller by using a smaller \mathbf{C}_F . This also implies that the finger will first contact the grasping point that has smaller uncertainty, and it is demonstrated in the experiments that this can improve the grasp success rate.

3.6 IMPLEMENTATION AND EXPERIMENTAL RESULTS

In this section, we will present our implementation and experimental results for object surface modeling and grasp planning using the Allegro hand.

3.6.1 RESULTS FOR OBJECT SURFACE MODELING

We evaluate our GP based object shape modeling method on four different objects: a cylinder, a bunny rabbit, a spray bottle and a jug, as shown in Fig. 3.7. The object point clouds are obtained from laser scanner and 1000 data points are randomly sampled from the original point cloud. To speed up the object shape modeling procedure, we further adopt a GP-based filter to select the most informative data points (Sommer et al., 2014) for representing the GP. The filtered data points for GP are shown as spheres on the object surface, Fig. 3.7(e)-(h). Table 3.3 shows the number of filtered GP data points and the computation time for filtering and shape modeling ².

As shown in Fig. 3.7(e)-(h), when the training data points are sampled from the whole object point cloud, the variance or shape uncertainty is generally very small on the whole surface except the parts with sparse or with even no data points, such as the bottom of the jug. In robotic grasping tasks, due to the occlusion (Bohg et al., 2011; Gori et al., 2014) or non-reachability from tactile exploration (Sommer et al., 2014), it is usually the case that some parts of the object are not perceivable and point-clouds exhibit holes. To evaluate our method under missing data points, we use MeshLab ³ to simulate partial view of point clouds with a fixed camera view, and then obtain object point cloud from that virtual camera. The results of object shape modeling with partial object point cloud are shown in the last two rows of Fig. 3.7. We can see that although the objects are partially viewed, our method can still model the shapes, and importantly, with explicitly computed uncertainties.

Table 3.3: The number of training data points (Nb.) for GP and the computation time for filtering (Time1) and shape modeling (Time2) on a 8 GB machine with a cpu at 2.4 GHZ.

Object	Nb.	Time1(s)	Time2(s)
Cylinder	59	24.51	0.34
Bunny	71	24.15	0.51
Spray	61	22.62	0.28
Jug	71	16.25	0.31

3.6.2 RESULTS FOR GRASP PLANNING

For each object model shown in the fourth row of Fig. 3.7, 1000 initial points for each of grasping point $\mathbf{p}^i \in \mathbb{R}^3, i = 1, 2, 3$ are randomly sampled from a sphere with radius 0.1. The coefficient of friction is set to 0.8. In our implementation, we set $S_{\text{thresh}} = 0.03$, which is three times larger than the noise level σ in Eq. (3.7). The number of final optimal grasps, the average computation time and the grasp quality are shown in Table 3.4. Some examples of the obtained optimal grasps for each of the four objects are shown in Fig. 3.8. It can be noticed that all the grasping points are in the area with small uncertainty due to the explicit shape uncertainty constraint.

²Note that the main time consumption comes from the filtering, this, however, can be done during the data collection if the object point cloud is collected on-line using vision or tactile exploration.

³<http://meshlab.sourceforge.net/>

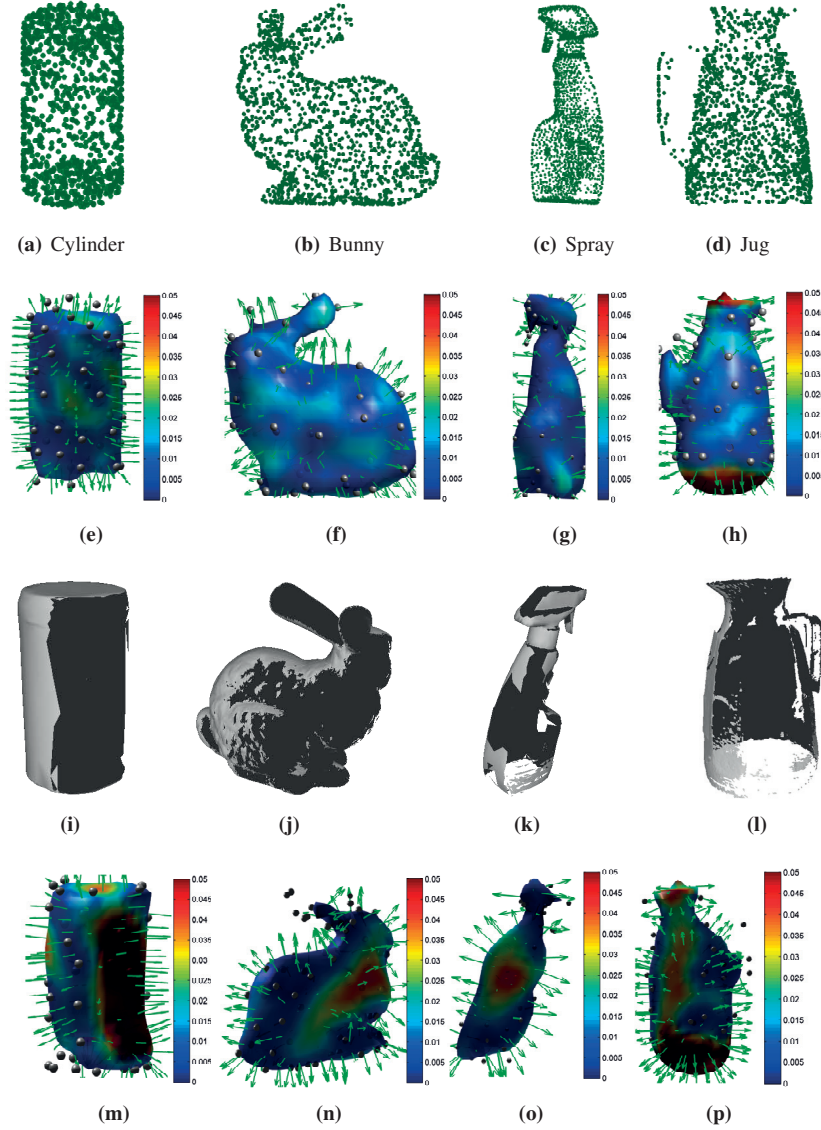


Figure 3.7: Four 3D objects and their corresponding GP representations. The first row show the original 3D point cloud object models, (a) a cylinder, (b) a bunny rabbit, (c) a spray, (d) a jug. The second row shows object shapes modeled by GP with whole object point cloud. Spheres on the object model are the 3D points on the object surface used to train the GP model. The arrow on the surface represents the normal direction predicted by GP on that point. The color of the surface represents the variance of the shape prediction on that point. The third row shows partial object point cloud from a fixed camera. The fourth row shows object shapes modeled by GP with partial object point cloud.

3.6.3 RESULTS FOR HAND CONFIGURATION QUERY

In this evaluation, we adopt two robotic hands – the 4 DOF Barrett hand shown in Fig. (3.9(a)), and the 16 DOF Allegro hand shown in Fig. (3.9(c)), as examples to evaluate the effectiveness of the learned probabilistic model described in Section 3.4.1. For each hand, we randomly sample $N_g = 10^6$ self-collision free hand configurations in the finger joint space in *OpenRave* (Diankov, 2010), and then use this dataset for

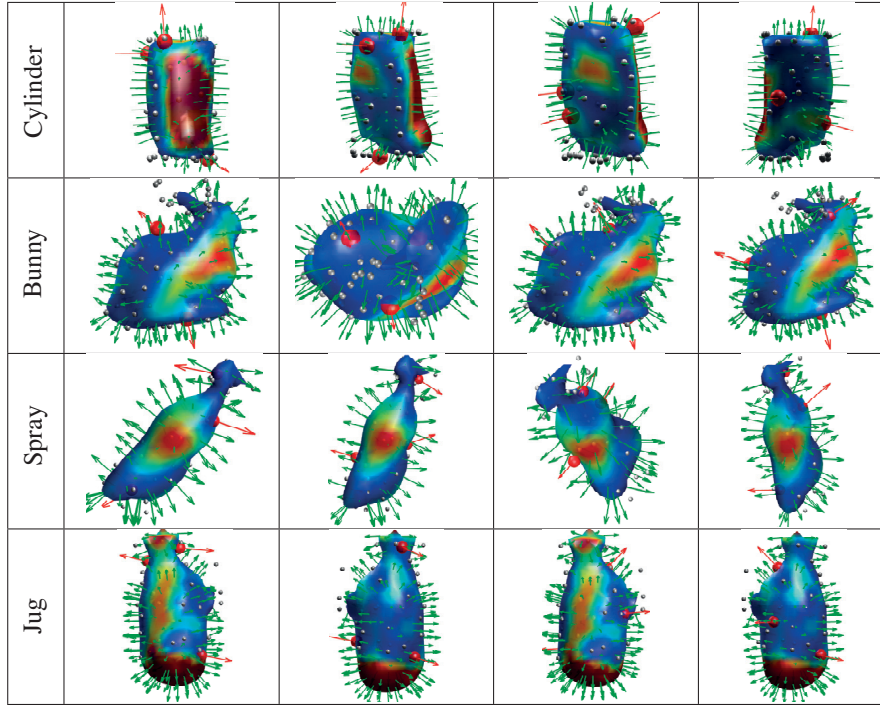


Figure 3.8: Four of the obtained grasping points (red balls on the surface) for each object and the red arrows represent the normal directions. The grasping points are outside the uncertain region.

Table 3.4: The number of final optimal grasps (Nb.) out of 1000 trials and the average computation time (Time) for each trial and the grasp quality (Q) using Eq. (3.14).

Object	Nb.	Time(s)	Q(cm)
Cylinder	797	17.51 ± 8.03	0.065 ± 0.064
Bunny	864	18.31 ± 10.25	0.0104 ± 0.029
Spray	986	8.46 ± 3.25	0.013 ± 0.32
Jug	914	26.52 ± 13.58	0.38 ± 0.75

further model training and testing.

For the model evaluation on Barrett hand, we use the first 4×10^5 hand configurations for model training and the rest for evaluation. For this hand, each data point is 10-dimensional: $\Theta \in \mathbb{R}^4, L \in \mathbb{R}^3, N \in \mathbb{R}^3$. The number of Gaussians in Eq. (3.18) is set to $m = 36$, as shown in Fig. (3.9(b))⁴. Using all the data from the test dataset as grasping points queries, we evaluate the accuracy of the model in terms of the Mean Absolute Error (MAE) in radians, as reported in Table. 3.5. Fig. 3.10 shows examples of predicted joint angles in comparison with the ground truth joint angles for four different testing cases, the MAE of these four examples are also reported in Table. 3.5.

To keep the number of grasping points the same for the Allegro hand, we consider only its first three fingers as show in Fig. 3.10. Therefore, we have $\Theta \in \mathbb{R}^{12}$, and the training data is 18-dimensional. Same as before, we use the first 4×10^5 hand

⁴The BIC gives a range of the number of Gaussians and the final number is determined by a 10-fold cross validation on a separate dataset.

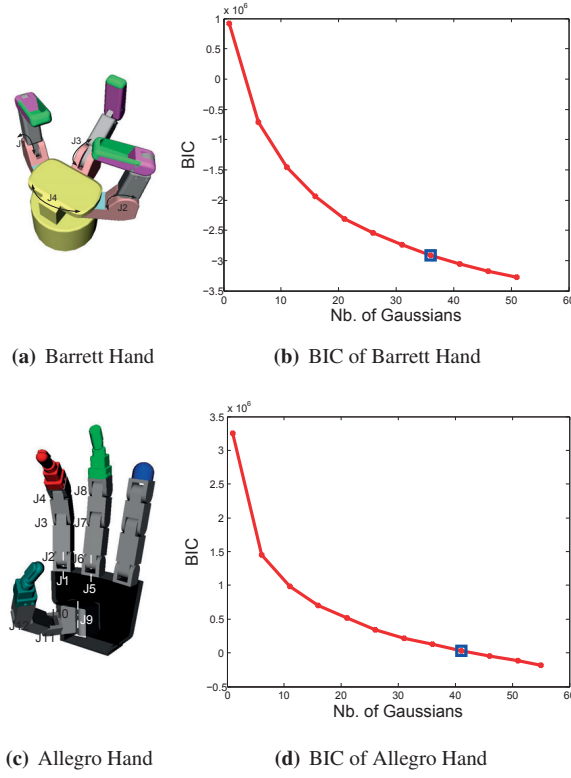


Figure 3.9: The hands models for the testing examples and the selection of number of Gaussians for training the model using BIC.

configurations for model training and the rest for testing. The number of Gaussians is chosen as $m = 41$, as shown in Fig. (3.9(d)). For the overall evaluation on the Allegro hand, the 6×10^5 data points have been used to test the probabilistic model, and the MAE with the standard deviation for the 12 joint angles is reported in Fig. 3.11.

From Table 3.5 and Fig. 3.11, the average error is around 0.2 rad. Note that the performance of our probabilistic hand inverse kinematics model depends largely on how we sample the valid hand configurations. In this chapter, we sample the hand configurations randomly in the finger joint space with self-collision rejection. We noticed that our dataset includes a lot of hand configurations that are self-collision free, but are unlikely to be valid grasp configurations. This is one cause of deterioration of the model performance. If some other information regarding the infeasibility of some postures

Table 3.5: The Mean Absolute Error between the prediction and ground truth joint angles for the four cases shown in Fig. 3.10 and the average error with the standard deviation over all 6×10^5 testing data for the evaluation on Barret hand. (Unit: Radian)

Joint	case 1	case 2	case 3	case 4	Overall
J1	0.03	0.06	0.27	0.26	0.16 ± 0.01
J2	0.17	0.01	0.27	0.002	0.17 ± 0.01
J3	0.04	0.07	0.25	0.08	0.21 ± 0.007
J4	0.03	0.2	0.06	0.56	0.36 ± 0.02

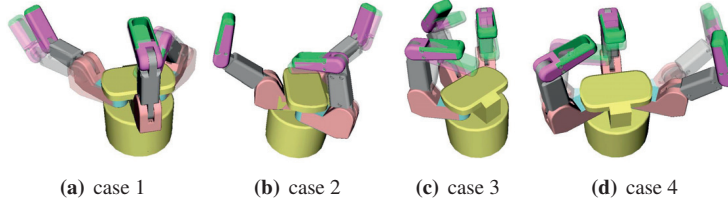


Figure 3.10: The joint angle prediction error of four examples for Barrett hand. The solid color hand shows the ground truth joint angles and the transparent color shows the predicted joint angles.

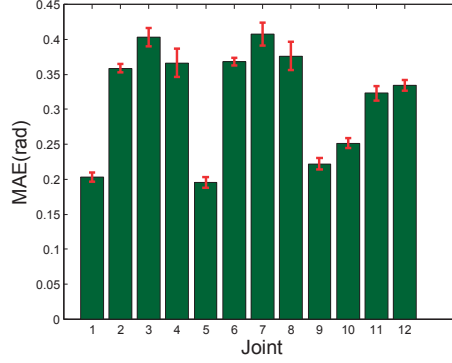


Figure 3.11: The prediction error for the 12 joints of Allegro hand.

(e.g. through models of the finger synergy or through human teaching) was provided to guide the sampling, this would likely improve the performance of the feasible solutions generated by our hand inverse kinematic model. This is one of our future working directions.

At this moment, to improve the performance, we adopt a local derivative-free optimization technique, called Constrained Optimization by Linear Approximation (COBYLA) (Powell, 1994). This optimization algorithm is based on linear approximation of the optimized objective function and all the constraints and transform the original problem to a linear program to solve. With this optimization technique, we can locally adjust the hand pose as well as the finger joints to improve the performance with respect to the objective function, which is chosen as the sum of the distance between desired grasping points and fingertip positions.

3.6.4 RESULTS FOR GRASP REALIZATION

In this section, we show qualitative examples of grasp execution described in Section 3.4. Fig. 3.12 shows that the same grasping points can be realized by two different hand configurations for the same hand. In Fig. 3.13, we show that the same grasping points can also be realized by two different hands.

In Fig. 3.14 and Fig. 3.15, we show some grasps for each hand using our learned probabilistic model. Note that all the grasps are generated using object models with partial point cloud, see Fig. 3.7. Collision between the hand and the object has also

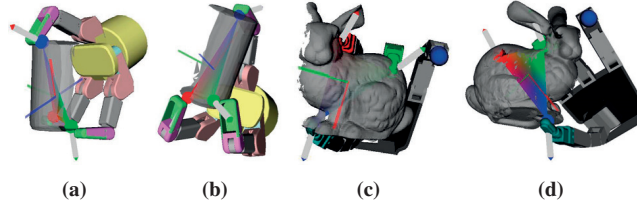


Figure 3.12: Two examples show that the same grasping points can be realized by the same hand but with two different hand configurations. The frame attached with the object is the obtained VF and the arrows at the contact points are the predicted normal directions.

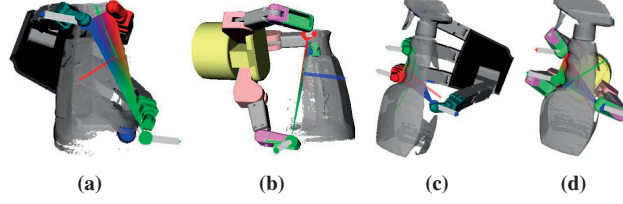


Figure 3.13: Two examples show that the same grasping points can be realized by two different hands

been integrated in the last step of our planning algorithm.

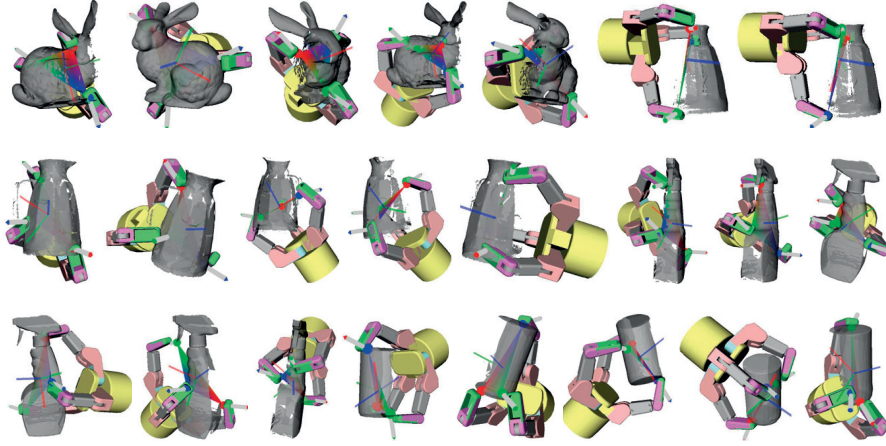


Figure 3.14: Some example grasps for Barrett hand. The red, green and blue point corresponds to finger 1, 2, 3 respectively. Notice that there are still some position errors between the fingertips and desired grasping points due to the probabilistic model we use.

3.6.5 IMPLEMENTATION ON REAL ROBOTIC HAND

Using the proposed grasp controller, we demonstrate several of our planned grasps using an Allegro hand mounted on a 7 DOF arm – KUKA LWR. Each fingertip of the Allegro hand is equipped with BioTac tactile sensors⁵ to estimate the contact normal force, namely $\mathbf{f}_c \in \mathbb{R}^3$ in Eq. (3.24). After our calibration, we can obtain a force prediction with an accuracy around 0.1N in the normal direction and 0.3N in the tangential directions. In this work, we only control the normal force, i.e., \mathbf{f}_c , represented in the

⁵<http://www.syntouchllc.com/>

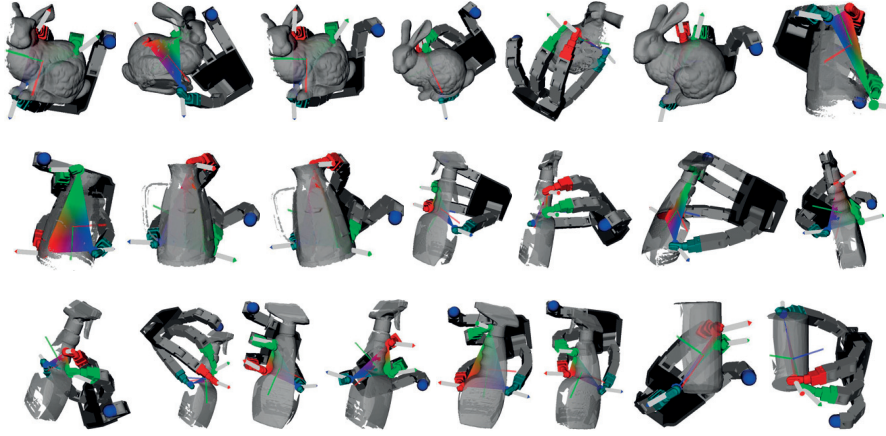


Figure 3.15: Some example grasps for Allegro hand. The red, green and blue point corresponds to finger 1, 2, 3 respectively.

hand frame. The object to be grasped is placed on the table and the position is obtained using vision tracking system – OptiTrack⁶. Once our grasp controller has assessed that all three fingertips are at the desired positions and that the contact forces are stabilized, we command the robot to lift the object and switch the hand controller to an object-level impedance controller (Li et al., 2014b), in order to adaptively regulate the grasp stiffness and keep the grasp stable. Some of the realized grasps are shown in Fig. 3.16.

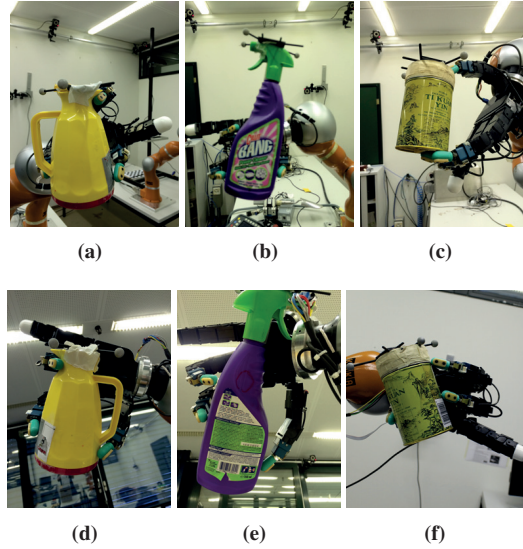


Figure 3.16: The implementation results of several planned grasps using Allegro hand. For the jug and spray bottle, the objects are placed on the table before lifting. For the tea can, the object is held by a human in order to realize grasp (c) which is unreachable from the bottom.

To evaluate how shape uncertainties affect grasp planning, we relax the upper bound of shape uncertainty S_{thresh} in Eq. (3.12), and generate grasps with different

⁶<https://www.naturalpoint.com/optitrack/>

level of uncertainties. The obtained grasps are ranked according to the sum of uncertainty on each grasping point and the first three grasps with the largest score are selected for implementation on the real robotic hand. When the uncertainty increases, one of fingertips is more likely to contact with the object first and then push the object away (or tilt the object), while the other fingertips are still approaching the object. As a result, the other fingertips may end up being far away from their desired positions and thus less likely to achieve a stable grasp.

For each grasp, the test is repeated ten times to capture the uncertain effect from pushing motion of fingertips⁷. The percentage of the grasps that are stable over all the trials and grasps for each object is reported in Table 3.6. We can see that the percentage of stable grasps after lifting decreases when the grasps become more risk-seeking, i.e., grasping on the object surface with large shape uncertainties. Note that the total success rate significantly increases if we decrease the upper bound of the shape uncertainty, which demonstrated the benefit obtained from explicit shape uncertainty representation. However, due to the imprecise hand dynamics, large friction at finger joints and joint actuator limitation, etc., the total success rate of the grasping system is still far from practical use, which implies a requirement for a reliable hand embodiment as well as more intelligent approach to address other various uncertainties in robotic grasping.

Table 3.6: Percentage of stable grasps achieved for each object under with different level of shape uncertainty, averaged over 30 trials (3 grasps x 10 times).

Object	Cylinder	Spray	Jug
$S_{\text{thresh}} < 0.03$	40%	63.3%	56.7%
$S_{\text{thresh}} < 0.06$	43.3%	33.3%	30%
$S_{\text{thresh}} < 0.10$	30%	26.7%	23.3%

To evaluate the performance of the grasp controller for finger closing, we compare it with a position controller for each fingertips with same isotropic gain⁸, and the finger stops when the contact force reaches $0.5N$ or the finger reaches desired position. The same grasps in the first row of Table 3.6 are tested with two different controllers: uncertainty-aware controller and position controller. The percentage of stable grasp achieved is reported in Table 3.7, which shows clear improvements given by our uncertainty-aware controller. As we have observed in the experiments, the object is more likely to be moved away when using a position controller with an isotropic gain. This is usually due to the fact that one fingertip would get contact with the object where it has larger uncertainty, and thus the object is shifted before other contacts are made. However, with our uncertainty-aware controller, the position error becomes less important when the fingertip is close to the desired position, and in most cases, the fingertips usually first get contacts on the object where it has smaller uncertainty, and thus smaller position error and less possibility to shift the object away.

⁷During the ten trials, the initial condition is the same. However, in practice we notice that the pushing from one fingertip may lead to very different object position and thus the outcome of final grasps.

⁸In practice, a PI controller is used and the gain is hand tuned to achieve the best performance from our experience.

Table 3.7: Percentage of stable grasps achieved for each object using different controllers: uncertainty-aware controller(**unc.**) and position controller(**pos.**), averaged over 30 trials (3 grasps x 10 times).

Object	Cylinder	Spray	Jug
unc.	40%	63.3%	56.7%
pos.	16.7%	10%	13.3%

3.7 DISCUSSION AND SUMMARY

While dexterous grasping is considered important for in-hand object manipulation, it is still very difficult to realize using real robotic hands. One of the main challenges resides in how to overcome the uncertainties in sensing, actuation and imperfect representation of the environment. This work addressed this challenge and considered shape uncertainty both in grasp planning and control stages. During grasp planning, the uncertainty of the generated grasp can be explicitly determined. Moreover, during grasp execution, the uncertainty of the generated grasp is fed into a compliant grasp closing controller to further improve the grasp stability.

To be more specific, we proposed an approach for grasp planning and control considering object shape uncertainty. A probabilistic model for estimation of hand inverse kinematics model is adopted to compute feasible hand configurations. During grasp execution stage, a compliant finger-closing controller taking into account the uncertainty has been devised to improve and retain grasp stability. Experiments on a real robotic hand demonstrate the effectiveness of our proposed method.

As a final remark, we list a number of limitations and future research directions. First, our controller considers only one type of uncertainties, namely uncertainties linked to the shape of the object. Other sources of uncertainties (e.g. imprecise finger positioning, inaccurate model of the object’s mass and friction coefficient) affect importantly the chance of success of a grasp. The grasp planning and control in this work is limited to shape uncertainty while other sources of uncertainties are not taken into account. Therefore, finding an analytical representation of these uncertainties and considering them in robotic grasping, as the framework in Fig. 3.1 will be a promising direction for the extension of this work. Second, the proposed approach in this chapter goes one-way from the planning phase to control phase, and there is no feedback from the control phase to the planning phase. In next chapter, we will propose a grasp adaptation approach that reuses the information from control phase to locally adjust the grasp configuration, and therefore can close the loop between grasp planning and control.

DYNAMIC GRASP ADAPTATION

4.1 INTRODUCTION

In the previous chapter, we addressed robust grasp planning and finger closing, considering the uncertainty resulting from imprecise perception of the object's shape. However, the executed grasp could still be unstable in the post-grasp stage (after the object is lifted, see Fig. 2.3), due to unexpected perturbations or change in the object's weight. Human hands can deal with these kinds of situations extremely well. For example, imagine you grasp a paper cup and someone else is adding some more water into your cup, your hand will gradually increase the grasping forces to balance the weight, and meanwhile attempt to not apply excessive forces to squash the cup. This whole process is so delicate that can hardly being noticed. Even though it is still not clear how this process of grasping is organized in the human brain, studies from neuroscience have demonstrated the tactile system on the hand plays a considerable important role during this process. Human brain may build a repertoire of sensory-motor coupling from previous grasp experience, which are used later to select proper corrective actions from tactile sensing (Castiello, 2005; Johansson and Flanagan, 2009).

Inspired from these human grasp studies, in this chapter we are attempting to endow robotic hands with the same capability like human hands that can adapt to the unexpected perturbations in the post-grasp stage (including the external disturbance acting on the object and the change in object's weight). To achieve this goal, two fundamental questions need to be answered, namely, *when to adapt* and *how to adapt*. We propose a dynamic grasp adaptation approach to address these two issues in a single unified framework. It allows a robotic grasp to dynamically predict the stability of the grasp under perturbation, and to flexibly react to the perturbation accordingly if the grasp is predicted as unstable. Specifically, the dynamic grasp adaptation framework consists of:

1. A grasp stability estimator is learned from human-guided grasp experience (i.e., control variables and tactile sensing) and encoded as a probabilistic model, for on-line grasp stability prediction.
2. An adaptation strategy is proposed to dynamically stabilize the grasp under perturbation, by either regulating the grasping forces on the fingertips or relocating the fingertips' positions in-hand. The desired position for the finger gaiting is informed by either previous grasp experience or a fast on-line grasp re-planning

algorithm, depending on the availability of the object’s shape.

3. A closed-loop integrated system is built that links grasp planning, grasp stability estimation and in-hand grasp adaptation together.

The novelty of our approach will be detailed in the problem formulation in next section. In general, item 1 answers the question of *when to adapt* and item 2 responds to the question of *how to adapt*. These two parts were published in (Li et al., 2014a) and form the main content of this chapter. The chapter also includes the experimental results from (Hang et al., 2015) where the experiments were conducted together with Kaiyu Hang from KTH. In this joint work, we integrated KTH’s fast grasp planning algorithm (Hang et al., 2014b) with our grasp adaptation framework (Li et al., 2014a). In this integrated system, when the finger gaiting is required, the desired fingertip’s position is provided through fast on-line replanning, instead of acquiring from previous experience in our original grasp adaptation strategy.

Related Publications:

- (Li et al., 2014a) Li, M., Bekiroglu, Y., Kragic, D., and Billard, A. (2014a). Learning of grasp adaptation through experience and tactile sensing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3339–3346
- (Hang et al., 2015) Hang, K., Li, M., Stork, J. A., Bekiroglu, Y., Pokorný, F. T., Billard, A., and Kragic, D. (2015). Hierarchical fingertip space: A unified framework for grasp planning and in-hand grasp adaptation. *IEEE Transactions on Robotics*. (Conditional accepted)

4.2 PROBLEM STATEMENT

As discussed in the introduction, to allow the robotic hand to adapt to the perturbation in post-grasp stage, two key questions that need to be addressed are: *when to adapt* and *how to adapt*. Generally, a robotic hand needs to adapt the grasp mainly for two purposes: to stabilize the grasp or to prepare for the next manipulation task. In this work, we only target the first purpose, namely to adapt a grasp so as to maintain the grasp stability. Thus we need to design a corresponding grasp adaptation strategy that allows to change a grasp from unstable to stable. In the remainder of this section, we will discuss why these two issues are challenging and then depict the novelty of our approach in relation to these challenges.

4.2.1 GRASP STABILITY PREDICTION

Although grasp stability is commonly used in the robotic grasping literature, the definition of grasp stability is still vague. For the study of grasp planning, grasp stability is often taken as synonymous to force closure, which is only a geometric (quasi-static) stability criteria that merely depends on the grasp configuration. This quality can not be used to predict the stability of a real grasp, due to their neglect of the dynamics of

the hand-object system and the inevitable uncertainties involved in this process (Kim et al., 2013; Zheng and Qian, 2005).

For a real grasp, the grasp stability has at least two meanings (Nakamura et al., 1989; Montana, 1991; Fearing, 1986). One is the ability to prevent incipient slippage (contact-level stability) and another more important one is the ability for the object to return to an equilibrium position after perturbation (object-level stability). While the former grasp stability is possible to predict using some cutting-edge tactile sensors (Takahashi et al., 2008; Kanno et al., 2013; Veiga et al., 2015), the latter one (i.e., the object-level stability) is more difficult to estimate and achieve in real-world scenarios. To explain this, let us take a look at the object dynamics equation (i.e., Eq. (2.1)) again and discuss how the object-level stability is addressed in conventional approaches.

$$\begin{array}{c} \text{object dynamics} \\ \mathbf{M}_r(\mathbf{x}_r)\ddot{\mathbf{x}}_r + \mathbf{C}_r(\mathbf{x}_r, \dot{\mathbf{x}}_r)\dot{\mathbf{x}}_r + \mathbf{g}_r(\mathbf{x}_r) = \mathbf{w}_f + \mathbf{w}_{\text{ext}} \end{array}$$

$\mathbf{w}_f = \mathbf{G}\mathbf{f}$

Figure 4.1: The object dynamics equation. In this section, object dynamics represents force that comes from the object’s dynamic terms, i.e., inertial, damping and gravity term. The detailed notation is given in Table 2.1.

Given a desired trajectory of the object $\mathbf{x}_d \in \mathbb{R}^6$, previous studies usually choose the resultant object wrench \mathbf{w}_f as (Jen et al., 1996; Nakamura et al., 1989; Wimböck et al., 2012). Here we drop the dependency on state variable for \mathbf{M}_r , \mathbf{C}_r , etc, to simplify the notation.

$$\mathbf{w}_f = -\mathbf{w}_{\text{ext}} + \mathbf{C}_r\dot{\mathbf{x}}_r + \mathbf{g}_r + \mathbf{M}_r\{\ddot{\mathbf{x}}_d + K_1(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}_r) + K_2(\mathbf{x}_d - \mathbf{x}_r)\} \quad (4.1)$$

Then the object motion is governed by the equation

$$\mathbf{M}_r(\ddot{\mathbf{x}}_d - \ddot{\mathbf{x}}_r) + K_1(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}_r) + K_2(\mathbf{x}_d - \mathbf{x}_r) = 0 \quad (4.2)$$

where $K_1 \in \mathbb{R}^{6 \times 6}$ and $K_2 \in \mathbb{R}^{6 \times 6}$ are constant matrices that guarantee asymptotic stability, which means that the object’s actual pose \mathbf{x}_r will converge to the desired one \mathbf{x}_d (i.e., dynamically stable).

Although the control law (Eq. (4.1)) can guarantee the object stability, it requires some precise information that can be extremely difficult to obtain in real-world situation. First, to compute the desired resultant object wrench, i.e., the right side of Eq. (4.1), requires to estimate the *object’s dynamics* (2nd, 3rd and 4th terms on the right-hand side) and to estimate the *external perturbation* (1st term on the right-hand side). It may be still possible to compensate the object’s dynamics if the object is known beforehand, however, the external perturbation is usually unknown before it applies, and it is difficult to measure it precisely measured due to the lack of sensors

mounted on the object that feedback this measurement to the hand.

Another issue arises from the fact that the wrench applied on the object, $\mathbf{w}_f = \mathbf{G}\mathbf{f}$, results from the forces applied by the fingers in coordination. Control of the forces at fingertip to generate the desired object-level wrench requires a very precise *grasp controller* and *force sensing* at the fingertips for close-loop control of forces. Also, the type of *friction* model can also affect the grasping forces on the fingertips. Moreover, this control law also assumes that the *grasp configuration* \mathbf{G} is precisely known and does not change during the whole process. However, the actual grasp configuration will largely depend on the strategy of *finger closing*, the precision of the actual object’s *pose* and *shape*. For instance, if during the finger closing, one finger contacts first with the object, the object may be moved or tilted and thus the final grasp configuration will not be the same as the planned one.

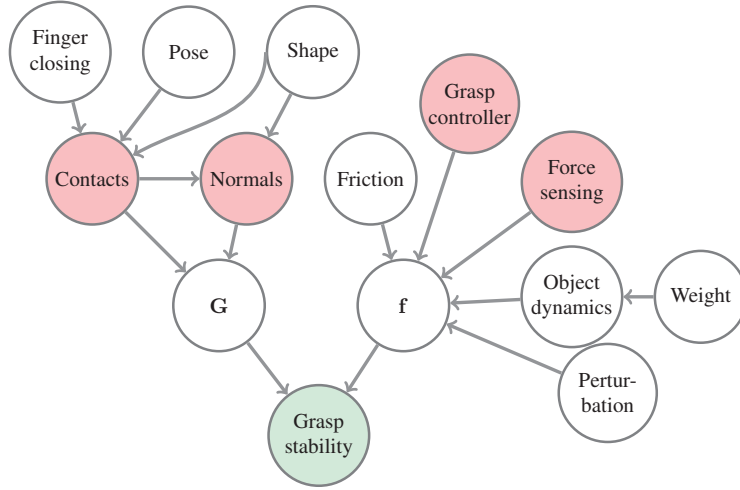


Figure 4.2: The main factors that can influence the final grasp stability. The arrow between two nodes means the former node can influence the latter one. \mathbf{G} is the grasp matrix in Eq. (2.3) and \mathbf{f} is the vector of grasping forces. The nodes with red color are the variables or parameters that can be directly observed or regulated in our dynamic grasp adaptation framework. The meaning of each node is explained in the main text above.

To sum up the above discussion, there are a large number of factors that can actually affect the final grasp stability, as summarized in Fig. 4.2, which makes the control law in Eq. (4.1) impractical to use and thus impossible to estimate the grasp stability. To address these challenges, we propose a learning-based approach that learns a probabilistic grasp stability estimator from human-guided grasp experience. Recall the control law Eq. (4.1), it actually means that a proper \mathbf{G} and \mathbf{f} should be chosen and realized to achieve stability, given different object dynamics and perturbations. Rather than relying on the precise analytical model to estimate the grasp stability (Eq. (4.2)), our approach leverages the power of grasp learning from human demonstration that shows a number of “good” grasps to the robotic hand, with different object dynamics and perturbations by changing the object’s weight. During the demonstration, human demonstrator gradually increases the grasping forces through passive guidance of the

fingers¹. When the object’s weight is totally balanced, the corresponding grasp is considered as a good grasp. With the demonstration, a great amount of grasp experience can be collected, consisting of the grasp configuration \mathbf{G} , the control variables \mathbf{f} , and the tactile sensing \mathbf{S} on the fingertips. Given the collected grasp experience, the grasp stability can be defined as an indicator function to evaluate if a new grasp is similar to these demonstrated grasps or not

$$I_G = g(\mathbf{G}, \mathbf{f}, \mathbf{S}) \quad (4.3)$$

where I_G is the indicator of grasp stability, which can be binary or continuous, and g is the mapping from $(\mathbf{G}, \mathbf{f}, \mathbf{S})$ to the grasp stability.

Our definition of grasp experience actually attempts to encode an implicit relations among the grasp configuration \mathbf{G} , the control variables \mathbf{f} , and the tactile sensing \mathbf{S} . Compared with Eq. (4.1), it actually assumes the right side of Eq. (4.1), namely the perturbation and the object dynamic force can be encapsulated by the tactile sensing \mathbf{S} . This assumption is valid if the tactile sensing is sensitive enough to normal contact force and tangential contact force, and different external forces exerted on the object will cause different tactile readings. It is important to note that since we don’t use the analytical model (Eq. (4.2)) to evaluate the grasp stability, the grasp configuration \mathbf{G} and the control variables \mathbf{f} in Eq. 4.3 can be represented by the variables that they are dependent on instead of their original representation. For example, the grasp configuration \mathbf{G} is largely dependent on the contact positions and the contact normal directions, thus in the stability estimator Eq. 4.3 the grasp configuration \mathbf{G} can be replaced by the features defined by the contact positions and the contact normal directions. This will become more clear in Section 4.3 where we explain the data collection and preprocessing.

4.2.2 DYNAMIC GRASP ADAPTATION

As discussed in the introduction, besides the grasp stability estimation, the second key question that needs to be answered is *how to adapt*. The adaptation strategy usually depends on the type of uncertainties, the grasp controller and the hand’s embodiment. For example, if we use a gripper with a position controller, then the options of the adaptation strategy could be quite limited, compared with a multi-fingered dexterous hand. The uncertainty from object shape and the uncertainty from object weight will require totally different adaptation strategies. In short, it is not a trivial task to design a proper grasp adaptation strategy.

Many grasp adaptation strategies have been proposed before to deal with different types of uncertainties, as summarized in Table 3.1. Depending on the grasp stage that the uncertainties are considered (see Fig. 2.3), these strategies can be classified into two types. First, during the pre-grasp stage, namely, reaching and finger closing, the grasp adaptation is usually performed in terms of local adjustment of the robot’s wrist

¹Note that this kind of demonstration is possible if the fingers of the robotic hand are back-drivable.

pose, which actually locally adapt the grasp configuration \mathbf{G} . Second, during the post-grasp stage when the object is already in-hand, a typical grasp adaptation strategy is to regulate the grasping forces \mathbf{f} to prevent incipient slippage of the object (i.e., contact-level grasp stability). Note that neither of these two types of grasp adaptation strategies are designed for the object-level grasp stability, and moreover the proper values of the grasping forces \mathbf{f} can be tricky to design if more than two fingers are used for grasping. Finally, regulation of the grasping forces \mathbf{f} may be insufficient if the perturbation or the object's weight is varying vastly.

As mentioned at the beginning of this chapter, this work also attempts to address the grasp adaptation during the post-grasp stage to deal with the perturbations and the change in object's weight. Based on the grasp stability estimator described in previous section and an object-level impedance grasp controller we adopted (Appendix A), we propose a dynamic grasp adaptation strategy that consists of two different corrective actions: *impedance adaptation* and *finger relocation*. The impedance adaptation changes the grasp stiffness that in fact implicitly regulates the grasping forces, and the finger relocation changes the position of one fingertip and thus changes the grasp configuration. Once a grasp is predicted as unstable, these two corrective actions are launched accordingly depending on current tactile sensing (which is assumed to encode the knowledge of the perturbation and the object dynamics). The details of our grasp adaptation strategy will be presented in Section 4.4.

Note that there are several key features that distinguishes our dynamic grasp adaptation from previous approaches. (1) The whole process from grasp stability prediction to grasp adaptation is fast and efficient. (2) The finger gaiting is performed in-hand and on-line. The object will drop if the gaiting finger is not moving fast and correctly. (3) During the finger relocation, less than three fingers can be in contact with the object. The grasp forces of these two fingers may be not balanced and thus the grasp is statically unstable during this short period.

4.2.3 PIPELINE OF OUR APPROACH

In this section, we summarize the pipeline of our dynamic grasp adaptation approach, which consists of two main parts: the off-line learning of *grasp stability estimation* and the on-line execution of *dynamic grasp adaptation*, as shown in Fig. 4.3. During the off-line learning, the human demonstrator shows a large number of good grasps to the robotic hand through passive guidance of the fingers. The corresponding grasp experience in terms of control variables, grasp configuration and the tactile sensing are collected and encoded as probabilistic model, which is used as an indicator of the grasp stability. The details of the grasp stability learning is presented in Section 4.3.

During the on-line execution stage, the corresponding grasp status of the real grasp can be evaluated on-line by the learned grasp stability estimator. Once the real grasp is predicted as unstable, the grasp adaptation strategy is triggered and two corrective

action will be launched respectively depending on the detailed situation of the current grasp. If the grasp is unstable, but still not so far away from the collected grasp experience, then an impedance adaptation strategy is used to stabilize the grasp, see Section 4.4.1. While the grasp is unstable and is far away from the collected grasp experience (this is usually the case if the perturbation is too large), then a finger relocation strategy is launched to change the location of one fingertip. The desired new location of the fingertip can be obtained through a guided local finger exploration (see Section 4.4.2) or through a finger gaiting informed by a fast grasp replanning algorithm (see Section 4.6). To be more clear, if the object’s shape is unknown during the grasp adaptation, the new desired position for the relocation finger is obtained through local exploration. On the other hand, if the object’s shape is available during the adaptation, then a fast on-line grasp replanning algorithm (Hang et al., 2015) is used to search for a better position for this relocation finger (finger gaiting). To avoid confusion, in this chapter, we will first present the work that only focuses on the grasp stability estimation and the grasp adaptation strategy without the knowledge of object’s shape. Then in Section 4.6, we will present an integrated system that assumes the object’s shape is known and thus a finger gaiting strategy can be used. Note that through the fast in-hand grasp adaptation and the on-line grasp stability estimation, the on-line grasp execution in fact becomes a closed-loop system which bridges the gap between grasp planning and control.

4.3 GRASP STABILITY ESTIMATION

In this section, we describe our approach to learn the grasp stability estimator. Compared with the previous methods by Bekiroglu et al. (2011) and Dang and Allen (2013), where the grasp stability is defined over the joint positions and hand pose, here we use a different grasp feature that is directly extracted from the object-level impedance controller (Tahara et al., 2010; Li et al., 2014b), see Fig. 4.4. In this controller, two control parameters are used for stable grasping: the stiffness K and the rest length L . The stiffness is computed at the object’s level and the rest length is the desired distance between each fingertip and the origin of the virtual frame. By varying the stiffness, the grasping forces at each contact point can be regulated, while the grasp configuration can be locally adjusted by changing the rest length. Therefore, the dynamic grasp adaptation can be naturally integrated with this controller.

To begin with, we describe the data used for training the grasp stability estimator. In this chapter, we only consider precision grasps by three fingers, however our method can be easily extended to grasps using more than three fingers. The notation used is as follows:

- $D = \{(K^i, L^i, S^i)\}_{i=1\dots N}$ denotes a dataset with N observations.
- $K^i = (K_1^i, K_2^i, K_3^i) \in \mathbb{R}^3$ denotes the grasp stiffness at each fingertip, as shown in Fig. 4.4.

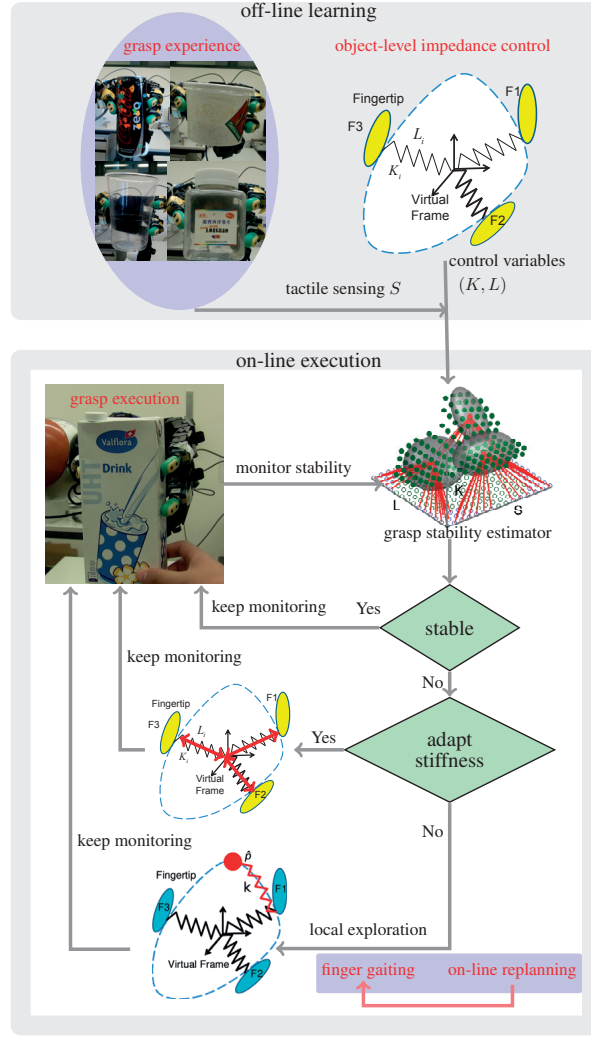


Figure 4.3: The pipeline of our approach, including the off-line learning of *grasp stability estimation* and the on-line execution of *grasp adaptation*. Note the link from on-line grasp replanning to finger gaiting requires the knowledge of the object’s shape, and this part will be presented in the integration section, i.e., Section 4.6.

- $L^i = (L_1^i, L_2^i, L_3^i) \in \mathbb{R}^3$ denotes the rest length at three fingertips, where L_j^i is the distance from the j -th fingertip to the center of virtual frame.
- $S^i = (S_1^i, S_2^i, S_3^i) \in \mathbb{R}^{57}$ denotes the tactile reading at three fingertips, where $S_j^i \in \mathbb{R}^{19}$ is the tactile reading from the j -th fingertip. In this work, we use the tactile sensor – BioTac², which can provide 19 dimensional tactile reading.

The recorded data thus consist of tactile readings S and parameters of the object-level controller, grasp stiffness K and rest length L . The tactile readings are high dimensional and redundant. Thus we use the Principal Component Analysis (PCA) to reduce the dimensionality. Hereafter, we will use S to denote the tactile reading after dimensionality reduction. Before the training procedure, all the data are normalized to

²<http://www.syntouchllc.com/>

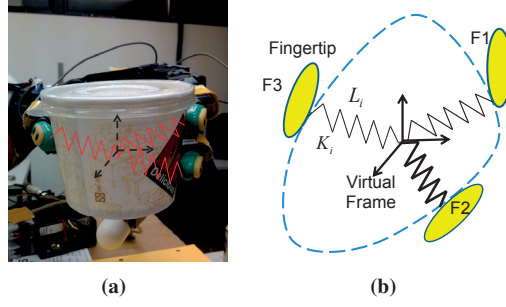


Figure 4.4: The scheme for object-level impedance controller. The Virtual Frame (VF) is defined as the center of the three contact points and the controller will control the position and orientation of the VF instead of the real object configuration.

zero mean with range $[-1, 1]$.

Remark: Recall Eq. 4.3, the grasp stability depends on grasp configuration, the control variables and the tactile sensing. The control variables include K, L and the tactile sensing consists of S . It is clear that the grasp configuration depends on the contact position and contact normal directions. Here we use L to encode the features of the contact positions. However, the normal directions at contact are not included. There are two reasons for this: first the hand we use has a limited capability to perform very different grasps; Second, for most of the everyday objects we use, their grasps have very similar normal directions at contact, see the grasp experience in Fig. 4.3.

With the recorded data, we formulate the grasp stability estimation as a *one-class classification* problem. This means that only the positive data, i.e., data from stable grasps, are used to learn the boundary of the stable grasp region. In general, the unstable grasps in our experiments can be seen as the noisy versions of the stable grasps, i.e., too many ways to perform a bad grasp, because they do not have inherit structures to learn. Due to this, binary classification experiments did not improve the classification accuracy. Therefore it is sufficient to use only one-class approach for our data. Moreover, we noticed that in practice, it may require a huge amount of time and expense (damaging the object or the finger) to collect sufficient negative data, which would make the approach infeasible in practice. Therefore, in this work, we only use the positive data to learn the region of stable grasps.

In order to learn the boundary of the stable grasp region, two types of nonlinear classifiers, namely Gaussian Mixture Model (GMM) and Support Vector Machine (SVM) are used. GMM is a generative approach that models the probability distribution over data. The likelihood of a grasp $X_* = (K_*, L_*, S_*)$ under a GMM model denoted by Ω with m Gaussian components is given by:

$$p(X_*|\Omega) = \sum_{i=1}^m \pi_i \mathcal{N}(X_*|\mu_i, \Sigma_i) \quad (4.4)$$

where π_i is the prior of the i th Gaussian component and $\mathcal{N}(\mu_i, \Sigma_i)$ is the Gaussian

distribution with mean μ_i and covariance Σ_i as follows:

$$\mu_i = \begin{bmatrix} \mu_{K,i} \\ \mu_{L,i} \\ \mu_{S,i} \end{bmatrix}, \Sigma_i = \begin{bmatrix} \Sigma_{KK,i} & \Sigma_{KL,i} & \Sigma_{KS,i} \\ \Sigma_{LK,i} & \Sigma_{LL,i} & \Sigma_{LS,i} \\ \Sigma_{SK,i} & \Sigma_{SL,i} & \Sigma_{SS,i} \end{bmatrix} \quad (4.5)$$

A new grasp is said to be stable if its likelihood of being generated by the model is greater than a fixed threshold, i.e. $p(X_*) > te$. The discrimination threshold $te \in [a, b]$ is chosen according to the ROC (Receiver Operating Characteristic) curve, where the bounds a and b correspond to the minimal and maximal likelihood of each Gaussian component at two standard deviation, respectively, that is,

$$\begin{aligned} Lik_{2\sigma}(i) &= (2\pi)^{-\frac{d}{2}} |\Sigma_i|^{-\frac{1}{2}} e^{-2} \\ a &= \min_{i=1\dots m} Lik_{2\sigma}(i) \\ b &= \max_{i=1\dots m} Lik_{2\sigma}(i) \end{aligned} \quad (4.6)$$

As a comparison, we also applied the SVM classification to our problem. SVM for one class classification is a maximum margin classifier that attempts to separate the data from the origin with maximum margin in the feature space (Schölkopf et al., 2001). Here, we use the SVM implementation from Chang and Lin (2011).

It is worth mentioning that other one-class classification methods (Chandola et al., 2009), can also be adopted here for the grasp stability estimation. In our work, GMM is chosen because: (1) it has already shown its ability at estimating region of stable grasps in the high-dimensional joint space for different hands (Huang et al., 2013; Sauser et al., 2012); (2) the grasp adaptation strategy can be naturally derived from the learned model, as will be explained in the next section.

4.4 GRASP ADAPTATION

The grasp adaptation procedure is required when the current grasp is predicted to be unstable. In this case, corrective actions should be launched and driven by the current tactile information. In our controller, grasp adaptation consists of changing the object level impedance controllers according to the tactile feedback. Specifically, we will adapt the grasp stiffness K and the rest length L . The goal of grasp adaptation is to find a similar stable grasp using our model constructed in Section 4.3. The following parts of this section will present two corrective actions according to the similarities between the acquired tactile readings and the training dataset.

4.4.1 IMPEDANCE ADAPTATION

As illustrated in Fig. 4.3, when a grasp $X = (K, L, S)$ is predicted to be unstable, our first adaptation strategy is to adapt the grasp stiffness. This regulates indirectly the contact forces at each finger. This adaptation strategy corresponds to a typical response in humans whereby the grip force is increased to maintain the stability of grasped (or

manipulated) object (Johansson, 1998). Once an unstable grasp is detected, such as slippage occurring on one fingertip, the desired grasp stiffness \hat{K} is predicted from the GMM through regression, given the current rest length L and tactile reading S .

During the execution of grasp adaptation, however, we first need to check whether the current query point $q = [L^T, S^T]^T$ is likely enough with respect to learned model. In other words, we need to check if the current query point q is close enough to the training examples. This step is required mainly because a query point with low likelihood, i.e., far away from the training examples, may give a very poor prediction that may lead to even more unstable grasps.

In order to determine if the current query point q is likely under the learned model Ω , we compute the Mahalanobis distance from q to the center of each Gaussian component. The distance to the i th component is defined as:

$$f_i(q, \Omega) = \frac{1}{2}(q - \mu_{q,i})^T \Sigma_{q,i}^{-1} (q - \mu_{q,i}) \quad (4.7)$$

where $i = 1, \dots, m$ is the index of Gaussian components, $\mu_{q,i}$ and $\Sigma_{q,i}$ are the corresponding components in (4.4) as follows:

$$\mu_{q,i} = \begin{bmatrix} \mu_{L,i} \\ \mu_{S,i} \end{bmatrix}, \Sigma_{q,i} = \begin{bmatrix} \Sigma_{LL,i} & \Sigma_{LS,i} \\ \Sigma_{SL,i} & \Sigma_{SS,i} \end{bmatrix} \quad (4.8)$$

Note that compared to the marginal likelihood $p(q|\Omega)$, the Mahalanobis distance $f_i(q, \Omega)$ has the advantage to give the same importance to each Gaussian component, due to the absence of the prior in (4.7) and the fact that we normalize the distance to the center of the Gaussian. This has also the merit to avoid biasing the selection of grasps toward large Gaussian component, which may happen because of the non-uniform distribution of the training dataset.

In this work, we consider a query point q is likely enough to belong to the learned model, if its Mahalanobis distance to at least one of the Gaussians of the model is below two, i.e., $\exists i, i = 1, \dots, m, f_i(q, \Omega) < 2$. In other words, if the query point is inside the two standard deviations of any Gaussians of the model, then it is considered to be close enough to the learned model.

When the current query point q is likely enough under the model, the expected grasp stiffness is obtained by taking the expectation over the conditional distribution, $p(\hat{K}|L, S, \Omega)$ (Cohn et al., 1996), which can be computed as follows:

$$E\{p(\hat{K}|L, S, \Omega)\} = \sum_{i=1}^m h_i(\mu_{k,i} + \Sigma_{Kq,i} \Sigma_{q,i}^{-1} (q - \mu_{q,i})) \quad (4.9)$$

where $\Sigma_{Kq,i} = \begin{bmatrix} \Sigma_{KL,i} \\ \Sigma_{KS,i} \end{bmatrix}$, and $h_i = \frac{\pi_i \mathcal{N}(q|\mu_{q,i}, \Sigma_{q,i})}{\sum_{j=1}^m \pi_j \mathcal{N}(q|\mu_{q,j}, \Sigma_{q,j})}$.

The expected grasp stiffness value will be thus set directly to the object level impedance controller, as shown in Fig. 4.3. The stability of this adaptive controller is proved in Appendix B, which shows an upper bound on the change rate of grasp

stiffness can guarantee the stability of whole hand-object system.

4.4.2 LOCAL EXPLORATION

When the current query point q is far away from all the training examples, a second adaptation strategy takes place to project the current query point to the closest point q_* in the training dataset. Since in the controller only the rest length is required to determine the grasp configuration, we move the current grasp configuration towards the closest center of the Gaussian components, which is chosen according to the same distance metric defined above (4.7). The rest length of the closest center of Gaussian component is denoted as $L^c = [L_1^c, L_2^c, L_3^c]^T$.

However, differently from the grasp stiffness, we can not simply change these parameters in the controller since the definition of the virtual frame depends on the contact position at each fingertip. Moreover, the kinematics of each finger and the local geometry of the object surface will also impose several constraints on the possible movements of each fingertip. We therefore adopt an iterative approach by defining the following objective function:

$$J = \sum_{i=1}^3 (L_i - L_i^c)^2 \quad (4.10)$$

where $L = [L_1, L_2, L_3]^T$ is the current rest length. In this work, we only consider the adaptation motion from one of the fingers, denoted here as finger 1, see Fig. 4.5. For a more general case with multi-fingered grasp (more than three fingers), a more elaborate strategy that considers each finger's workspace and the grasp stability would be required to determine which finger should be used for adaptation.

In order to minimize the objective function, the gradient of the objective function with respect to the position of finger 1, i.e., $P_1 = [X_1, Y_1, Z_1]$, is given as $v = [\frac{\partial J}{\partial X_1}, \frac{\partial J}{\partial Y_1}, \frac{\partial J}{\partial Z_1}]^T$, where each component of v is computed as:

$$\frac{\partial J}{\partial X_1} = \sum_{i=1}^3 \frac{\partial J}{\partial L_i} \frac{\partial L_i}{\partial X_1} \quad (4.11)$$

$\frac{\partial J}{\partial L_i}$ and $\frac{\partial L_i}{\partial X_1}$ can be easily computed from the definition of J in (4.10) and the definition of L_i as given in Section 3.4.1.

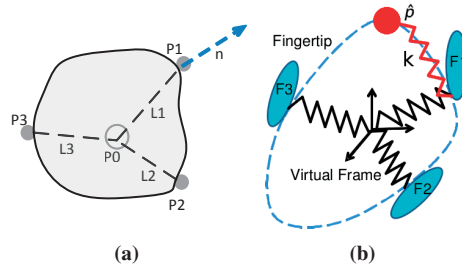


Figure 4.5: The scheme for local object exploration. Only finger 1 is used here for local exploration and the normal direction n obtained from tactile sensing is used for guiding the exploration directions.

Due to the constraint of the object surface, the fingertip 1 can not penetrate inside the object surface. Therefore we need to project the gradient onto the tangential surface at fingertip 1: $v^* = v - \langle v, n \rangle n$, where n is the normal direction at fingertip 1. Then the next desired position of finger 1 is given by: $P_1^* = P_1 - \alpha v^*$, where $\alpha \in (0, 1)$ is the step size and is manually set as 0.03 in our experiment³. In order to move finger 1 to the desired position, we implement a fingertip impedance controller for finger 1, which is superimposed on the object-level impedance controller.

It is worth noticing that during the local exploration, only the rest length L can be controlled. This does not guarantee that a similar query point q , consisting of the rest length L and the tactile sensing S , can be always found. To this end, a maximum number of steps for local exploration is set in our controller, i.e., $N_{max} = 5$. If a similar query point cannot be found within N_{max} times steps, our controller will recompute the closest center of the Gaussian components according to (4.7) and repeat the local exploration loop.

Remark: Note that during the local exploration, even though the moving direction of the finger is guided by the objective function Eq. (4.10), it is not guaranteed that the new position is actually on the object's surface since we don't require any priori knowledge of the object shape. Moreover, during the local exploration, the hand reachability is not checked and it is possible that the desired fingertip's position is not reachable for the finger. Therefore, if the object's shape is known, it is desirable that our grasp adaptation strategy can make use of this information. In Section 4.6, we will present such an integrated system that takes the advantage of the information of the object's shape during the grasp adaptation. The desired position for the exploration finger is provide by a fast on-line grasp replanning algorithm, rather than by a guided local exploration.

4.5 EXPERIMENTAL EVALUATION

In this section, we demonstrate the effectiveness of our approach on several everyday objects. We first present the experimental setup and the data collection procedure, then present the results on grasp stability estimation and grasp adaptation.

4.5.1 SETUP AND DATA COLLECTION

As shown in Fig. 4.6, we use a 16 DOFs (degree of freedom) *Allegro Hand* from Simlab⁴ with four fingers. Each finger has four independent torque-controlled joints. In our experiments we only use three of these four fingers. Each fingertip has been equipped with the tactile sensor *BioTac* from SynTouch⁵. BioTac can provide multi-modal tactile information, such as vibration, temperature and pressure. Here, we only use the 19 dimensional pressure data from the electrode impedances (see Fig:4.6(b)), which are related to the contact features such as contact force, contact location and

³Note that there is a trade-off here. If α is small, the object may fall before finding a stable grasp. If α is too large, the finger may overshoot the desired position or move out of the object surface.

⁴<http://www.simlab.co.kr/Allegro-Hand.htm>

⁵<http://www.syntouchllc.com/>

deformation. The sampling rate is 100Hz in our experiments.

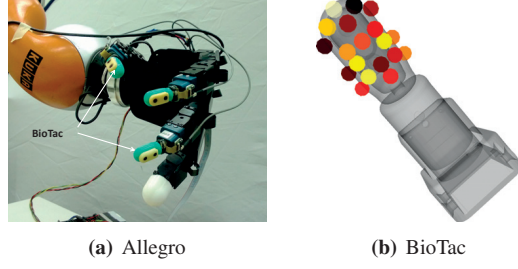


Figure 4.6: Allegro Hand equipped with BioTac on the fingertips.

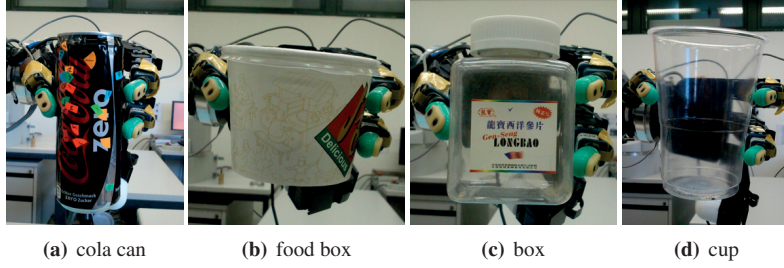


Figure 4.7: The four objects used in the experiments for collecting training dataset.

Our primary goal in this paper is to deal with physical uncertainties and therefore we assume that the object is already grasped with a given preshape, Fig. 4.6. The geometric information about the object is not known a-priori. Four different objects are used in our experiments, shown in Fig. 4.7. For each object we have five different weights by filling them with different amount of pepper, see Table 4.1. The initial grasp stiffness is manually chosen as a minimal value that can grasp the object in a stable way. Before recording, the initial grasp may be slightly adjusted by the human to change the grasp configuration, i.e., to change the possible value of the rest lengths. Once recording has started, the stiffness is increased linearly from the initial value to 125% of the initial value with 10% random noise. Both the change of initial grasp configuration and the increase of grasp stiffness in the experiments are aiming at increasing the variety of our collected data.

Table 4.1: The weights of different objects during training

object	object weight(g)				
cola can	16	38	50	66	80
food box	10	26	52	68	88
box	34	56	76	99	121
cup	10	49	98	135	155

The recording procedure lasts 25 seconds for each trial and the whole trial is repeated 5 times for each object weight. Fig. 4.8 provides an outline of the data collection. In total, we collected $4 \times 5 \times 5 \times 25 \times 100 = 250000$ positive examples. For each

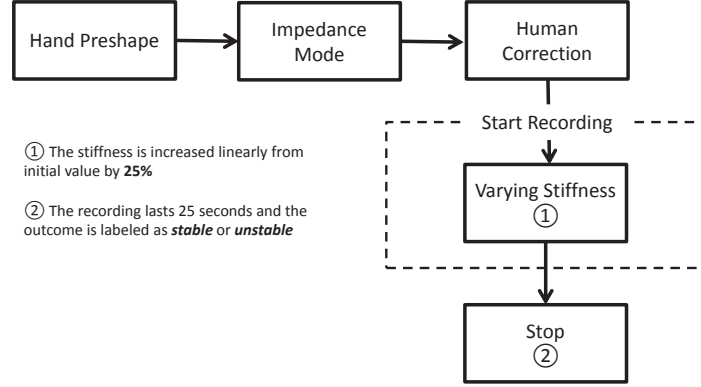


Figure 4.8: The data collection procedure. Only one hand preshape is used in our experiment, as shown in Fig. 4.6. The object is put inside the hand and the control mode is changed to impedance mode. One or more fingertips’ position is locally adjusted to set an initial stable grasp. This step is also required since we want to vary the rest length for each trial.

object, we also collected several negative examples by either setting the initial stiffness to a very small value or filling much more pepper into the objects. In total, we have 37500 negative examples.

4.5.2 RESULTS FOR GRASP STABILITY ESTIMATION

The dataset is divided into training and test sets. For the first three objects (cola can, food box, box), one weight is selected randomly and the data collected with this weight is used as the test set. For the cup, all the collected data are grouped into test set. All the negative datapoints are also used for testing. The tactile data has originally $19 \times 3 = 57$ dimensions and the dimensionality is reduced to 8 dimensions based on PCA, hence in total the data has $8 + 3 + 3 = 14$ dimensions.

For training the model defined by (4.4), the number of Gaussian components, m , is selected using *Bayesian Information Criteria* (BIC) and set to 16 in our experiments, $m = 16$. The other parameters in GMM are learned using the EM algorithm. Figure 4.9 shows the learned stable grasp region projected onto the first two principal components of the tactile reading. Examples of stable grasps and unstable grasps are given for each object. As shown in Fig. 4.10, in these projected lower dimensions, some unstable grasps can be easily separated, such as unstable grasp 2, (grasp of food box). This grasp is unstable because one finger (finger 2) loses contact with the object, see the right upper picture in Fig. 4.9. But in most other cases, when the object starts to slip or tilt slowly from the hand, these unstable grasps cannot be easily separated from stable grasps in the projected lower dimensions. We need to use all the dimensions to compute the marginal likelihood according to (4.4) and compare it with the threshold te .

In order to select the threshold te in (4.4), we vary the value of te in (4.6) with $a = -708.4$ and $b = -35.30$ (The logarithmic likelihood is used here). The ROC curve for the test dataset is shown in Fig. 4.11. The variance on the curve is calculated

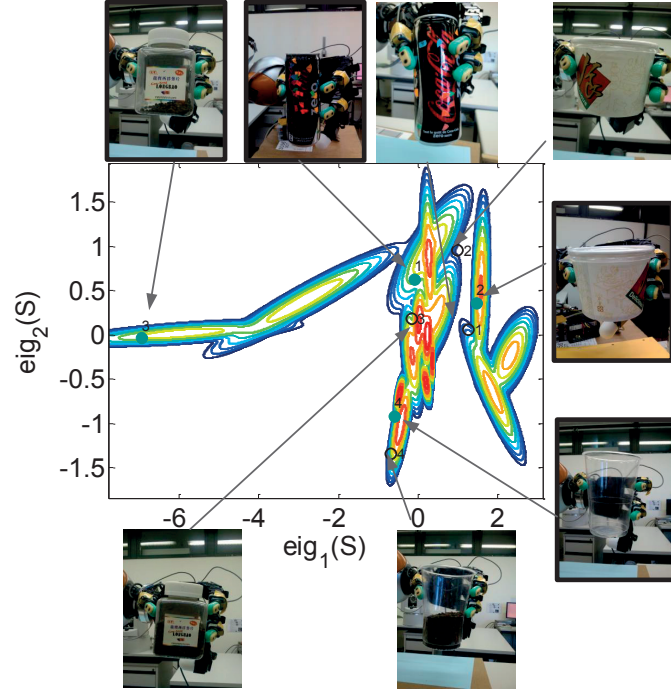


Figure 4.9: The density distribution of the learned GMM model. The axes correspond to the projection of the tactile sensing S on the first two principal components. Contours correspond to the isocurve with constant marginal likelihood value $p(S|\Omega)$. Solid dots denote the stable grasps and circles denote the unstable grasps.

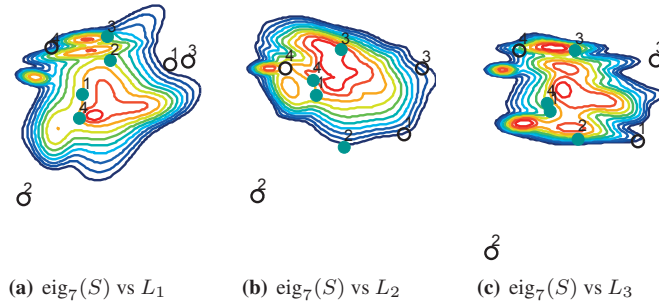


Figure 4.10: The density distribution of the learned GMM model projected into lower dimensions. The X-axis corresponds to the 7th dimension of the tactile readings and the Y-axis corresponds to the dimension of L_1 , L_2 and L_3 , respectively. Contours correspond to the iso-curve with constant marginal likelihood value $p(S|\Omega)$. Solid dots denote the stable grasps and circles denote the unstable grasps.

based on 5 fold cross-validation.

For our application, false positives should be avoided at all costs, as they correspond to the cases where an unstable grasp is mistaken for a stable grasp. For this reason, we chose a very high te value which corresponds to $TPR = 82.27\%$ and $FPR = 15.01\%$. This result is comparable with the results obtained in (Bekiroglu et al., 2011) for unknown objects.

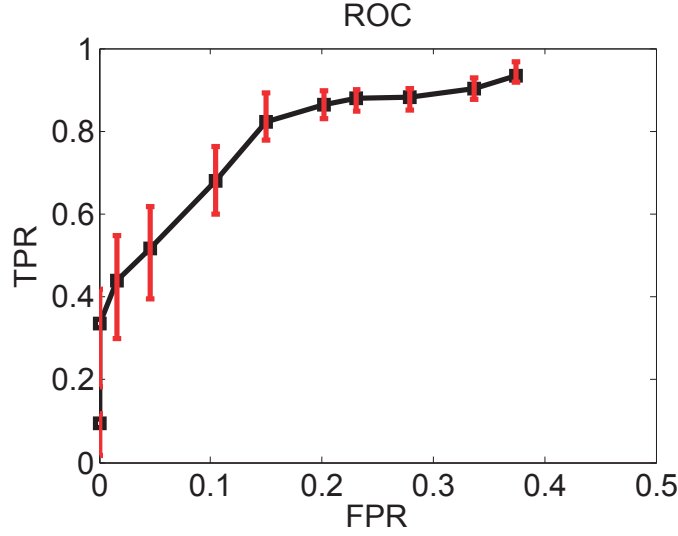


Figure 4.11: ROC curve to select the threshold te .

For comparison, the SVM for one-class classification is also used to predict the grasp stability. We use the RBF (radial basis function) kernel and the parameters are selected using multi-scale grid search. The best performance SVM can obtain is: $TPR = 84.66\%$ and $FPR = 29.72\%$. Although the true positive rates are similar, the SVM has a higher false positive rate. This can be explained by the fact that one-class SVM only tries to separate the training data with origin in the feature space (Chandola et al., 2009), which is a considerable loose constraint compared with a GMM that attempts to model the density of the stable grasp region.

4.5.3 RESULTS FOR GRASP ADAPTATION

To test the validity of our grasp adaptation strategy, first we only use the collected negative datapoints to demonstrate that these grasps can be predicted to be unstable grasps. Also, depending on their similarities with the training dataset, different adaptation strategies should be able to react. For all the 37500 negative datapoints, only 8.70% of them are misclassified as stable grasps. 33.70% of them require the impedance adaptation while 57.60% of them require the local exploration.

We also tested our grasp adaptation approach on the real robotic hand with different objects, see Fig. 4.12-4.17. When the object weights are changing by adding pepper or disturbed by a human⁶, the grasp adaptation strategy is triggered to keep the grasp stable, either by varying the grasp stiffness or by changing the location of finger 1. Taking the test of the cup as an example, see Fig. 4.13, with the increase of the weight, first the stiffness is increasing until the distance computed from (4.7) is larger than 2. At the second stage (shown as red dots), the finger 1 starts to explore local area until the query point is close again. Finally, the stiffness is changed again to find a stable

⁶We use human perturbation to simulate the change of the object weight.

grasp.

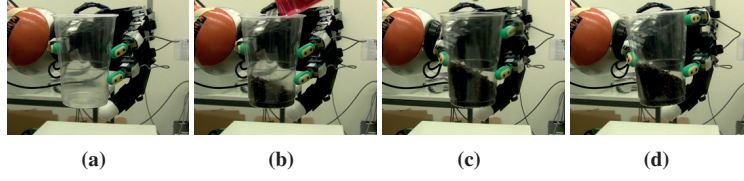


Figure 4.12: A snapshot for the experiment on the cup. (a) The initial grasp stiffness is set manually. (b) Pepper was poured into the cup to change the cup's weight. (c) Finger 1 is changing its location for local exploration. (d) The new grasp configuration after the exploration of finger 1.

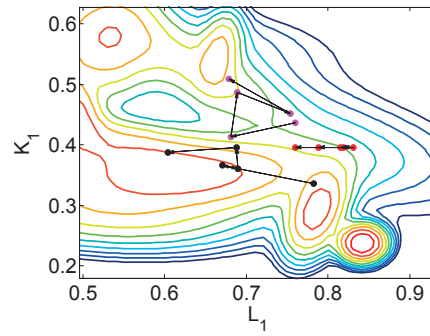


Figure 4.13: The joint density distribution of the dimensions L_1 and K_1 ; The black dots correspond to the impedance adaptation stage for the testing of the cup (Fig. 4.12(b)). Red dots correspond to the local exploration stage and magenta dots correspond to the impedance adaptation after local exploration. For each stage, we only plot 5 datapoints.

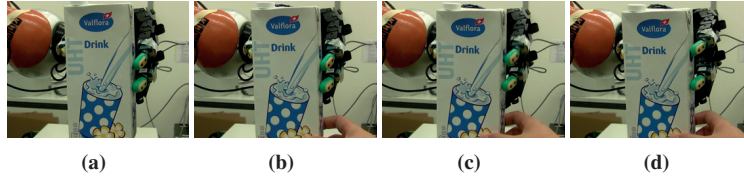


Figure 4.14: A snapshot for the experiment on the milkbox. A human is pulling the milkbox downwards (b) and at the time shown in (c), finger 1 starts to change its location in order to keep the stability of the grasp.

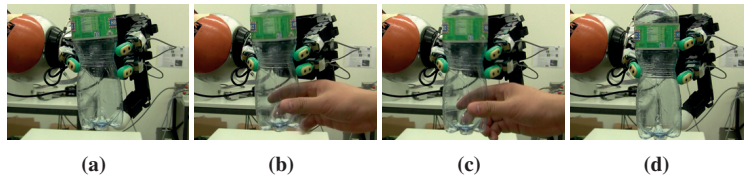


Figure 4.15: A snapshot for the experiment on the water bottle.

To quantify the results of our grasp adaptation strategy, we compare the maximal object weights that the grasp can support with and without the grasp adaptation strategy,

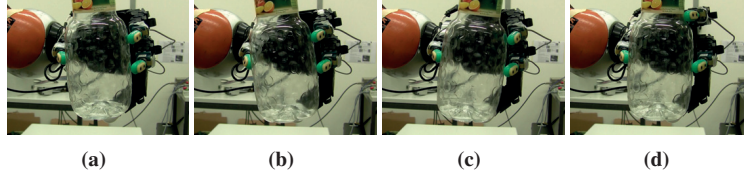


Figure 4.16: A snapshot for the experiment on the juice bottle. A human is trying to push the bottle downwards from the top of the bottle (b) and finger 1 starts to adapt its location at the moment shown in (c).

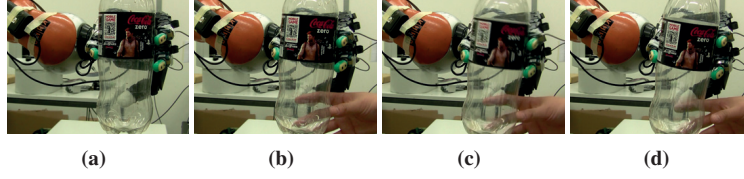


Figure 4.17: A snapshot for the experiment on the cola bottle.

see Table 4.2. We use the same initial grasp configuration and grasp stiffness for each object as in the data collection procedure, i.e., the setup for data collection in the first column of Table 4.1. The object weight is still varying by adding pepper and the object is considered as unstable once there is a steady noticeable slippage between the fingertips and the object. Each object is tested five times, both with grasp adaptation and without grasp adaptation.

In Table 4.2, each row corresponds to the average value of the maximal object weight (grams) that the grasp can support for each object, with (with) and without (w/o) grasp adaptation. The standard deviation during the five experiments is also computed and given as $mean \pm std$. The comparison of the results shows that the grasp adaptation can have a significant improvement on the maximal weight that a grasp can support.

Table 4.2: The comparison of the supported object weights (with vs. w/o grasp adaptation).

obj.	cola can	food box	box	cup
w/o	17.2 ± 1.92	12.8 ± 0.84	37.2 ± 2.59	15.0 ± 2.55
with	69.0 ± 6.52	84.0 ± 3.80	121.2 ± 9.20	146.4 ± 5.46

4.5.4 DISCUSSION

In our controller, since the virtual frame does not assume prior information about the object's shape, this may lead to problems for local exploration. The exploring finger 1 may move out of the object surface after the local exploration, although this problem can be avoided by detecting the loss of contact and moving the finger back to the previous contact position. Also, instead of using only finger 1, other fingers can also be used for local exploration, which will be especially useful when the object is grasped by more than 3 fingers. This requires a more complicated exploration strategy as well as a grasp planning strategy that takes each finger's adaptability into account during

the planning stage. In next section, we will present a unified framework that integrates a grasp planning algorithm with our in-hand adaptation strategy.

Another issue is about the implementation of local exploration. At present, to do the local exploration, we only implemented a fingertip impedance controller to move finger 1 to the desired position. However, we found that in practice it is not so trivial to choose the proper parameters for the fingertip impedance controller. As shown in Fig. 4.12 and Fig. 4.16, the finger 1 may “jump” to the desired position, which sometimes leads to unstable grasps due to either the loss of contact during jump or the large impact during contact. It is more reasonable to implement a contour following controller that can move the exploring finger (finger 1) to the desired position by following the object’s contour. However, since the object is grasped and supported by the other fingers during the exploration, the force that the exploring finger applies on the surface should be adapted to the stiffness of the grasped object.

At present, we didn’t consider any constraints from the hand or finger, such as kinematic constraints. It could be possible that the finger can not move to some desired positions due to these constraints. In some cases, even when the finger can move to the desired position, an exploratory path should be specified beforehand or replanned online. For example, one could move the finger under the handle of the cup. These strategies can be integrated with our grasp adaptation framework. However, a thorough investigation of these exploration strategies is beyond the scope of these thesis.

4.6 INTEGRATION WITH GRASP PLANNING

As discussed in 4.2.1, our dynamic grasp adaptation framework can close the loop between grasp planning and control through stability estimation and grasp adaptation. In the grasp adaptation strategy in Section 4.4, we don’t assume any knowledge from the object’s shape and only rely on the tactile sensing to guide the adaptation. This makes sense if the object’s shape is completely unknown or the gaze of vision is not placed on the object. However, in many real-world scenarios, the robots in fact have some priori information of the object’s shape. For example, during the grasp planning stage, usually partial or complete object point cloud can be obtained from vision system or from the object’s model in simulation. The knowledge from object’s shape should be used to improve the performance of the grasp adaptation system. In this section, we present a unified framework that integrates the grasp replanning into the adaptation strategy, resulting in a more informed adaptation strategy (i.e., finger gaiting) that takes the hand reachability (adaptability) and the knowledge about object shape into account. This whole integrated system is based on a joint work with Kaiyu Hang and Danica Kragic at KTH ([Hang et al., 2015, 2014a](#)).

4.6.1 SYSTEM SETUP

The components of the whole integrated system is shown in Fig. 4.18. Compared with Fig. 4.3, besides the components of off-line training and on-line execution, there is

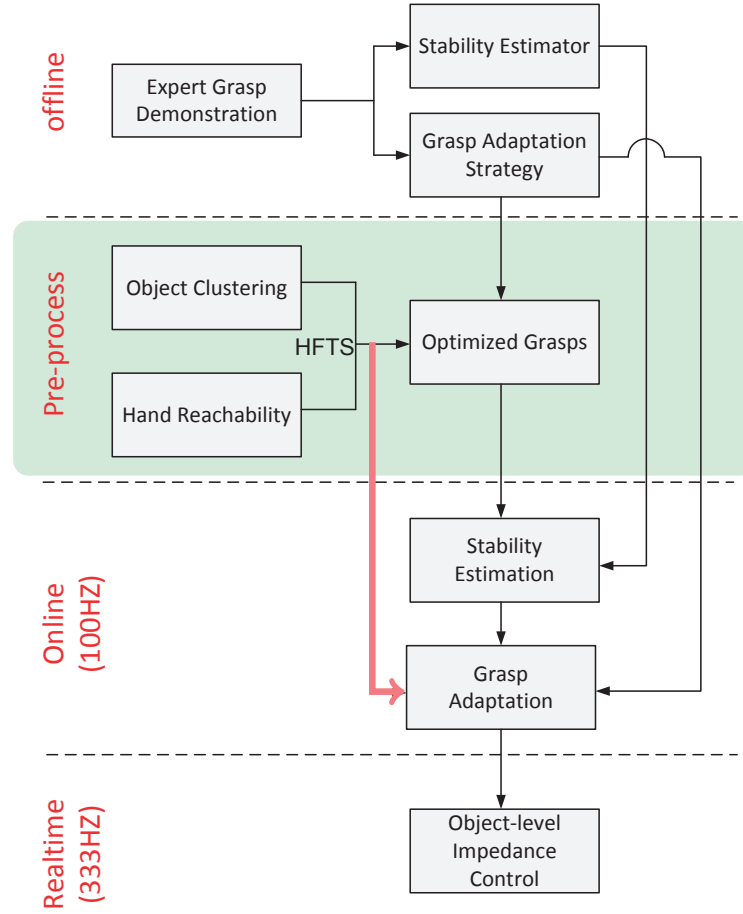


Figure 4.18: The components of the integrated system. In the pre-process stage (see the light green block), the object shape information together with the hand reachability are processed and encoded as a Hierarchical Fingertip Space (HFTS) (Hang et al., 2014a). This representation enables fast on-line grasp replanning, resulting in a more informed local exploration (finger gaiting) that takes the hand reachability and object shape into account (see the red arrow).

one more layer in the integrated system – the pre-process (pre-grasp) stage, where the knowledge about object shape and the hand reachability are addressed and encoded as a Hierarchical Fingertip Space (HFTS). Fingertip space represents a finite set of contacts on an object surface that are locally flat and large enough for a fingertip, as shown in Fig. 4.19. To be more specific, initial fingertip locations are determined by optimizing grasp stability and adaptability simultaneously using a hierarchical discretization of the object surface (see the light green block in Fig. 4.18) and then an object-level impedance controller is used to balance grasping forces and the grasp stability estimator is launched to monitor the status of the grasp. If a large disturbance occurs, the grasp is adapted by finger gaiting to maintain grasp stability. The new fingertip location is computed using an on-line local optimization in the HFTS to simultaneously search for a similar (in terms of Eq. (4.10)) and reachable grasp. The grasp reachability determines whether the new combination of fingertip’s locations can be realized by the given hand or not, as defined in Hang et al. (2015).

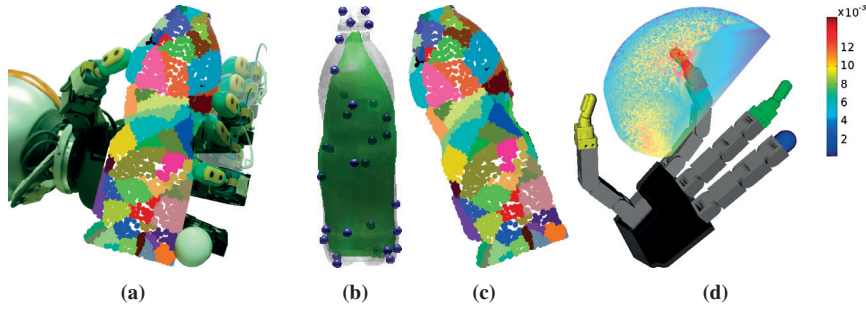


Figure 4.19: (a) A visualization of the proposed Hierarchical Fingertip Space concept. (b) The centers (blue balls) of the hierarchical discretization of the object surface are selected using Gaussian Processes (Sommer et al., 2014). (c) The final refined Hierarchical Fingertip Space. (d) Grasp adaptability for fingertip 1. Adaptability is computed for fingertip positions sampled in joint space. The colored volume shows fingertip positions and their corresponding adaptability values.

4.6.2 EXPERIMENTAL RESULTS

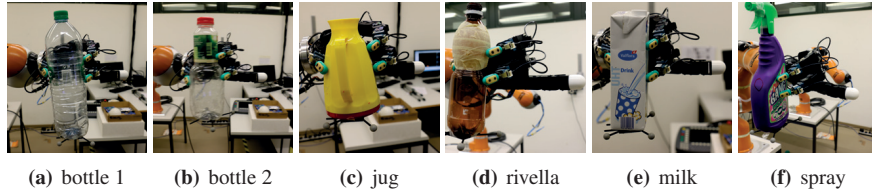


Figure 4.20: Six example objects used in the evaluation: there is both variation in global geometry as well as local surface properties.

The performance of the integrated system is evaluated using six objects shown in Fig. 4.20, which are tracked using the OptiTrack⁷ real-time motion tracking system.

⁷<http://www.naturalpoint.com/optitrack/>

The results presented below demonstrates the performance of the integrated system for grasp adaptation increases significantly.

For the evaluation, we run two sets of experiments: 1) We continuously increase the objects' weight by filling them to evaluate the maximum weight each grasp can withstand, and 2) We shake the grasped objects by linearly increasing acceleration in different directions to evaluate the maximum acceleration each grasp can withstand. For comparison, we conduct the same experiments without any grasp adaptation and on the system proposed in in Section 4.4 which does not consider object shape information when relocating fingertips.

TESTING MAXIMUM WEIGHT

For each object, we execute the best out of 100 grasps generated from our planning algorithm. We then gradually fill object with black pepper beans and record the maximum weight the grasp can withstand. The maximum weight is reached when the stability estimator predicts unstable grasp for more than 2 seconds or if the object drops. We repeat this test for each grasp 5 times and summarize the results in Table 4.21. For comparison, we perform the same tests for the system proposed in Section 4.4 and a system without adaptation.

Object	Weight	Without	With adaptation	Improved
bottle1	34	55.1 ± 7.11	153.1 ± 12.31	165.3 ± 13.27
bottle2	39	62.8 ± 6.63	102.3 ± 13.38	121.3 ± 9.91
jug	112	125.3 ± 14.90	147.4 ± 9.62	162.1 ± 13.12
rivella	24	36.0 ± 6.96	76.5 ± 9.4	92.7 ± 7.45
milk	34	63.5 ± 8.20	151.8 ± 7.24	157.4 ± 8.35
spray	63	75.7 ± 7.21	102.2 ± 6.02	121.6 ± 7.15

Figure 4.21: The comparison of the supported object weights(Unit: gram). **Without:** without grasp adaptation; **With adaptation:** with grasp adaptation in Section 4.4; **Improved:** the new adaptation approach in this section.

Naturally, the system without any adaptation performs the worst and the integrated system outperforms the system with adaptation from Section 4.4. This is since our system: i) takes into account grasp reachability during the exploration, and ii) the new location is computed in the *HFTS*, thus ensuring it is valid, avoiding moving out of the object surface, and iii) considers two fingers for gaiting, resulting in increased flexibility.

SHAKING TEST

External disturbances, such as collision, may occur once a grasp has been executed. To evaluate the proposed system, we designed a shaking test. We first execute the best out of the 100 generated grasps for each object from our grasp planning algorithm, and then rotate the arm to the configuration shown in Fig. 4.22. Thereafter, we start to shake the arm in either vertical or horizontal directions while linearly increasing the acceleration from $2m/s^2$ to $8m/s^2$. The shaking magnitude is limited to $10cm$ in either

$J0$	$J1$	$J2$	$J3$	$J4$	$J5$	$J6$
-30°	30°	2°	-60°	-20°	0°	-60°

Init. $K = (K_x, K_y, K_z)$	Horizontal Acc.	Vertical Acc.
$(12, 2, 2)$	$2m/s^2 - 8m/s^2$	$2m/s^2 - 8m/s^2$

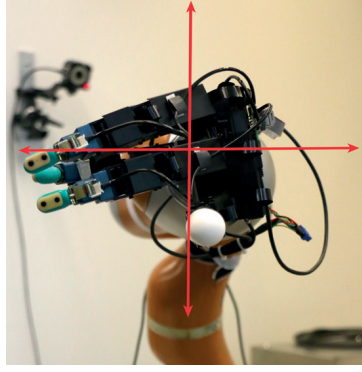


Figure 4.22: The setup of grasp shaking test, in which the arm shakes each grasp in horizontal and vertical directions. $J0$ to $J6$ are the joint values for the initial pose of shaking test. When shaking horizontally, the shaking direction is fixed to be perpendicular to the palm.

directions, which means that the hand is accelerating in the first $5cm$ and decelerating in the second $5cm$. After every period of shaking, we increase the acceleration by $1m/s^2$ and therefore have 14 shakes for every test.

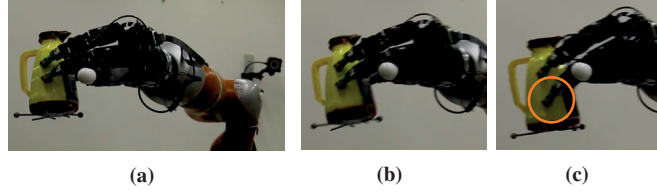


Figure 4.23: The snapshots for the horizontal shaking of the jug.

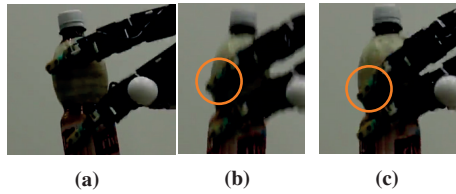


Figure 4.24: The snapshots for the vertical shaking of the rivella.

Similarly to the supported weight test, we evaluate each grasp by measuring the maximum acceleration it can withstand. The criterion is similar: the maximum acceleration is recorded when the grasp is predicted as unstable for more than 2 seconds or if the object drops. The shaking test is conducted in both directions separately and on each object by filling it with $10g$, $20g$, $30g$, $40g$ and $50g$ black pepper beans. Each

test is repeated 5 times. For comparison, we also conduct the same test for a system without adaptation and the system proposed in Section 4.4.

Some snapshots during the testing of the horizontal shaking of the jug and the vertical shaking of the rivella is shown in Fig. 4.23 and Fig. 4.24 respectively. The quantitative experimental results are summarized in Fig. 4.25. The X-axis is the additional weight we added to the object and the Y-axis is the maximum acceleration that the grasp can withstand. If the maximum acceleration rate is $8m/s^2$, it means that the grasp has been kept stable during the test. On the other hand, if the maximum acceleration rate is $2m/s^2$, it means that the grasp could not withstand any shaking. We can see that our system outperforms both the system without the adaptation and the system proposed in Section 4.4. The strength of our approach is that we ensure that the finger gaiting has resulted in an actual contact with the object which is not the case in Section 4.4. In addition, the flexibility of gaiting two fingers provides additional strength.

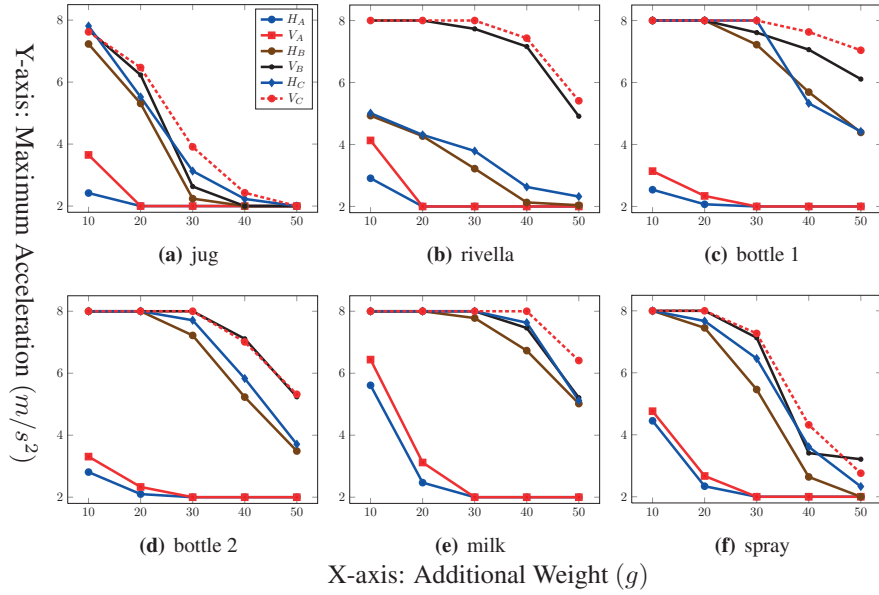


Figure 4.25: Results of shaking tests on grasps shown in Fig. 4.20. In the legend, H and V refer to horizontal shaking test and vertical shaking test respectively. A , B and C refer to 3 grasp strategies: grasp without adaptation, grasp adaptation in Section 4.4 and the grasp adaptation proposed in this section.

4.7 DISCUSSION AND SUMMARY

In this chapter, we first presented a new framework for dynamic grasp adaptation under physical uncertainties of the grasped objects (e.g., weight, friction, perturbation), without the availability of the object’s shape information. The adaptation strategy is derived from an object-level impedance controller and learned with training data that is generated using a real robotic hand. We first formulated the grasp stability estimation as a one-class classification problem. A Gaussian Mixture Model is used to model the region of the stable grasps. During the grasp execution, if a grasp is predicted to

be unstable, two different adaptation strategies, i.e., impedance adaptation and local exploration, are selectively triggered according to the similarity between the current unstable grasp and the examples in the stable grasp regions.

Moreover, when the object's shape is known, the proposed probabilistic model for grasp stability estimation and adaptation has shown its feasibility in closing the loop between grasp replanning and control. We have evaluated the performance of the proposed system quantitatively and shown that the proposed system significantly improves the robustness of grasp execution, compared with grasp execution without any adaptation strategy. To the best of our knowledge, this is so far the only system that accomplishes grasp synthesis, stability estimation, on-line replanning and in-hand adaptation in a unified framework, as well as evaluating this on a real physical system with extensive experiments.

While we have demonstrated that our dynamic grasp adaptation approach can largely improve the robustness of the grasp, there are still limitations that need further improvement. First, we only tackle with precision fingertip grasp in this work. It is clear that humans can use other parts of the hand to stabilize the grasp, such as the palm or the side parts of each fingers. Therefore, one potential extension of this work will be incorporate other grasp types into the grasp adaptation strategy. Second, during the design of the grasp experience, we only use the tactile sensing to encapsulate the object dynamic force and the perturbation. It is possible to integrate other sensory modality (e.g., force sensor at wrist, vision) into the definition of grasp experience. Lastly, the whole system is still largely limited by the hardware of the hand. It will be beneficial to design a robotic hand that can accommodate part of the perturbations by its structure or its passive dynamics, such as the hands in (Deimel and Brock, 2015; Odhner et al., 2014; Ciocarlie et al., 2014).

Besides the above-mentioned practical issues to improve the robustness of the grasp, there are several theoretical challenges arise from our dynamic grasp adaptation approach that deserve much more efforts in future work. First, due the separation of stiffness adaptation and the finger relocation, the whole system is hybrid and discontinuous. As a result, instabilities can arise (although we did not observe instabilities during our experiments). How to ensure the stability of the whole hybrid system is a very difficult but important problem. The second issue is concerned with the controllability of the fingers in the virtual frame. In our current work only one finger is allowed to relocate during grasp adaptation. However, if more fingers are used, one must devise a gaiting strategy to determine the kinds of grasp configurations and thus the corresponding virtual frames can be achieved given a series of different finger gaiting, which shares some similarities with gait controllability of legged robot (Goodwine and Burdick, 1998). The last issue relates to the adaptation of impedance. In addition to the stiffness adaptation, it also possible to vary the damping parameters in real-time during robotic grasping and manipulation. This can be a very important factor to maintain the stability in the dynamic sense (Kronander and Billard, 2016).

TASK MANIFOLD FOR CONSTRAINED OBJECT MANIPULATION

5.1 INTRODUCTION

Until now, the objective of our grasping controller was to ensure grasp stability. In other words, the manipulation task considered so far stopped at holding the object securely in the hand. However, there are many other tasks (e.g., inserting a bulb, drawing with a pen, cleaning a table with a grasped sponge, etc.), that requires the grasped object to satisfy some additional constraints in order to accomplish the task. We term these tasks as Constrained Object Manipulation (COM) because the task constraints are imposed directly on the grasped object itself rather than on the robot's configuration. These constraints can be either physical constraints (e.g., the bulb can only be rotated around the axis of the socket) or virtual constraints (e.g, the object is asked to follow a prescribed path in the air during dexterous manipulation). Some of these task constraints may be easy to specify analytically in the form of differential and algebraic equations, while others can be stringent to write down. Moreover, due to the uncertainties from environment, e.g., external forces acting on the object or imprecise contact dynamics, compliant controller and adaptive strategies are required for performing these tasks. To address this issue, in this chapter we propose a manifold learning approach for COM, which is declined along 3 steps:

1. A task manifold is learned off-line that encodes the constraints of COM as a vector field. This represents a finite-dimensional subspace of the relevant task variables and allows efficient path planning on the manifold.
2. An adaptation strategy is devised and modeled using Gaussian Process Regression, to embed a model of uncertainties from environment. This model is then used to bring the hand back to the manifold, and hence to ensure that the tasks constraints are satisfied.
3. The learned manifold is further integrated with an object-level impedance controller that offers a task-consistent adaptation during on-line execution, where the proper control parameters are learned from human demonstration with the local geometric information of the manifold.

This chapter's sections on learning of task manifold for COM and the associated adaptation strategy have appeared in (Li et al., 2015b). The part on learning the proper impedance parameters for an object-level impedance controller was partially published

in (Li et al., 2014b). Here we further provide the task manifold during the impedance learning, leading to impedance parameters consistent with the task requirements.

Related Publications:

- (Li et al., 2014b) Li, M., Yin, H., Tahara, K., and Billard, A. (2014b). Learning object-level impedance control for robust grasping and dexterous manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6784–6791
- (Li et al., 2015b) Li, M., Tahara, K., and Billard, A. (2015b). Learning task manifold for constrained object manipulation. (In submission)

5.2 PROBLEM STATEMENT

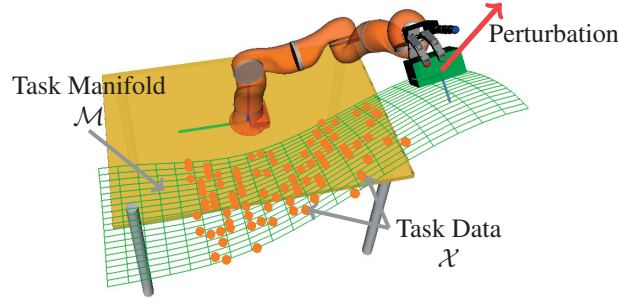


Figure 5.1: An typical example of constrained object manipulation: an object grasped by a robotic hand-arm system, is required to follow a surface which may be unknown or difficult to represent as an analytical form. Two main components are generally required to accomplish a COM task: (1) A task manifold \mathcal{M} to represent the task and generate feasible path fulfilling all the task constraints. (2) A robust adaptive controller to execute the planned path in case of disturbances from imprecise contact dynamics or external perturbations.

Almost all the robotic grasps are goal-directed (see Fig. 2.2), that is, the robot is expected to perform some tasks with the grasped object. Among these expected tasks, COM is of particular importance as it specifies the desired behaviors in terms of the object motion directly, which includes a large variety of everyday tasks, ranging from the fine scale dexterous manipulation (insert a bulb, open a bottle cap) to the large scale contour-following task (clean a table, polishing). As shown in Fig. 5.1, at least two major components are required to accomplish the COM task: (1) a *task representation* to generate feasible path fulfilling all the task constraints; (2) a robust controller to execute the plan and *adapt* to the potential disturbances. In this chapter, we propose a manifold learning approach to encode the COM task as a vector field, which enables an intuitive task-consistent adaptation built on an object-level impedance controller.

In the following section, we review the state-of-the-art approaches for COM in a broader context – constrained manipulation, in relation to our approach to addressing the difficulties in planning and control for COM. Depending on the robots and the tasks, constrained manipulation may entail a variety of constraints at different levels, such as

joint limits, collision avoidance, pose constraints of end-effector, desired interaction with external environments. They arise in applications such as carrying a cup of water, inserting a bulb, two arms carrying a common object, or polishing a workpiece etc. Though ubiquitous, these constraints presents significant challenges for the planning and control of constrained manipulation.

5.2.1 CONSTRAINED MANIPULATION PLANNING

Two main issues make the planning of constrained manipulation difficult. First, the task constraints are difficult to represent. The previous works usually assume a priori knowledge, usually in an analytic form of these task constraints (Stilman, 2010; Berenson et al., 2011; Rodriguez et al., 2008; Yao and Gupta, 2005; Oriolo et al., 2002). For example, Stilman (2010) defines a motion-constraint vector to analytically express many common task constraints, including fixed-frame constraints, kinematic-closure constraints, and path or surface constraints. The concept of Task Space Region (TSRs) has been proposed to allow robots to plan in the presence of constraints on the end-effector pose (Berenson et al., 2011, 2009a). However, for robots working in unstructured environments, an explicit or analytical representation may be not available beforehand. For example, the surface to be polished is unknown or difficult to build an analytical representation. For COM, this issue is more serious, as the object used to perform the task can also have uncertain or even unknown shape and size, making it difficult to specify the constraints analytically. Taking Fig. 5.1 as an example, if the object shape is unknown, it will be impossible to write down the constraints between the object and the surface in an analytical form.

To address this issue, in this work we adopt a different approach that constructs the task constraints from a set of observed data points satisfying the task requirements. There has been an large amount of work within the field of learning from demonstration (LfD) (Billard et al., 2008). To be specific for constrained manipulation, most previous sampling-based approaches purely use the demonstration to prune the search space to speed up the sampling (Phillips et al., 2013; Berenson et al., 2012; Phillips et al., 2012), which means that the task constraints are not extracted for further use. Other approaches include an explicit construction of the constraint manifold in the form of an experience graph (Şucan et al., 2012; Phillips et al., 2012) or a local atlas (Porta et al., 2011; Havoutis and Ramamoorthy, 2013). Our work follows the later approaches to build a vector field that represents the tangent space of the task manifold, see Section 5.3. The advantage of our approach comes from the fact that (1) it directly operates in the object’s configuration space and not in the higher-dimensional robot’s configuration space, as most of the existing methods do; (2) it has a more meaningful definition of the point-to-manifold distance and also a fast prediction method of the distance.

Second, the task constraints often reside on a low-dimensional manifold, which can not be directly sampled using rejection sampling (the sampling method used by most sampling-based planners), because the probability of finding samples on the manifold will tend to 0. To address this issue, a large number of more sophisticated sampling

techniques are proposed (Kavraki et al., 1996; Oriolo et al., 2002; Yao and Gupta, 2005; Berenson et al., 2009b; Porta et al., 2011; Stilman, 2010; Suh et al., 2011; Ko et al., 2014). Most of these approaches require a distance function to determine how far a given point is from the manifold, and also the projection direction from the given point to the closet point on the manifold. The closest point is usually found through an iterative gradient-descent process, which can be time-consuming for fast re-planning. Moreover, these approaches compute the projection from scratch every time without exploiting previous experiences. To this end, we propose a new approach that uses previous experiences to learn the distance function as well as the projection direction off-line through Gaussian process regression, which can be updated later when more experiences are available during on-line execution, see Section 5.4 for more details.

5.2.2 CONSTRAINED MANIPULATION CONTROL

Most of the works mentioned above only need kinematic control where the robot end-effector is moved along the planned path. However, COM entails the control of interactions of the object with the task manifold (environment). Achieving the task goal often requires the robot to comply with its environment, that is, to react to perturbation (i.e., external forces acting on the objects), or adapt the object’s motion to uncertainties of the environment. Two most common control strategies that accommodates the physical interaction with the environment are force and compliance control (Siciliano and Khatib, 2008; Siciliano et al., 2008). The *hybrid position/force control* is the basic form of force control that assigns position and force control to two subspaces that are consistent with task requirements. However, the separation of the force-controlled and position-controlled subspace is not a trivial task, and also a reliable force feedback for COM is not feasible as the object is usually grasped by the hand, rather than fixed with the force-torque sensor at the end-effector of robot arm. To address the later issue, a large number of applications actually design specialized fixture to hold the object rigidly and thus the force-torque at the wrist can still be used. However, the specialized fixture obviously possesses a much lower dexterity and flexibility than that of a dexterous robotic hand.

As an alternative approach to accommodating interaction, compliance or impedance control reacts to the external forces such that a given relationship is held between the applied force and the displacement. For COM, it is a natural choice to specify this desired relationship in object’s frame. Therefore, an object-level impedance controller is usually adopted to assign corresponding impedance to move the object to desired configuration (Wimböck et al., 2012; Tahara et al., 2010; Wimbock et al., 2008; Schneider and Cannon, 1992). These approaches typically assume a predefined reference trajectory, where the impedance controller is merely used in an inner loop to track the reference trajectory at each time instant. However, in many applications, the actual timing in the reference trajectory is unimportant, compared to the coordination requirements between various degrees of freedom. COM is one of the typical example, where the key requirement is to ask the object to follow the task manifold. The authors in (Li and

Horowitz, 1999, 2001a,b) proposed to encode these tasks (called contour following in their paper) in terms of velocity fields, and designed a passive velocity field control (PVFC). Some variants of PVFC are proposed in (Kishi et al., 2003; Saitoh et al., 2003; Cheng and Wang, 2009) to deal with uncertainties from the environmental geometric constraints and to construct the task velocity field. However, these approaches do not specify how to design the proper control parameters on these task velocity fields.

On the other hand, impedance selection for grasping and robot manipulation has been studied independently. Early works use analytical approaches to qualitatively (small or large stiffness in different directions) determine the proper impedance (Shimoga and Goldenberg, 1991; Kim et al., b,a). Learning-based approach are used later that attempts to extract proper impedance from available task information. In (Yang and Asada, 1996), the impedance selection is formulated as a model-based reinforcement learning problem, where the impedance parameters can gradually update to improve the task performance. In (Buchli et al., 2011), the authors constructed a variable impedance controller with a model-free, sample-based reinforcement learning method. However, the reward function needs to be carefully defined to capture the essence of the task. Sikka and McCarragher (1997) presented a method that can learn the robot end-point stiffness of contact tasks from human demonstration. An on-line, incremental algorithm has been proposed in order to learn varying end-point stiffness from human demonstration (Kronander and Billard, 2012). These works only addressed the impedance learning problem in the robot’s task space, rather than in the object’s frame.

For object constrained manipulation, we propose a learning-based approach to extract the desired varying object-level impedance from human demonstration (Li et al., 2014b). Recently, in collaboration with Bidan Huang (a visiting PhD student from Bath University), a modular approach is introduced to learn an implicit manipulation strategy from human demonstration, which can essentially modularize an adaptive control strategy for a large variety of tasks with very different task conditions (Huang et al., 2015). However, this approach did not take the task constraints into account explicitly. Therefore, the impedance selection and task encoding (including path planing) are still decoupled, which can result in impedance parameters inconsistent with the task requirements (Ott, 2008; Siciliano et al., 2008). Our work in this chapter combines the idea to encode the COM task as a vector field with impedance controllers, by offering an object-level impedance controller on the vector field. Both the task manifold and the proper impedance parameters are learned from human demonstration.

5.3 TASK MANIFOLD LEARNING

As discussed above, our approach of learning task manifold is related to some previous investigations on representing the task model as an approximated vector field (Havoutis and Ramamoorthy, 2013; Ko et al., 2014; Mussa-Ivaldi and Giszter, 1992). Consistent with this view and given the fact that COM general forms a low-dimensional constraints manifold, we regard the issue of COM as a task manifold learning problem. Our approach for manifold learning, building on (Dollar et al., 2007), attempts to learn

an approximation of the tangent space at each point, by assuming that the neighboring data points are located in the same tangent space.

Given a dataset of D dimensions $\mathcal{X} = \{x^i \in \mathbb{R}^D, i = 1 \dots N\}$, which are generated from a manifold of d dimensions, $d < D$. For the example task shown in Fig. 5.1, $D = 3, d = 2$. The objective of manifold learning is to learn a mapping from a point $x \in \mathbb{R}^D$ to the basis vectors of its tangent space $H(x) \in \mathbb{R}^{D \times d}$, as shown in Fig. 5.2(a).

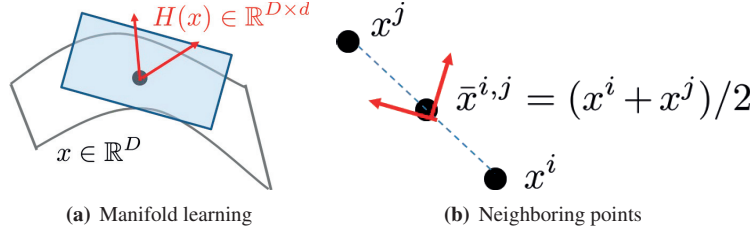


Figure 5.2: Illustration of manifold learning. (a) A manifold is locally represented by the basis vectors of the tangent space, shown as red arrows. (b) We assume the neighboring points are located in the same tangent space, i.e., they can be represented by the same basis vectors.

From the assumption that neighboring data points are located in the same tangent space, we have:

$$\Delta^{i,j} \triangleq x^j - x^i \approx H(\bar{x}^{i,j})\epsilon^{i,j} \quad (5.1)$$

where x^j and x^i are two nearby points and $\bar{x}^{i,j}$ is their center, see Fig. 5.2(b). $\epsilon^{i,j} \in \mathbb{R}^d$ are the coordinates of $\Delta^{i,j}$ in the local tangent space.

For each point in \mathcal{X} , we have a constraint as Eq. (5.1). To learn the basis vectors for each point, we put together all the constraints with two regularization terms, leading to the following objective function:

$$\begin{aligned} \min_{\{\epsilon^{i,j}\}} : & \sum_{i=1}^N \sum_{j \in \mathcal{N}^i} \|H(\bar{x}^{i,j})\epsilon^{i,j} - \Delta^{i,j}\|^2 + \lambda \sum_{i=1}^N \sum_{j \in \mathcal{N}^i} \|\epsilon^{i,j}\|^2 \\ & + \lambda \sum_{i=1}^N \sum_{j,j' \in \mathcal{N}^i, j \neq j'} \|H(\bar{x}^{i,j}) - H(\bar{x}^{i,j'})\|_F^2 \end{aligned} \quad (5.2)$$

where \mathcal{N}^i is the k nearest neighbors of point x^i and $\|\cdot\|_F$ represents the Frobenius norm. The last two terms in the objective function are added to explicitly enforce the smoothness of the learned mapping H . In our application, k is chosen between 3 and 10, depending the task data \mathcal{X} .

With the learning objective function Eq. (5.2), now we need to find a method to parametrize H . In principle, any regression techniques can be adopted to model H , however, the local kernel regression method is used here for its simplicity. We can define H as:

$$H(x) = [\theta^1 \mathbf{f}(x) \dots \theta^D \mathbf{f}(x)]^T \quad (5.3)$$

where $\theta^i \in \mathbb{R}^{d \times p}$ are the parameters to estimate and $\mathbf{f}(x) = [f_1(x) \dots f_p(x)]^T \in \mathbb{R}^{p \times 1}$

are the predefined features with p number of features. The RBF (Radial Basis Function) kernel is used for each feature in $\mathbf{f}(x)$, i.e., $f_i(x) = \exp(-\|x - \mu^i\|^2/2\sigma^2)$.

The centers μ_i are selected using K-means in the training dataset and σ is set to twice of the minimal pairwise distance among all the centers. The parameters θ^i and $\epsilon^{i,j}$ are obtained with EM-like algorithm, see (Dollar et al., 2007) for more implementation details. In order to get out of the local minimum, we restart the optimization procedure several times and choose the parameters that can give the best performance in terms of the objective function defined in Eq. (5.2).

5.4 PLANNING ON TASK MANIFOLD

5.4.1 PATH PLANNING ON MANIFOLD

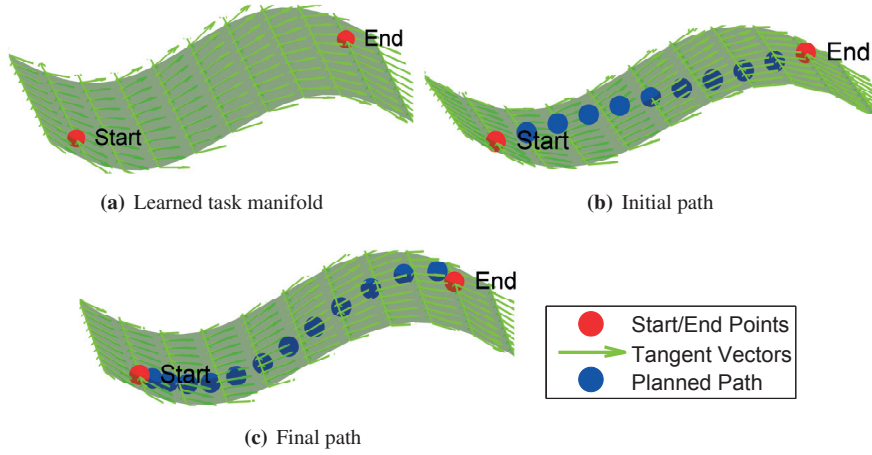


Figure 5.3: An example of path planning on task manifold. (a) An learned manifold with the basis vectors in the tangent space and the start/end points are given on the manifold. (b) An initial path is given by interpolating between the start and end points in the ambient space. (c) A final path is obtained by projecting the initial path onto the tangent space of the learned manifold.

The learned task manifold provides not only a representation of the task but also an framework for the feasible path planning, which are presented in this section. The goal of path planning is to find a feasible path on the manifold that connects the start point and the end point. Given the start and end points, we initialize the path by interpolating between the start point and the end point in the ambient space. Then the interpolated points are recursively projected onto the learned manifold. This procedure is explained in Algorithm 5.1. In Fig. 5.3, we give an example of a 2-dimensional manifold embed in 3-dimensional ambient space. With the learned manifold in Fig. 5.3(a), the initial planned path in Fig. 5.3(b) is projected onto the learned manifold, resulting in a final feasible path shown in Fig. 5.3(c) that satisfies the task constraints.

Note that additional quality criteria can be incorporated to optimize the final feasible path on the manifold. Since this work only focuses on how the manifold can benefit the fast on-line replanning and the impedance learning for COM, we will not address the path or trajectory optimization on the task manifold.

Algorithm 5.1 Path planning on manifold

```

procedure PATHPLANNING
  Input:  $H(q), \delta, q_{start}, q_{end}, \alpha$ 
  Output:  $\mathbf{q} = \{q_{start}, q_2, q_3, \dots, q_{end}\}$ 
  initialization:
  ▷ linear interpolation between  $q_{start}, q_{end}$  with interval  $\delta$ 
     $\mathbf{q} = \{q_{start}, q_2, q_3, \dots, q_{end}\}$ 
    step size  $\alpha = 0.5\delta, \mathbf{q} \leftarrow \{q_{start}\}, q_l \leftarrow q_{start}, q_r \leftarrow q_2$ 
     $err \leftarrow \|q_{end} - q_l\|, q_{next} \leftarrow q_{start}, i = 2$ 
  while  $err > \delta$  do
    ▷ project onto manifold:
     $q_{next} = q_{next} + \alpha H(q_l) * H(q_l)^T * (q_r - q_l)$ 
     $q_l = q_{next}$ 
    if  $i < end$  then
       $i \leftarrow i + 1, q_r \leftarrow q_i$ 
    else
       $q_r \leftarrow q_{end}$ 
    end if
     $err \leftarrow \|q_{end} - q_l\|$ 
     $\mathbf{q} \leftarrow \{\mathbf{q}, q_l\}$ 
  end while
end procedure

```

5.4.2 PATH REPLANNING ON MANIFOLD

After obtaining the path on the manifold, during the execution, the manipulated object may still go out of the manifold due to some unexpected disturbances. Therefore, it is necessary to have an adaptation strategy that can bring the object back to the manifold. One of the natural choices is to project the off-manifold point onto the manifold. However, it is not a trivial task to compute the projection from a given point to a manifold, especially when the explicit representation of the manifold is not available. In the remainder of this section, we will first introduce an iterative gradient-descent process to find the projection point. In next section, to speed up the computation, the distance to the manifold and the projection direction is further modeled through Gaussian Processes.

For the iterative computation, the projection is performed by first finding the closest point on the original planned path, and then moving the closet point in the local tangent space until the minimum distance is achieved, which corresponds to the projection point on the manifold. After finding the projection point, we can reuse the path planning procedure, i.e., Algorithm 5.1, to find a feasible path to the end point on the manifold. This procedure is summarized in Algorithm 5.2. An example of the path replanning after perturbation is given in Fig. 5.4.

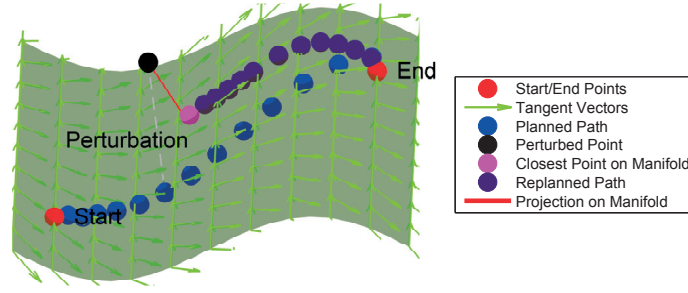


Figure 5.4: The path replanning after perturbation. During the execution of the initial planned path, the system is perturbed to a new point (black point), and this point is projected onto the manifold (magenta point). Starting from this projection point, a new path is replanned on the manifold.

Algorithm 5.2 Path replanning after perturbation

procedure PATHREPLANNING

Input: $H(q), q_*, \mathbf{q} = \{q_{start}, q_2, q_3, \dots, q_{end}\}, \delta$

Output: $\mathbf{q}' = \{q_{proj}, q'_2, q'_3, \dots, q_{end}\}$

initialization:

perturbed point $q_*, \delta = 0.001, err_0 = 1, err_1 = 0$, step size: $\alpha = 0.1, \mathbf{q}' = \{\}$

▷ find closest point on \mathbf{q} :

$$q_{proj} = \underset{q_i}{\operatorname{argmax}} \|q_* - q_i\|, i = start, 2, 3, \dots, end$$

▷ find projection point on manifold:

while $|err_0 - err_1| > \delta$ **do**

$$err_0 = \|q_* - q_{proj}\|$$

$$q_{proj} = q_{proj} + \alpha H(q_{proj}) * H(q_{proj})^T * (q_* - q_{proj})$$

$$err_1 = \|q_* - q_{proj}\|$$

end while

Recall Algorithm 5.1 with new starting point $q_{start} = q_{proj}$

Replanned path: $\mathbf{q}' = \{q_{proj}, q'_2, q'_3, \dots, q_{end}\}$

end procedure

It is noteworthy that the learned manifold locally represents the geometric structure of the task constraints and thus gives a more meaningful metrics of projection. Also, all the procedure of the projection and replanning are performed on-line, making the approach applicable to situations where precise models of object and environment are not available. However, since the projection is an iterative gradient-descent process, it can still be time-consuming if the perturbed point is far away from original planned path. Besides, for every perturbation, the projection is computed without exploiting previous projection experiences. It will be desirable to reuse previous projection experiences to speed up the projection procedure. Most importantly, to know if a perturbed point is on or off the manifold, we also need a fast distance computation method to monitor the state of the object, which is depicted in next section.

5.4.3 LEARNING PROJECTION ONTO MANIFOLD

In this section, we present an approach to predicting the distance to the manifold through Gaussian Processes Regression (GPR). To this end, we reuse the idea of Gaus-

sian Process Implicit Surface (GPIS) introduced in Chapter 3, which uses GP to approximate a function $f(y) : \mathbb{R}^D \rightarrow \mathbb{R}^{D+1}$ that maps the position of a perturbed point y to its distance to the manifold d and its projection direction n . The main reason to use GP here is the fact that, in practice only very few perturbed points and their projection points are available. Also, the GPIS model allows an on-line update of the learned model when more data points (projection experience) are collected. The update procedure is similar as the GP filter presented in our previous work (Sommer et al., 2014) and we will not depict it here.

Given a set of N_p perturbed points $\mathcal{Y} = \{y^i \in \mathbb{R}^D\}_{i=1 \dots N_p}$ and their corresponding projection points on the manifold $\mathcal{Z} = \{z^i \in \mathbb{R}^D\}_{i=1 \dots N_p}$, which can be obtained during task execution or in simulation, the training dataset for learning the model $f(y)$ contains two parts. The first part is $\{y^i, d^i, n^i\}_{i=1 \dots N_p}$, where the position of the perturbed points y^i is the input, and the output includes the distance to the manifold $d^i = \|z^i - y^i\|$ and the projection direction $n^i = (y^i - z^i)/d^i$. The second part is $\{z^i, d^i, n^i\}_{i=1 \dots N_p}$ where points on the manifold z^i is the input and the output are all zeros, i.e., $d^i = 0$ and $n^i = 0_D$. The computation of the covariance matrix and the selection of the model parameters are the same as the procedure described in Section 3.3.1. With the learned model, given the position of a new perturbed point, we can quickly predict its distance to the manifold as well as its projection direction using Eq. 3.7. A toy example is shown in Fig. 5.5 where $N_p = 30$ perturbed points are used to learn the distance function.

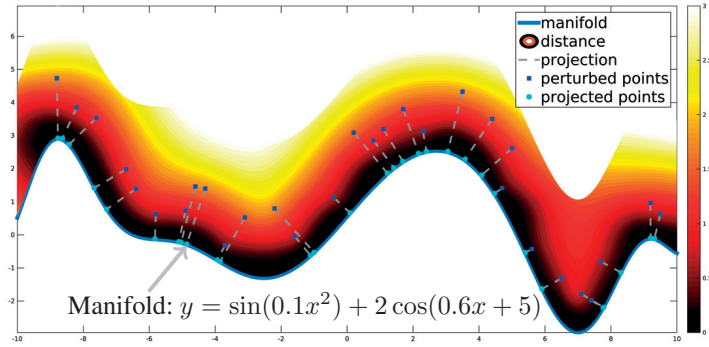


Figure 5.5: A toy example of learning the distance function to a curve. $N_p = 30$ perturbed points and their corresponding projected points (obtained using Algorithm 5.2) are used to learn the distance function. The color of each point above the curve represents the predicted distance from that point to the curve.

5.5 OBJECT-LEVEL IMPEDANCE CONTROL ON TASK MANIFOLD

As discussed in Section 5.2.2, given the planned path for COM, an object-level impedance controller is adopted to compliantly execute the task. However, it is still difficult to specify the proper impedance for a given task. This section presents a learning approach to select the proper impedance parameters, and to adapt to external perturba-

tions by exploiting the local manifold geometry and the human demonstrated experience.

5.5.1 OBJECT-LEVEL IMPEDANCE CONTROL

The detailed derivation of an object-level impedance controller is given in Appendix A. In general, the object-level impedance control law consists of two parts: the stable grasping impedance and the object manipulation impedance. How to select the impedance for stable grasping has been addressed in Chapter 4. This chapter will only address the object manipulation impedance control, which is given as

$$\mathbf{f}_{f,o} = D_d(\dot{\mathbf{x}}_r - \dot{\mathbf{x}}) + K_d(\mathbf{x}_r - \mathbf{x}) \quad (5.4)$$

where $\mathbf{f}_{f,o} \in \mathbb{R}^6$ is the summation of manipulating forces $\mathbf{f}_{f,oi}$ exerted on the object from each fingertip. $\mathbf{x}_r, \dot{\mathbf{x}}_r \in \mathbb{R}^6$ is the reference(desired) trajectory and $M_d, D_d, K_d \in \mathbb{R}^{6 \times 6}$ are the desired inertia, damping and stiffness matrix, respectively.

In order to implement this controller, we need to address at least the following issues: (a) measure the object position and orientation $\mathbf{x}, \dot{\mathbf{x}}$; (b) design the reference trajectory $\mathbf{x}_r, \dot{\mathbf{x}}_r$; (c) choose the impedance parameters K_d, D_d . While (a) is addressed already by estimating the object configuration using VF and robot forward kinematics, and the path planning approach in Section 5.4 can provide desired reference trajectory $\mathbf{x}_r, \dot{\mathbf{x}}_r$, the next section addresses the selection of manipulation impedance parameters K_d, D_d from human demonstration.

5.5.2 IMPEDANCE LEARNING ON TASK MANIFOLD

The human demonstration can be performed in the form of kinesthetic teaching whereby the human expert holds the robotic hand to move the object, or in a more natural form that the human demonstrator directly moves the object with his/her hands. Both approaches are allowed since the desired interaction is specified from the object's perspective, which naturally solves the correspondence problem in imitation learning and is considered as one of the key advantages for the object-level impedance control and learning.

During the demonstration, at each sample instant $i, i = 1 \dots N_s$, the motion of the object $\{\mathbf{x}(i), \dot{\mathbf{x}}(i)\}$ and the sum of manipulating forces $\mathbf{f}_{f,o}(i)$ applied on the object are recorded¹. Consider $t = 1 \dots N_t$ consecutive samples of data obtained over a short time window. Assuming the impedance parameters and reference trajectory remain constant over this time window, the relationship between the object motion and the force exerted on object is given by:

$$\mathbf{f}_{f,o}(i) = D_d(\dot{\mathbf{x}}_r - \dot{\mathbf{x}}(i)) + K_d(\mathbf{x}_r - \mathbf{x}(i)), i = 1 \dots N_t; \quad (5.5)$$

¹In practice, only the contact forces on each fingertip can be measured, which include the grasping forces and the manipulating forces. The aim of the grasping forces is to keep the object stable and by definition their sum is equal to the constant external force (e.g., the gravity of the grasped object), which is ignored in our case.

Within each time window, since we assume that the object's impedance parameters and the reference trajectory are not changing with time, they can be obtained by minimizing the following objective function.

$$\min_{D_d, K_d, \dot{\mathbf{x}}_r, \mathbf{x}_r} : \sum_{i=1}^{N_t} \|\mathbf{f}_{f,o}(i) - \{D_d(\dot{\mathbf{x}}_r - \dot{\mathbf{x}}(i)) + K_d(\mathbf{x}_r - \mathbf{x}(i))\}\|^2 \quad (5.6)$$

Note that the reference trajectory $\mathbf{x}_r, \dot{\mathbf{x}}_r$ is also unknown during the demonstration. In practice, the damping term is usually ignored, assuming that the desired velocity trajectory is the same as the measured one, and thus Eq. (5.7) can be simplified as

$$\min_{K_d, \mathbf{x}_r} : \sum_{i=1}^{N_t} \|\mathbf{f}_{f,o}(i) - K_d(\mathbf{x}_r - \mathbf{x}(i))\|^2 \quad (5.7)$$

Note that the assumption that the object's impedance parameters and reference trajectory are stationary for short periods of time is only valid for tasks of low velocities, where the dynamics is mainly dominated by the compliance and contact condition. For fast tasks, it will require some high speed (force and motion) sensors to collect enough data points to estimate the desired impedance parameters.

Eq. (5.7) is a general objective function to extract the desired stiffness for COM task, however, it can not be solved directly since the desired stiffness K_d and reference trajectory \mathbf{x}_r are both unknown. Fortunately, since the task manifold is already known in Section 5.3, some additional constraints and apriori knowledge should be taken into account. To give the mathematical formulations of these constraints, we need to first define the concept of directional stiffness (El-Khasawneh and Ferreira, 1999), which can be physically interpreted in a relationship between changes in the magnitude of external force and the displacement in the direction of applied force. Therefore, for a given stiffness matrix K , the directional stiffness in the direction of a unit vector $\mathbf{v} \in \mathbb{R}^D$ is formulated as

$$k^v = \frac{\|\mathbf{F}\|}{\|(K^{-1}\mathbf{F})^T \mathbf{v}\|} = \frac{1}{\|\mathbf{v}^T K^{-T} \mathbf{v}\|} \quad (5.8)$$

where $\mathbf{F} = f\mathbf{v} \in \mathbb{R}^D$ is the force with magnitude $f \in \mathbb{R}$ along the direction of \mathbf{v} .

The directional stiffness can be also formulated in terms of eigenvalues $\lambda_i \in \mathbb{R}^+$ and eigenvectors $\boldsymbol{\xi}_i \in \mathbb{R}^D$ of the stiffness matrix $K = \sum_{i=1}^{i=D} \lambda_i \boldsymbol{\xi}_i \boldsymbol{\xi}_i^T$ as

$$k^v = \frac{\|\mathbf{F}\|}{\|\sum_{i=1}^{i=D} \mathbf{v}^T \frac{\mathbf{F}^T \boldsymbol{\xi}_i}{\lambda_i} \boldsymbol{\xi}_i\|} = \frac{1}{\sum_{i=1}^{i=D} (\mathbf{v}^T \boldsymbol{\xi}_i)^2 / \lambda_i} \quad (5.9)$$

The Eq. (5.9) shows that to change the directional stiffness, either the eigenvectors $\boldsymbol{\xi}_i$ or the eigenvalues λ_i of the stiffness matrix should be changed accordingly. The eigenvalue decomposition of stiffness matrix gives a more intuitive way to express stiffness and facilitates the estimation of desired stiffness in Eq. (5.7). With the definition of directional stiffness, now we depict the additional constraints to solve Eq. (5.7).

- The stiffness matrix should be semi positive definite, denoted as $K_d \succeq 0$.
- The elements of the stiffness matrix should be less than some maximum value since the demonstrator can not demonstrate arbitrarily large object stiffness, $K_{i,j} \leq k_{lim}, \quad i = 1...6, j = 1...6$.
- The stiffness along the normal directions \mathbf{n}^i of at each point on the manifold should be small to accommodate uncertainty. Using Eq. (5.8 or 5.9) to compute the directional stiffness, this constraint is denoted as $k^{n^i} < k_{lim}^n, \quad i = 1...N_t$. There can be more than one normal directions for the point on the manifold, but they can all be obtained from the local basis vectors.
- For the reference trajectory, it should not be too far away from the actual measured object trajectory. Moreover, from the definition of COM, the discrepancy between the reference and actual trajectory should have further smaller value along the normal direction of the manifold, compared with the value along tangential directions of the manifold. This results in the two following constraints

$$\begin{aligned} |(\mathbf{x}_r - \mathbf{x}(i))^T \mathbf{n}^i| &\leq \Delta x_{lim}^n, \quad i = 1...N_t \\ \|\mathbf{x}_r - \mathbf{x}(i) - (\mathbf{x}_r - \mathbf{x}(i))^T \mathbf{n}^i \mathbf{n}^i\| &\leq \Delta x_{lim}^t, \quad i = 1...N_t \end{aligned} \quad (5.10)$$

Put together the objective function and the additional constraints, the impedance learning on manifold can be formulated as follows

$$\begin{aligned} \min_{K_d, \mathbf{x}_r} : & \sum_{i=1}^{N_t} \|\mathbf{f}_{f,o}(i) - K_d(\mathbf{x}_r - \mathbf{x}(i))\|^2 \\ \text{subject to: } & K_d \succeq 0 \\ & K_{i,j} \leq k_{lim}, \quad i = 1...6, j = 1...6 \\ & k^{n^i} < k_{lim}^n, \quad i = 1...N_t \\ & |(\mathbf{x}_r - \mathbf{x}(i))^T \mathbf{n}^i| \leq \Delta x_{lim}^n, \quad i = 1...N_t \\ & \|\mathbf{x}_r - \mathbf{x}(i) - (\mathbf{x}_r - \mathbf{x}(i))^T \mathbf{n}^i \mathbf{n}^i\| \leq \Delta x_{lim}^t, \quad i = 1...N_t \end{aligned} \quad (5.11)$$

Note that there are some hyper-parameters $k_{lim} \in \mathbb{R}^+, k_{lim}^n \in \mathbb{R}^+, \Delta x_{lim}^n \in \mathbb{R}^+, \Delta x_{lim}^t \in \mathbb{R}^+$ that still require manual specification for a given task. k_{lim} determines the largest stiffness that the human demonstrator can provide. k_{lim}^n and Δx_{lim}^n determine the maximum stiffness and tolerance of trajectory along the normal direction of the manifold. Δx_{lim}^t specify the maximum discrepancy between actual and desired trajectory along the trajectory. The obtained stiffness parameters within each time window is constant, however, during the entire task, they can vary along the reference path \mathbf{x}_r . It is important to note that the obtained impedance parameters and the reference trajectory \mathbf{x}_r will depend on time. To account for this, we define a variable $\phi \in [0, 1]$ to represent the status of the task, which is a function of the current configuration \mathbf{x} , i.e., $\phi = \Phi(\mathbf{x})$. In this work, $\Phi(\mathbf{x}) = \frac{d_{\mathcal{M}}(\mathbf{x}, \mathbf{x}_{goal})}{d_{\mathcal{M}}(\mathbf{x}_{init}, \mathbf{x}_{goal})}$, where $d_{\mathcal{M}}(\mathbf{x}, \mathbf{x}_{goal})$ denotes the distance from the current configuration \mathbf{x} to the goal configuration \mathbf{x}_{goal} , computed along

the manifold as in Section 5.2.1. Thus, the impedance parameters are expressed as a function of ϕ .

Remark: First, the learned task manifold is necessary since it gives rise to the computation of the normal direction \mathbf{n}^i for each point on the manifold. Second, we do not put any upper bound constraints for the stiffness along the manifold, since the upper bound largely depends on the task. For example, cleaning a table requires a much smaller stiffness along the manifold (which is the surface of the table), compared with that of polishing a workpiece. Third, the task status variable ϕ actually determines how the demonstrated impedance parameters should be generalized across the whole task manifold. It means that for a similar task status, the desired impedance parameters should be similar too. Rather than the distance to the goal configuration, other means such as similarity on perceived tactile or force sensing are also possible to adopt to represent the task status. This is beyond the scope of this work and hence not addressed here.

5.5.3 IMPEDANCE SELECTION FOR PROJECTION

Until now, the impedance learning approach presented in the previous section only addresses the selection of the proper impedance parameter when the object is moving on the manifold. If the object is perturbed and moved away from the manifold due to external perturbations, we may want the object to quickly move back to the manifold. In Section 5.4.3, we have proposed a fast projection and distance prediction method to monitor the configuration of the object with respect to the manifold. When the distance from current configuration \mathbf{x} to the manifold is predicted to be larger than a small threshold, $d > d_{\text{thresh}}$, an impedance selection block is triggered to set an impedance only along the projection direction \mathbf{n}^i to bring the object back to the closest point \mathbf{x}_* on the manifold. In this case, it is easier to set the desired stiffness in the object's frame $K_d^{\text{obj}} = \text{diag}\{\alpha e^{\|\mathbf{x}-\mathbf{x}_*\|} \mathbf{n}^i\}$, where $\alpha \in \mathbb{R}^+$ is a scaling factor.

With all the building blocks from the previous sections, the entire COM can be implemented in a closed-loop framework as shown in Fig. 5.6. The key connections between different blocks are shown with red color. The impedance learning block takes the task manifold (task constraints) into account explicitly, that leads to a task-consistent impedance specification. The distance prediction block can efficiently predict the distance to the manifold and find the projection point on the manifold. Therefore, it allows on-line path replanning and impedance selection, resulting in a closed-loop system.

5.6 SIMULATION AND EXPERIMENTAL RESULTS

In this section, we present the simulation and experimental results to demonstrate the usefulness of our task manifold learning approach for COM. The task manifold learning approach is validated with several simulation examples, while the whole framework for COM is verified on a real robot system for dexterous manipulation, which is consid-

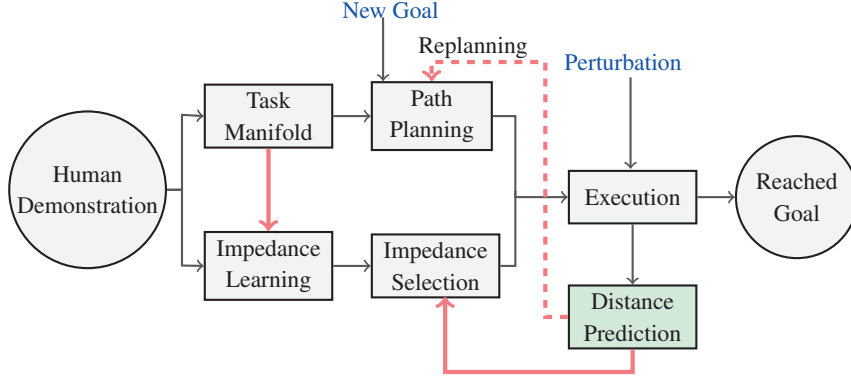


Figure 5.6: The pipeline for the whole COM, mainly consisting of three key blocks task manifold learning, impedance learning, and distance and projection prediction. The key connections between different blocks are highlighted with red color.

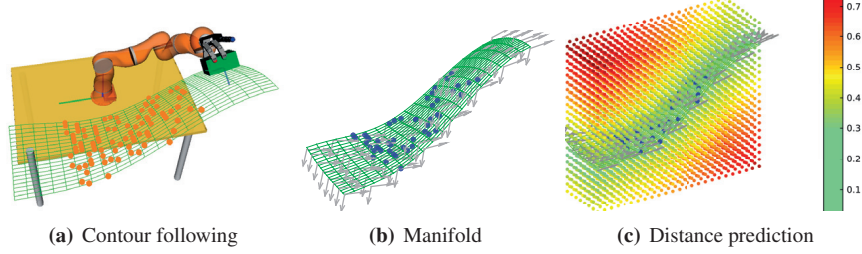


Figure 5.7: Contour following example. (a) A grasped object follows a contour. (b) Several data points are collected (blue dots) and used to learn the task manifold, represented by the basis vectors (gray arrows). (c) The predicted distance for each testing point to the manifold.

ered as one of the most challenging COM tasks (Okamura et al., 2000). An experiment of constrained manipulation is also conducted to demonstrate the effectiveness of our approach.

5.6.1 SIMULATION RESULTS FOR TASK MANIFOLD LEARNING

In this section, we present three simulation examples to show the effectiveness of manifold learning for COM task encoding and path planning. These examples are chosen because generally they are very difficult to plan in the robot joint space as discussed in Section 5.2.1, and compliant controller is required during the actual implementation of these examples.

EXAMPLE 1: CONTOUR FOLLOWING

The first example is to have a robot following a contour with a grasped object, as shown in Fig. 5.7(a). The contour is defined implicitly by

$$z = -h \sin(v)/v + 1.3 \quad (5.12)$$

where $v = \cos(ax - 1) + \sin(ay)$

where $[x, y, z] \in \mathbb{R}^3$ represents the Cartesian coordinates, with the contour parameters

$h = 1, a = 2.5, x \in [-0.8, -0.4], y \in [-0.6, 0.6]$. 50 training data are randomly sampled within region $x \in [-0.8, -0.4], y \in [-0.3, 0.3]$ and an uniformly distributed noise added on z , i.e., $z = -h \sin(v)/v + 1.3 + \mathcal{U}([-0.02, 0.02])$. For actual applications, these training data points will be collected through human demonstration.

To learn the manifold (see Section 5.3), the number of features, the number of nearest neighbors, and the parameter of regularization in Eq. (5.2) are selected as $p = 5, k = 5$ and $\lambda = 0.05$ respectively. The optimization process is restarted 100 times and the model of best performance in terms of the objective function in Eq. (5.2) is selected. With the learned model, the basis vectors for each point on the manifold can be computed from Eq. (5.3), as shown in Fig. 5.7(b). With randomly generated 20 off-manifold points, a function for distance and projection prediction is learned using GP as described in Section 5.4.3, the predicted distance is shown in Fig. 5.7(c).

To evaluate the performance of the learned manifold, we define an error metrics as

$$d_1 = \frac{1}{N_{\text{test}}} \sum_{i=1}^{i=N_{\text{test}}} \|g(\mathbf{x}^i)^T H(\mathbf{x}^i)\| \quad (5.13)$$

where $g(\mathbf{x}^i) \in \mathbb{R}^{D \times 1}$ represents the normal direction at \mathbf{x}^i , computed from Eq. (5.12), and $H(\mathbf{x}^i) \in \mathbb{R}^{D \times d}$ represented the normalized basis vectors obtained from Eq. (5.3). N_{test} is the number of testing data points. In this example, $N_{\text{test}} = 100$ test data points are generated on the manifold and the error is $d_1 = 0.2378 \pm 0.0306$, which corresponds to an orientation error of $(90 - \arccos(0.2378)) = 13.76$ deg. Consider the maximum change of orientation of the normal direction is 74.78 deg, the error percentage of the learned vector field is around $13.76/74.775 = 18.40\%$.

To evaluate the performance of the distance prediction, we define an error metrics as

$$d_2 = \frac{1}{N_{\text{test}}} \sum_{i=1}^{i=N_{\text{test}}} |d_{\text{gp}}(\mathbf{x}^i) - d_{\text{iter}}(\mathbf{x}^i)| \quad (5.14)$$

where $d_{\text{gp}}(\mathbf{x}^i) \in \mathbb{R}^+$ and $d_{\text{iter}}(\mathbf{x}^i) \in \mathbb{R}^+$ are the distances from a given point \mathbf{x}^i to the manifold, computed respectively using GP in Section 5.4.3 and using iterative gradient descent in Section 5.4.2. In this case, $N_{\text{test}} = 4725$ testing data points are generated by gridding the region $x \in [-0.8, -0.4], y \in [-0.6, 0.6], z \in [0, 1]$ with an interval 0.05 along each axis. The final error of the prediction distance is $0.042 \pm 0.021m$. Consider the maximum distance from all the test points to the manifold is $0.7169m$, thus the error percentage of the distance prediction is around $0.042/0.7169 = 5.86\%$.

In Fig. 5.8, we compare the path planning without and with the learned task manifold. With the learned task manifold, the robot can quickly bring the object back to the surface through fast projection and replan the path on the manifold. Without the learned manifold, the robot will directly move towards the goal (end point), disregarding the underlying task manifold (constraints), which is obviously not desirable for COM.

EXAMPLE 2: BIMANUAL MANIPULATION

In this example, an object grasped by two arms should follow a curve, as shown in

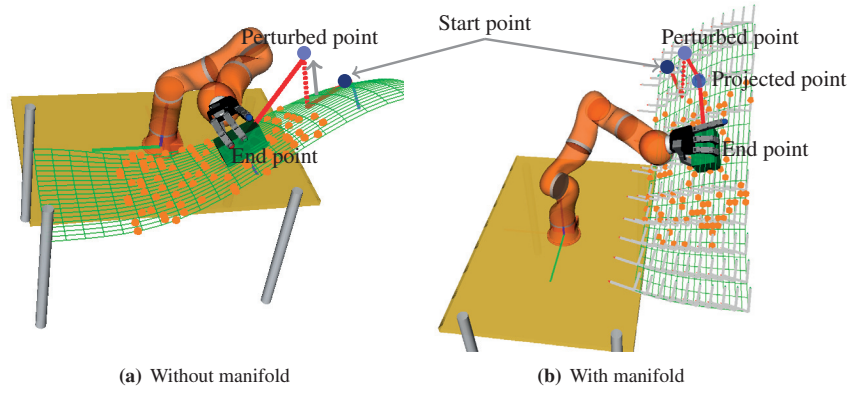


Figure 5.8: A toy example shows the on-line projection and the path replanning after perturbation with the learned task manifold. Note that the mapping from the points on the manifold to the feasible joints is highly non-linear due to the redundancy of the arm and the complexity of the manifold.

Fig. 5.9(a). For this case, since the underlying manifold is only 1D, the path replanning on the manifold is not really necessary. However, the learned task manifold is still useful for fast projection and distance prediction when the object is perturbed to a different position. The curve to follow is defined as

$$\begin{aligned} x &= -0.1 \sin(50(t + 0.15)) - 0.55 \\ y &= t \\ z &= 0.2 \sin(15(t + 0.15)) + 0.6 \end{aligned} \quad (5.15)$$

where $t \in [-0.15, 0.06]$. 100 data points are collected for training and the number of features, the number of nearest neighbors, and the parameter of regularization in Eq. (5.2) are selected as $p = 10$, $k = 3$ and $\lambda = 0.01$ respectively (see Section 5.3). The vector field of the learned manifold is shown in Fig. 5.9(b). The distance function is learned with 20 off manifold data points and the distance to the curve is shown as an iso-surface in Fig. 5.9(c). A comparison example is shown in in Fig. 5.10, which shows that the fast project enables a more meaningful task execution for this case.

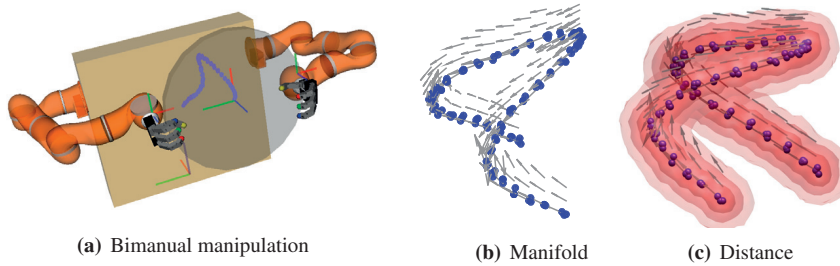


Figure 5.9: Bimanual manipulation example.

EXAMPLE 3: DEXTEROUS MANIPULATION

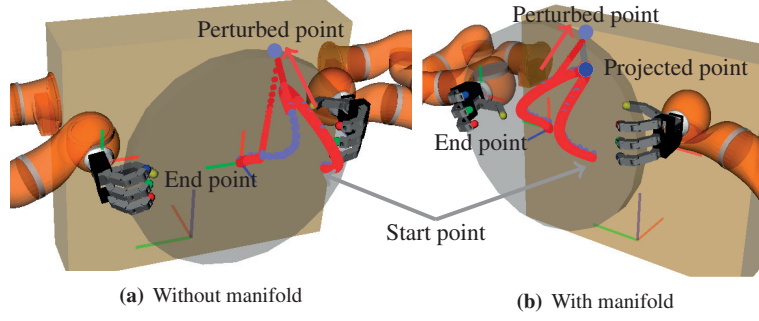


Figure 5.10: Bimanual manipulation example. (a) Without the task manifold, after the perturbation (red arrow), the object can not move back directly to the curve and the actual trajectory of the object will depend on the controller adopted. (b) With the learned manifold, the robot will move back to the curve first through fast projection and then move along the curve.

The last simulation example is dexterous manipulation, see Fig 5.11(a), where the object is grasped by a dexterous robotic hand and is tasked to follow a path. Here the path is defined as,

$$\begin{aligned} y &= r(1 + \sin(\theta)) \sin(\theta) \\ z &= r(1 + 0.5 \sin(\theta/2)^2) \cos(\theta) \end{aligned} \quad (5.16)$$

where $r = 1$ cm. We align the first point of this curve (i.e., $\theta = 0$) with the object's current position, as shown in Fig 5.11(a). The learned manifold and predicted distance to the manifold are shown as follows.

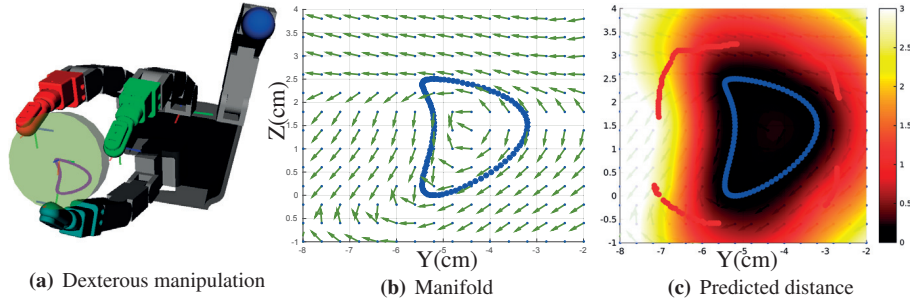


Figure 5.11: (a) Dexterous manipulation example. (b) The learned manifold. (c) The predicted distance to the manifold. The red points represents the perturbed points that are used to learn the distance function.

Note that in simulation, we assume the manifold to learn, (i.e., the curve and the surface) are known with an analytical representation. We use this known manifold to generate the training data and to evaluate the performance of the manifold learning. However, in actual application, we can only collect a set of demonstrated data points, from which the underlying manifold should be learned.

5.6.2 EXPERIMENTAL RESULTS FOR DEXTEROUS MANIPULATION

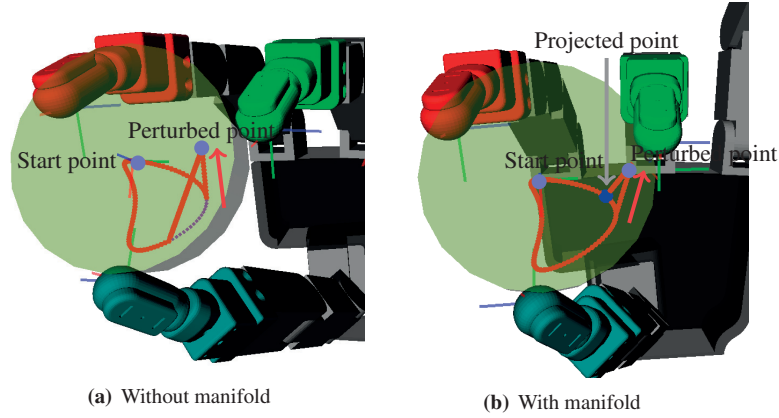


Figure 5.12: Dexterous manipulation example. (a) Without the task manifold, after the perturbation (red arrow), the object does not have an adaptation strategy and still tries to catch up with the predefined trajectory. (b) With the learned manifold, the robot will move back to projected point on the curve first and then move along the curve.

In the section, we present several experimental results of dexterous manipulation on a real robotic hand. We use a 4-fingered Allegro hand² to demonstrate the effectiveness of the proposed task manifold learning for COM tasks, including bulb insertion, linear path following and circular path following. Each of the four fingers of the Allegro hand has 4 independent torque-controlled joints with control rate of 333 Hz, see Fig. 5.13(a). In our experiments, only 3 fingers are used and each fingertip of these 3 fingers has been mounted with a biometric tactile sensor (BioTac)³, which has been calibrated to provide contact information such as contact position and contact force.

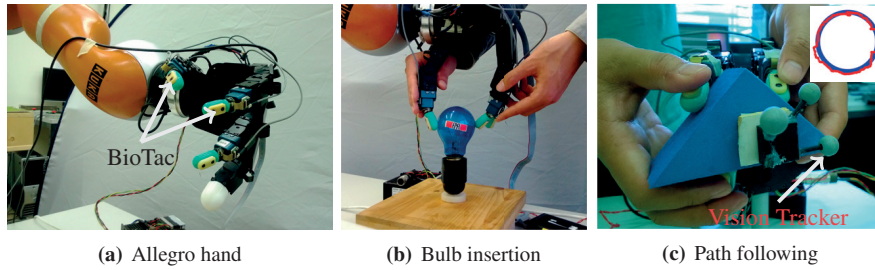


Figure 5.13: (a) The Allegro hand mounted with the SynTouch tactile sensors on the fingertips; (b) Human demonstration of the bulb insertion; (c) Human demonstration of the path following.

EXAMPLE 1: BULB INSERTION

The first example is bulb insertion, as shown in Fig. 5.13(b). The bulb is initially on the socket already. During the human demonstration, only two fingers are used as the impedance is learned in object's frame of reference and using two fingers is easier to

²<http://www.simlab.co.kr/Allegro-Hand.htm>

³<http://www.syntouchllc.com/>

demonstrate. The manipulating forces are measured using SynTouch mounted on the fingertips. The object real trajectory is tracked using a motion capture system from OptiTrack with a sampling rate of 240HZ. During the human demonstration, in order to track the object robustly, the experimenter must take care of not placing the fingertips on the vision markers. With the recorded data, we computed the rotation Euler angles, and the manifold is learned accordingly, which corresponds to the rotation around the Y axis, i.e., the axis direction of the socket, as shown in Fig. 5.14(a).

Recall the Eq. (5.11), with $\Delta x_{lim}^t = 60 \text{ deg}$, $k_{lim} = 100 \text{ N.mm/deg}$ ⁴. For this example, we only consider the stiffness on the manifold, i.e., the rotational stiffness around the axis, and we set the stiffness along the normal directions of the manifold to zero (i.e., rotational stiffness around X and Z axis). The optimization framework in Eq. (5.11) is solved using *fmincon* in MATLAB. The obtained reference trajectory and desired stiffness for bulb insertion are shown in Fig. 5.14(b) and Fig. 5.14(c), respectively.

If we compare the desired rotation angle with the actual rotation angle, we see that the difference varies during the whole task. This means that human demonstrator indeed regulates the difference between the actual and reference trajectories as well as the stiffness parameter. When looking at the desired object stiffness, Fig. 5.14(c), we see that the desired stiffness increases significantly during the last phase of the task. This is due to the fact that the resistance torque between the bulb and the socket increases significantly at the last phase. We repeated this demonstration 10 times, the obtained desired stiffness for each trial is shown in Fig. 5.14(d). With the learned reference trajectory and the desired rotational stiffness, the snapshot of the implementation on Allegro hand is shown in Fig. 5.15, and more implementation details has been reported in (Li et al., 2014b).

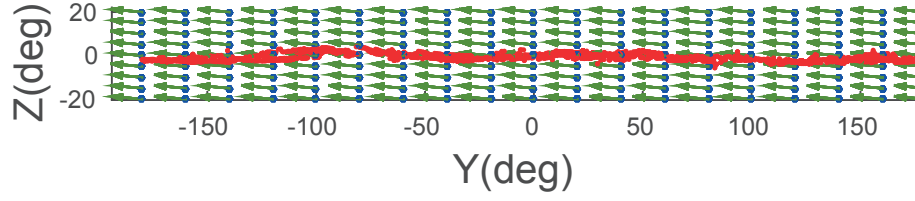
EXAMPLE 2: LINEAR MOTION

In the second example, a human experimenter demonstrates several linear trajectories of the grasped object with visual feedback, as shown in Fig. 5.13(c). With the recorded positions of the object, the underlying task manifold is first learned, which is simply a 1 dimensional straight line. Then with Eq. (5.11), the desired stiffness along the manifold as well as the reference trajectory are obtained. Finally, once a new goal position is specified, the path from current position to the goal is quickly planned and then executed with the learned manifold and desired stiffness (see the pipeline in Fig. 5.6).

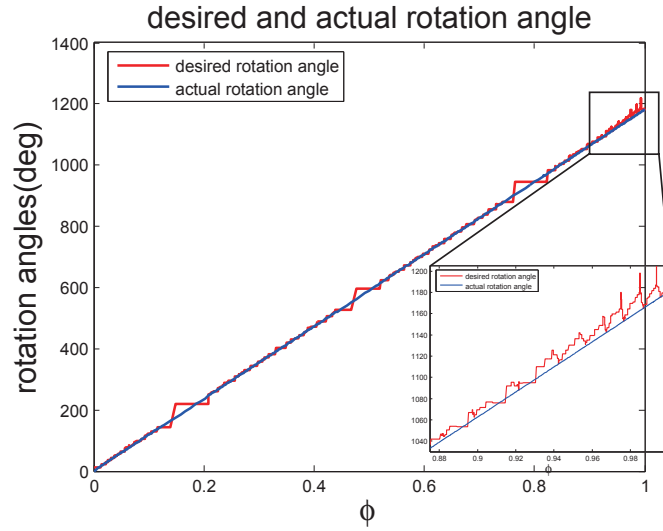
For the linear motion, three demonstrated paths are considered in the Y - Z plane (horizontal, vertical and diagonal), as shown in Fig. 5.16. The learned manifolds (vector fields), distance functions and the stiffness profiles along the manifold are shown as Fig. 5.17, Fig. 5.18 and Fig. 5.19.

To test the performance of the task manifold approach, a new goal position is given that is 5 cm away along the demonstrated direction. We compare the task performance in terms of the average distance to the task manifold, i.e., the distance to the straight line

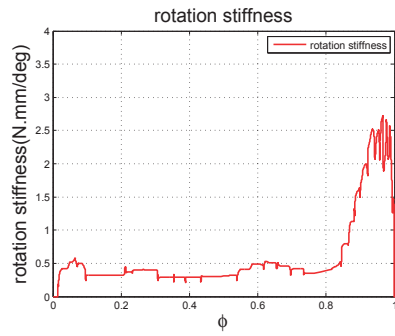
⁴ Δx_{lim}^t is chosen by considering the rotation limitation of the human hand and the Allegro hand.



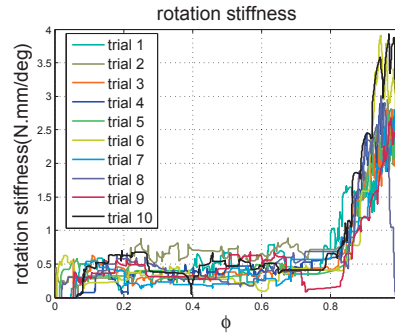
(a)



(b)



(c)



(d)

Figure 5.14: (a) The demonstrated rotation (Euler angles) around Y and Z axis and the learned manifold (arrows); (b) The learned reference trajectory for trial 5. ϕ is the variable that represents the status of completion of the task, which is chosen as the ratio between current rotational angle and maximal rotational angle; (c) The learned desired object stiffness for trial 5. The stiffness significantly increases during the last phase of bulb insertion. (d) The learned desired object stiffness for 10 different trials.

connecting the initial and the goal position. For both cases, the reference trajectories are obtained through linear interpolation with an interval of 0.5cm between current position and goal position. The hand is controlled at 100Hz (send joint torques and

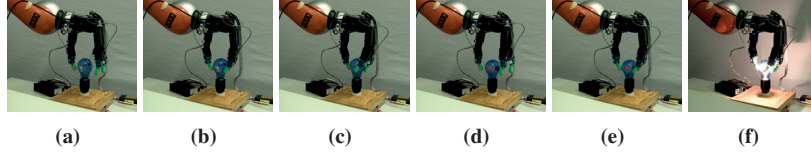


Figure 5.15: The snapshots for the implementation of bulb insertion.

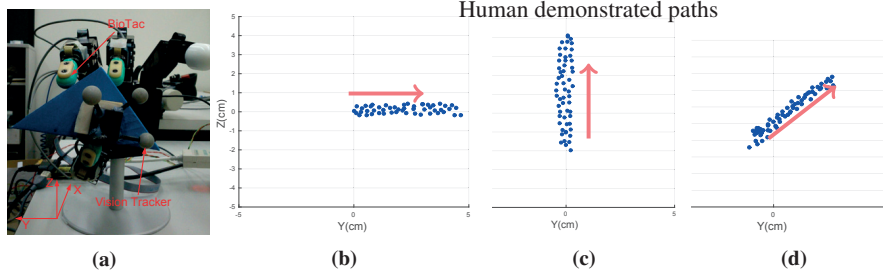


Figure 5.16: (a) The setup for path following. (b) Horizontal path; (c) Vertical path; (d) Diagonal path. The red arrows represent the moving direction during each demonstration.

update current object position), and the intermediate desired position is updated at 10Hz. For the case with manifold, if the distance to the manifold is predicted as bigger than $d_{\text{thresh}} = 0.2\text{cm}$, the object will be first moved towards the manifold and the reference trajectory is replanned.

As for the stiffness setting, for the case without the manifold, a constant stiffness is given $k_t = k_n = 2 \text{ N/cm}$, where k_t and k_n represent the stiffness along the manifold and perpendicular to the manifold. The stiffness value for the case with the learned manifold is determined according to the demonstration, as shown in Fig. 5.17(c), 5.18(c) and 5.19(c). It is obvious that the desired stiffness for this case are varying during the task, with a different stiffness values along the manifold and perpendicular to the manifold. The stiffness for projection (moving back to the line) is set to 2 N/cm.

The task performances for both cases are shown as Fig. 5.20(a) and 5.20(b). Each task is repeated five times and from the comparison of the average distance to the task manifold, see Fig. 5.20(c), it is indeed that our approach greatly improve the performance for linear path following – the distance error is decreased by 59.27% on average.

EXAMPLE 3: CIRCULAR MOTION

In this section, we present our results for circular path following. A human demonstrator first uses kinaesthetic teaching to demonstrate the task with visual feedback, as shown in Fig. 5.13(c).

To learn the task manifold with the demonstrated data, a grid search is used to optimize the hyper-parameters. The number of neighbours is $k = 5$ in Eq. (5.2) and the number of features is $p = 5$. The other parameters are selected as the best (in the sense of Eq. (5.2)) among 50 optimizations with different initial points. The learned

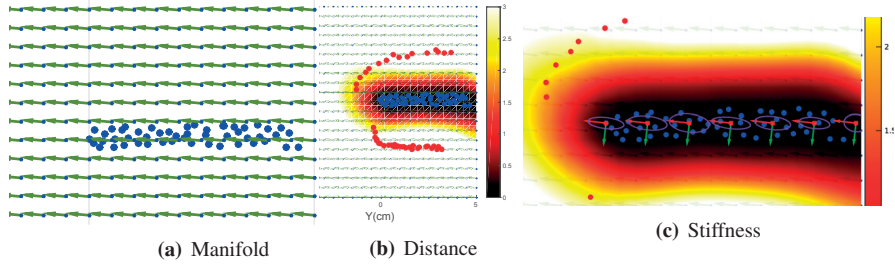


Figure 5.17: (a) The learned vector field for horizontal path. (b) The learned distance function, and the red points are the perturbed points used for training. (c) The stiffness profile (ellipse) along the demonstrated path (moving 5cm from left to right).

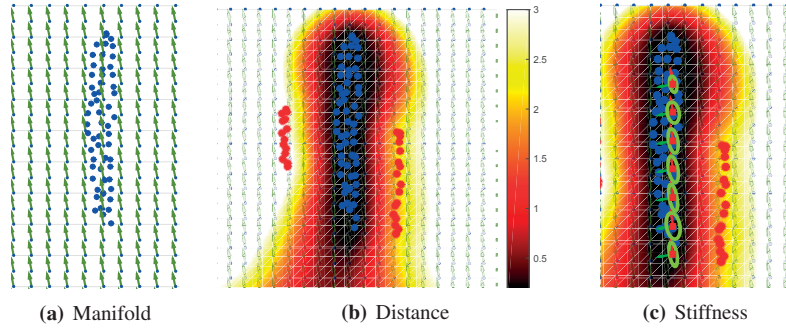


Figure 5.18: (a) The learned vector field for vertical path. (b) The learned distance function, and the red points are the perturbed points used for training. (c) The stiffness profile (ellipse) along the demonstrated path (moving 5cm from bottom to top).

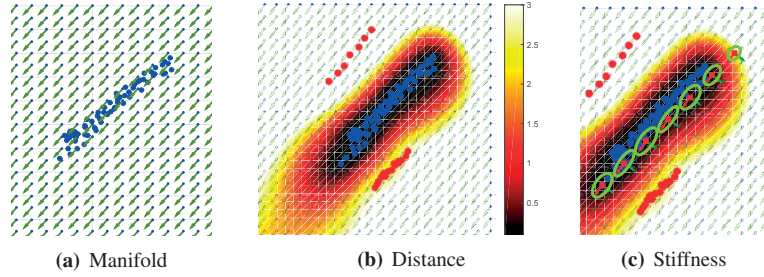


Figure 5.19: (a) The learned vector field for diagonal path. (b) The learned distance function, and the red points are the perturbed points used for training. (c) The stiffness profile (ellipse) along the demonstrated path (moving 5cm along the diagonal direction).

manifold for the circular motion task is shown in Fig. 5.21(a) and the desired stiffness along the circle is shown as Fig. 5.21(b).

With the learned task manifold, now we can implement the task that the robotic hand is to follow a circular path with radius of 2cm. The current position is set as the starting point of the circle, which is the highest point on the circle, shown as solid black circle in Fig. 5.22(a) and 5.22(b). The whole circle is divided into 100 segments and

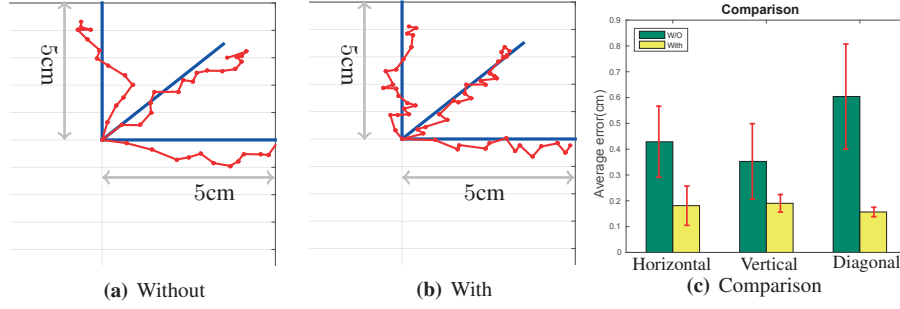


Figure 5.20: The object is commanded to move 5cm along each demonstrated direction with the learned manifold (b) and without the learned manifold (a). (c) The comparison of the average distance to the task manifold.

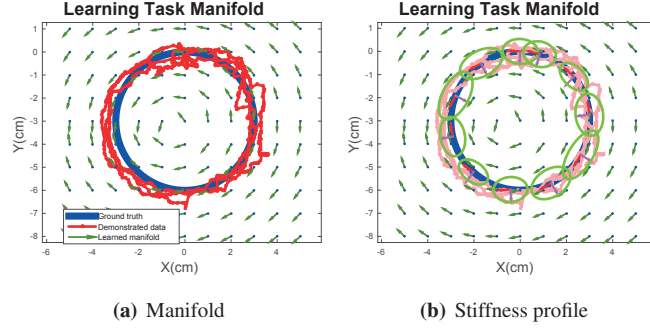


Figure 5.21: (a) A human demonstrator teaches an underlying task: follow a circle with radius of 3cm (anticlockwise direction). The trajectory of the object is recorded and the demonstration is repeated 3 times. The task manifold (vector field) is learned from the recorded data; (b) The desired stiffness profile (ellipse) along the circle.

the desired position for the object is updated in 10 Hz. For the case with the learned manifold, the parameters are set as the desired stiffness profile in Fig. 5.21(b). For the case without the learned manifold, constant stiffness values are given as follows: $K_n = K_t = 2 \text{ N/cm}$. The other parameters in the controller are the same for these two cases. As shown in Fig. 5.22(b), with the task manifold, the distance error to the circle is decreased by 71.43% on average, which is a significant performance improvement for dexterous manipulation, compared with the state-of-the-art results for dexterous manipulation (Wimböck et al., 2012; Kuo et al., 2015).

5.6.3 EXPERIMENTAL RESULTS FOR CONSTRAINED MANIPULATION

In this section, we present our experimental results of a fan blade cleaning task. The robot setup is shown in Fig. 5.23(a). This is one of the typical constrained manipulation examples, where proper impedance parameters (or forces) should be set to avoid excessive contact forces during the operation to damage the workpiece. Moreover, the

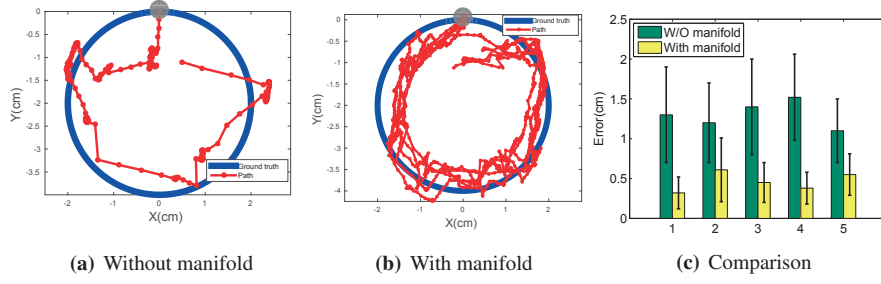


Figure 5.22: (a) The path of the object during the task without the learned manifold; (b) The path of the object during the task with the learned manifold (repeated five times). The solid black dot is the starting point. (c) The comparison of the average distance to the circle.

strength of the fan blade is varying across the surface, which may require different level of impedances at different locations on the surface. In the following subsections, we will first present our experimental setup for the human demonstration of the task, where the force and motion information are collected during the demonstration. The task manifold and the desired impedance parameters are learned with the collected data. The robot experiments are conducted to evaluate the task performance with the learned impedance parameters.

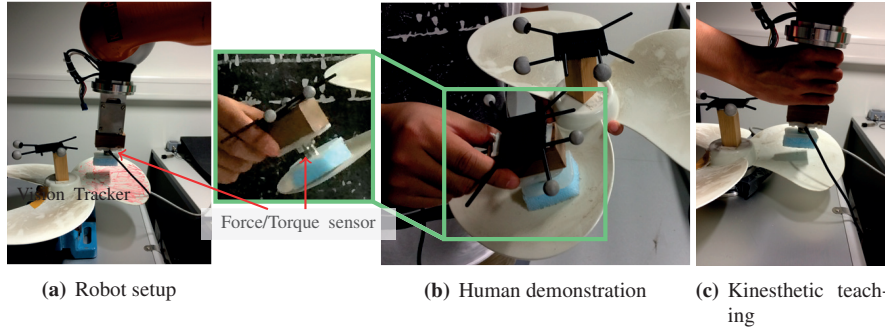


Figure 5.23: (a) The robot setup for the fan blade cleaning task; (b) The experimental setup for the human demonstration. (c) The experimental setup for the kinesthetic teaching.

HUMAN DEMONSTRATION AND DATA COLLECTION

We tried two possible ways to demonstrate the fan blade cleaning task to the robot, namely human demonstration and kinesthetic teaching, as shown in Fig. 5.23(b) and Fig. 5.23(c) respectively. Both approaches have their own pros and cons for the data collection and the robot implementation, and here we will only present the results for the human demonstration.

During the human demonstration, the trajectories of the fan and the cleaning tool are tracked using a vision tracking system (OptiTrack) with a sampling rate at 240HZ. The forces required by the task are measured using a force/torque sensor (ATI Nano 17) with a sampling rate at 100HZ. The cleaning tool is performing some kind of

“random” motion along the anticlockwise direction⁵, and part of the demonstrated path (red line segments) of the tool is shown in Fig. 5.24(a). The force trajectory during the demonstration is shown in Fig. 5.24(b). It is clearly that the force in the Z direction varies within a range around -2N and -1N . While the forces in the X and Y directions also change, but they show some kind of coordination. In the next section, we will use the demonstrated data to learn the task manifold and the desired impedance parameters.

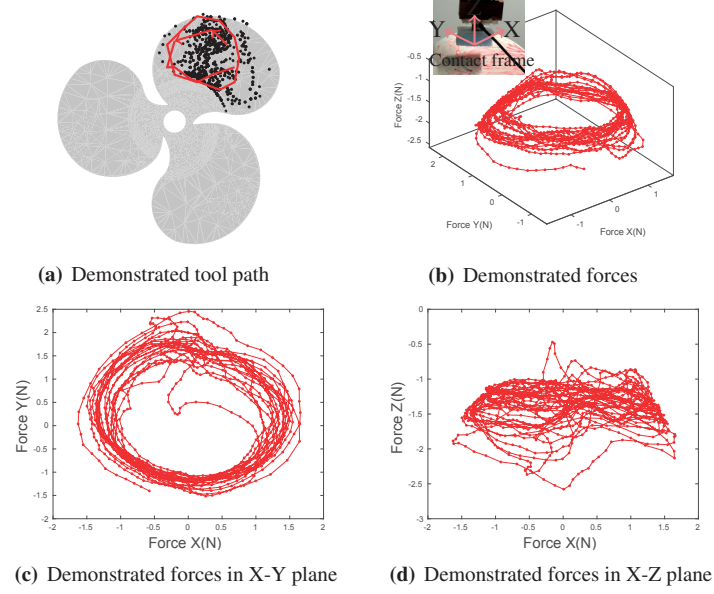


Figure 5.24: (a) Part of the demonstrated paths (red line segments) of the cleaning tool and the demonstrated positions of the cleaning tool (black dots); (b) The demonstrated forces in the contact frame. The orientation of the contact frame is the same as that of the force/torque sensor’s frame, and the origin of the contact frame is at the center of the tool tip; (c) Demonstrated forces in X-Y plane; (d) Demonstrated forces in X-Z plane.

TASK MANIFOLD LEARNING AND IMPEDANCE LEARNING

To learn the task manifold with the demonstrated positions of the tool, a grid search is used to optimize the hyper-parameters. The number of neighbours is $k = 8$ in Eq. (5.2) and the number of features is $p = 5$. The other parameters are selected as the best (in the sense of Eq. (5.2)) among 100 optimizations with different initial points. The learned manifold for the fan blade cleaning task is shown in Fig. 5.25(a), where the blue points are the testing points to show the local tangent vectors.

With the learned task manifold, now we can learn the desired impedance parameters for the demonstrated task using both the force and the motion information. Recall the optimization framework in Eq. (5.11), the size of time window is set to $N_t = 10$, the tolerance of the trajectories along the normal direction and tangential direction are set to $\Delta x_{lim}^n = 0.01\text{m}$ and $\Delta x_{lim}^t = 0.02\text{m}$ respectively, and the maximum normal

⁵This moving direction is not mandatory for our approach, however it will be helpful to visualize the force pattern during demonstration

stiffness is set to $k_{lim}^n = 50\text{N/m}$. With these settings, the impedance parameters and the reference trajectory for the demonstrated trajectory can be obtained. Thus, given a new point on the fan blade, its desired impedance parameters are chosen as the that of its nearest neighborhood (in terms of norm 2) in the demonstrated locations. The reference trajectory on the manifold is planned as mentioned in Section 5.4, and the reference trajectory perpendicular to the manifold (namely, the penetration distance at each point) is chosen as that of its nearest neighborhood (in terms of norm 2) in the obtained reference trajectory from demonstration. In Fig. 5.25(b), the desired impedance (stiffness) in the local frame, along the task manifold (X, Y directions) and perpendicular to the task manifold, is shown at each testing point on the surface of the fan blade.

It can be noticed that the desired impedance in the Z direction is generally much smaller than the impedances along the X and Y directions. This makes sense since it is desirable that the tool is more compliant along the normal direction of surface of the fan blade. The impedances along the X and Y directions are varying across the surface of the fan blade. Moreover, it seems that on the area with larger strength, larger impedance parameters are also expected.

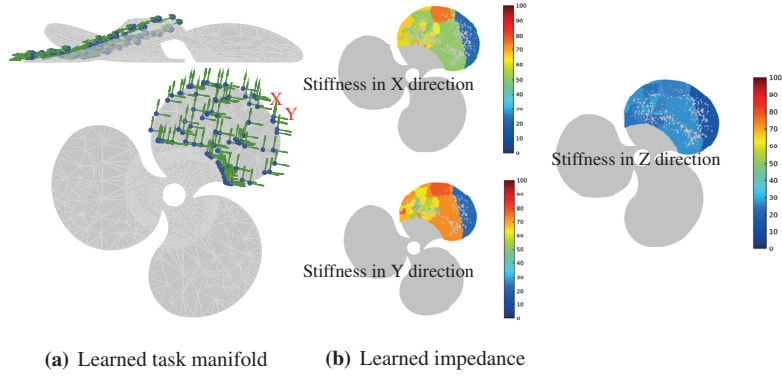


Figure 5.25: (a) The learned task manifold for the fan blade cleaning task. (b) The desired impedance (stiffness: N/m) in the local frame, along the task manifold (X, Y directions) and perpendicular to the task manifold (Z direction), is shown at each testing point on the surface of the fan blade.

ROBOT IMPLEMENTATION

With the learned task manifold and impedance parameters, the fan blade cleaning task is implemented using a 7 DOF KUKA LWR robot arm, as shown in Fig. 5.23(a). The cleaning path for the tool is first planned as anticlockwise circular motion with different radius and centers, and then this initial path is projected onto the manifold to obtain the final path for cleaning, as shown in Fig. 5.26(a). During the execution, at each location the stiffness parameters are set to the mean of the learned stiffness parameters of the nearest 5 demonstrated locations. To implement on the LWR, we actually scale the learned stiffness parameters 6 times to overcome the friction in the joints and damping terms in the robot's low-level controller. The executed tool path is shown in

Fig. 5.26(b) and the force during execution is shown in Fig. 5.26(c). Some snapshots of the execution are shown in Fig. 5.27.

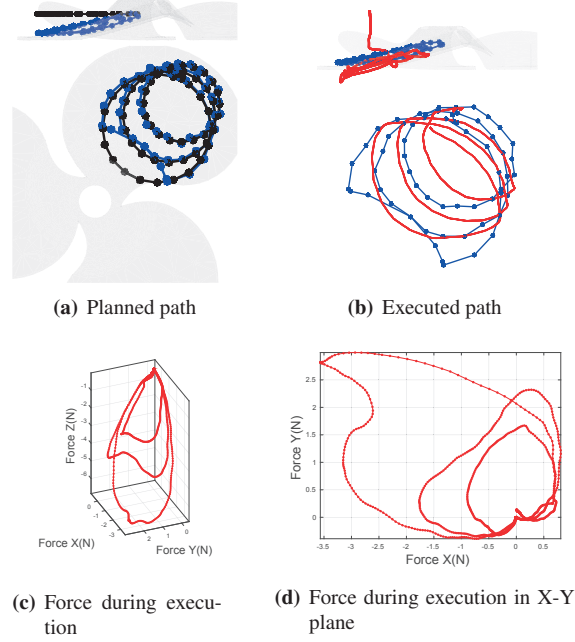


Figure 5.26: (a) The initial planned path (black) and final planned path (blue); (b) The executed tool path (red); (c) The recorded force trajectory during execution; (d) The recorded force trajectory during execution in X-Y plane.

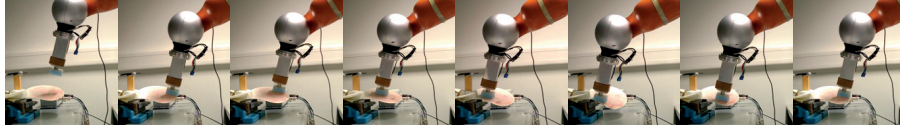


Figure 5.27: The robot implementation of the fan blade cleaning task.

To compare the task performance with other task implementations using constant stiffness parameters, we also implement the task with a set of low impedance parameters $K_x = K_y = 200\text{N/m}$, $K_z = 50\text{N/m}$, and a set of high impedance parameters $K_x = K_y = 1200\text{N/m}$, $K_z = 200\text{N/m}$. With low impedance, it is obvious that smaller forces will be generated during task execution. However, the shortcoming is that the cleaning tool may move very slowly and can detach from the surface of the fan blade during the task execution (due to the low impedance in normal direction), as shown in Fig. 5.28. On the other hand, with high impedance, the main drawback is that the fan blade can be deformed significantly in the area with low strength (near the edge of the lower part), and moreover when the tool collides with center support of the fan (see the highlighted circle in Fig. 5.29), large collision forces are generated that can be unfavorable for both the fan blade and the robot, see the force trajectory in Fig 5.29.

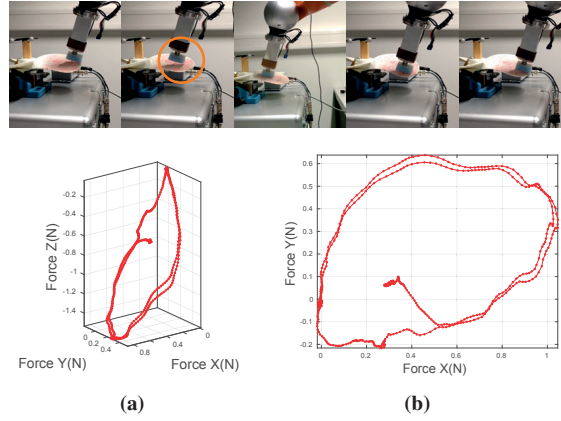


Figure 5.28: The robot implementation of with low impedance and the force trajectory during execution. The tangential forces (forces in X and Y directions) are too small, see plots in (b), compared with the demonstrated forces shown in Fig. 5.24(c).

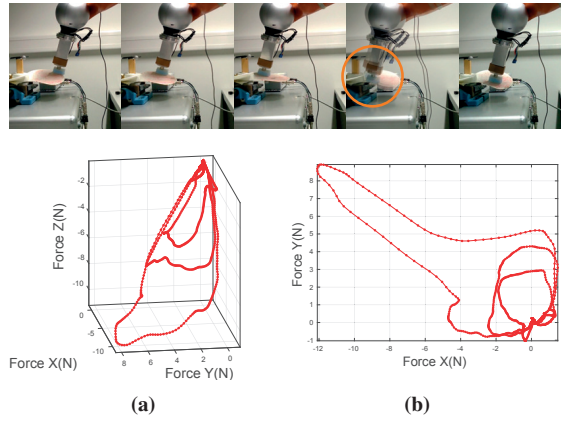


Figure 5.29: The robot implementation of with high impedance and the force trajectory during execution. It can be noticed that excessive forces are generated when the tool collides with the center support of the fan.

5.7 DISCUSSION AND SUMMARY

In this paper, we address a particular set of tasks – COM, where manipulation planning in robot joint configuration space can be quite difficult, and compliant control strategy is required to deal with the inevitable uncertainties from imprecise contact dynamics. We encode the task as a manifold which can be learned from several rounds of human demonstrations. Given the learned manifold, feasible paths can be efficiently generated and fast replanning can be performed on-line in case of perturbation. Moreover, taking advantage of the local geometry of the learned task manifold, we proposed an impedance learning framework that can learn the desired (varying) impedance parameters for the whole task. The effectiveness of our approach is verified with a number of simulation results and experiments on a real robotic hand for dexterous manipulation.

In this work, there are still several issues and limitations that has not been totally settled down. (1) To be able to represent the task as a manifold, the underlying task should be smooth and should not have “corners”. For example, the learned vector field

has a poor prediction of the tangent space near the corner, as shown in Fig. 5.11(b).

(2) For the path planning on the learned manifold (see Section 5.4), at this moment, the points on the initial path are merely projected back onto the manifold to obtain the final path on the manifold. However, we didn't incorporate any other criteria to further optimize the path on the manifold.

(3) While one of the key advantages of this work is to plan and control the task in object's frame, we didn't address the "redundancy problem" to distribute the task desired stiffness between arm and hand. For the case that the arm is compliant and the hand is stiff, the COM is the same as a typical compliant arm manipulation, as the experiment of the fan blade cleaning task in this work. For the other case that the arm is stiff and the hand is compliant, the COM is equal to the dexterous manipulation problem that we have demonstrated several experiments on a real robotic hand. It will be advantageous to combine the impedance of the arm and the hand in order to accomplish some more challenging tasks, which may require the change of impedance ranging from the scale that is suitable for the hand to the scale that the arm can offer.

(4) As with the previous chapter, this work only focused on the stiffness of the manipulated object and did not consider modulation of damping (or viscosity). One of the main advantages to learn the task manifold is that one can embed a model of stiffness variation in the task space and use this to directly control for the contact force and position of the object with respect to the task manifold. In addition, one could also envision to extend this approach to embed a model of damping variation, as it relates to the generated velocity of the object (in the in-hand manipulation example).

CONCLUSION AND SUMMARY

In this chapter, we summarize the main contribution of this thesis and discuss the limitations and potential working directions for future work.

6.1 MAIN CONTRIBUTIONS

Throughout this thesis, we have presented an approach that endeavours to address a fundamental problem in robotic grasping: *How to reason about and react to the perceptual and dynamic uncertainties in robotic grasping?*

In relation to the goal of this thesis stated in Fig. 1.2, the main contribution of this dissertation are four-fold:

First, we offer a learning-based approach to tackle with object's shape uncertainty in the pre-grasp stage. Our method parameterize the uncertainty of the object's shape explicitly using a Gaussian Process implicit surface, which is then incorporated as a constraint into a contact-level grasp planning algorithm. Furthermore, we propose a probabilistic hand inverse kinematics model to query on-line the feasible hand configurations, including hand poses and finger joints, that can realize the planned grasping points.

Second, this thesis proposes a dynamic grasp adaptation framework to deal with dynamic uncertainties in the post-grasp stage, including the unexpected perturbations acting on the object and the changes in object's weight. Our method takes advantage of grasp learning from human-guided grasp experience, which models the grasp stability as a probabilistic model that allows on-line estimation of the stability of the current grasp. Thanks to this on-line grasp stability prediction, we further devise a grasp adaptation strategy to maintain the stability of the grasp, by regulating the grasping forces at the fingertips or relocating the position of one fingertip through local exploration or finger gaiting. To the best of our knowledge, this is the first work that allows a robotic grasp to react dynamically to unexpected perturbation through on-line and in-hand grasp adaptation. From our perspective, this is the core contribution of this thesis that takes significant leap forward to move robotic grasping from static to dynamic.

Third, we provide a way of encoding the constrained object manipulation task as a manifold to deal with uncertainties from imprecise contact dynamics. Our method can learn the manifold in the form of a vector field from the imperfect human demonstrations. The learned manifold encodes the underlying task constraints and thus can be used to devise a task-consistent adaptation strategy that brings any off-manifold

deviation onto the manifold through fast projection and on-line path replanning. Furthermore, by exploiting the structure of the manifold, the proper impedance parameters for an impedance controller to execute the task can be obtained from the human demonstration. Our experimental results on dexterous manipulation (a typical example of COM) shows that the task manifold approach improves the task performance significantly, compared with the state-of-the-art reported results.

Fourth, although not explicitly emphasized in each chapter, this thesis also paves a cornerstone to bridge the gap between grasp (or task) planning and control by leveraging the competence of learning-based approach. As a key feature of robotic grasping or constrained object manipulation is to accommodate *interaction* between object and the environments, it obviously requires an integration between planning and control to allow flexible adaptation. Our approach benefits from the use of learning-based approach, especially data-driven approach, for both the encapsulation of the uncertainties during the planning stage and the derivation of the robust control laws during the execution stage.

6.2 LIMITATIONS AND FUTURE WORK

Although in each chapter, we have already described the limitations of our approach, we summarize them here and discuss possible working directions in relation to these limitations.

GRASP PLANNING UNDER UNCERTAINTY

In Chapter 3, we only deal with the uncertainty of the object’s shape. However, in the pre-grasp stage (including grasp planning and finger closing), there are many other types of uncertainties that affects the stability of the executed grasp, such as the uncertainties from object’s position and pose. It is obviously a very challenging task to plan an “optimal” grasp under all these uncertainties. One possible approach is to formulate the grasp planning as an optimization framework, as shown in Fig. 3.1, where various uncertainties are incorporated as constraints. However, this approach relies on analytical representations of these uncertainties, like the parameterization of the uncertainty of the object’s shape in our work.

PROBABILISTIC HAND INVERSE KINEMATICS

In Chapter 3, to learn the hand inverse kinematics model, we used a rejection sampling in the simulation to collect the training data. The learned model can predict the feasible hand configuration on-line that can render the given grasping points. This is achieved with a trade off between prediction speed and the accuracy of the prediction. To improve the model performance, it is possible to make use of the human-guided demonstration to collect context-dependent training data and to learn a context-specific inverse kinematics (Grochow et al., 2004; Hang et al., 2016). For example, the robot working in the kitchen and the robot working in a factory may require quite different

groups of grasps, and therefore they may need a specialized hand inverse kinematics for each case.

DYNAMIC GRASP ADAPTATION

Although we have demonstrated with extensive experiments that the dynamic grasp adaptation approach can greatly improve the robustness of robotic grasp, there are still two issues that need more attention. First, during the finger relocation, the hand quickly detaches one fingertip from the object and then contacts with the object again in the desired new position. The stability of the whole system has not been studied during this switching process. This is a very challenging task since it includes the continuous grasping dynamics and the discrete contact dynamics, and it might be an interesting direction to investigate the stability of this hybrid system in future work.

Second, we have noticed that the performance of our dynamic grasp adaptation is limited by the hardware of the hand design. The human fingers use the passive stiffness mechanism (tendons) to store energy temporarily and significantly lower the energy consumption, and thus can accommodate a very large part of perturbations during grasping. Several attempts have already been carried out that design compliant robotic hands to leverage passive mechanical adaptation to accommodate the uncertainties (Deimel and Brock, 2015; Odhner et al., 2014; Ciocarlie et al., 2014). It will be beneficial to combine the strengths of the passive mechanical adaptation and the active dynamic grasp adaptation strategy, to further improve the adaptability and robustness of robotic grasping.

GRASP ADAPTATION BEYOND STABILITY

The objective of the dynamic grasp adaptation in this thesis is to maintain the grasp stability, however, there are many other reasons that may require a robot to adapt its grasp. For example, during a hand-over task, we may want the robot to naturally release the object (rather than stabilize it) when humans attempt to hold the object. It is also important to notice that the grasp adaptation framework can be extended to a more broader context. For example, when a humanoid robot and a person are moving a table together (where the human arms and the robot arms can be seen as “big fingers”), rather than stabilize the table, these four “big fingers” are expected to collaboratively move the table. If there is a sudden change (for instance, human releases one hand to answer a cellphone), definitely the two robot arms require to adjust their efforts on the hand so as not to put too much weight on the human side.

PATH PLANNING ON VECTOR FIELD

In Chapter 5, we have encoded the COM task a manifold in the form of a vector field. The feasible paths can be quickly planed on the manifold, however, we didn’t use any further optimization criteria to optimize the path on the manifold. For some cases, it may be necessary to optimize the path according to the intrinsic metric defined on the manifold. Furthermore, as discussed in (Mussa-Ivaldi and Giszter, 1992), it is limited

to merely use one vector field to encode the task. This is also noticed in Fig. 5.11(b) that only one vector field can lead to a poor prediction near the corner where the gradient varies significantly. One way to address this is to partition the task space and to have multiple vector fields to encode the whole task, similar as the idea in (Shukla and Billard, 2012).

6.3 FINAL WORDS

During the last six years (from the beginning of my master study in 2009), I have put a large amount of efforts to work on robotic grasping and manipulation. Some of the most important lessons that I have learned about grasping are:

- *Explicitly reason about uncertainty:* The ability to reason about uncertainty is fundamental to make use of robotic grasping from the laboratory to the human-centric environments. This problem is extremely challenging mainly because of the various sources of uncertainties that can be encountered in real-world scenarios. I believe studies along this direction can truly advance robotic grasping to our vision of a robotic hand with a similar capability of human hands.
- *Exploit the dynamics:* Grasping is essentially a dynamic process rather than a kinematic or static state. Our study on dynamic grasp adaptation has demonstrated that extraordinary robustness and agility can be achieved by exploiting the dynamics of grasping. I believe some deep connections are possible – enabling intelligent reaction to the uncertainty by exploiting the dynamics of grasping.
- *Leverage the power of learning:* As demonstrated in this thesis and many other related studies, learning can significantly contribute to many aspects of grasping, ranging from planning to control. From my understanding, one of the most promising approaches to enable explicitly reason about and react to uncertainty, is to leverage the power of various learning techniques that allow us to generalize knowledge from humans or from efficient optimizations.
- *Embrace the contact with the world:* Grasping essentially requires contacts. Most of the works on grasping are, to some extent, studying the planning and control of the contacts. However, besides the contacts between the hand and the grasped object, there are many other contacts that grasping can greatly take advantage of, for example, the contact between fingers and the support surface such as a table top (Eppner et al., 2015; Roy et al., 2013), or the contact between the object and the environment (Dafle et al., 2014; Chavan-Dafle and Rodriguez, 2015). Some of these contacts can not be ignored or avoided, and moreover grasping can definitely benefit from these contacts if the robots can master them, to reduce the uncertainty or to shape the dynamics.
- *Optimize the hand embodiment:* Grasping relies on the embodiment of the hand design. A typical procedure in grasping is that, we first have a robotic hand

and then we devise different planning and control algorithm for this given hand. Therefore, it is obvious that the final grasping performance will be limited by the hand adopted. In other words, how can we formally choose or design a better robotic hand to improve the task performance? For example, for our dynamic grasp adaptation, it may be possible to design a hand to improve the adaptability that enables a larger area for the finger to relocate during finger gaiting. I believe this is also one of the areas where grasping can benefit from learning, especially learning from human demonstration.

Appendices

OBJECT-LEVEL IMPEDANCE CONTROL

In this part, we give an introduction of the object-level impedance controller (Tahara et al., 2012; Li et al., 2014b). An object-level impedance control can be separated into two parts: object manipulation impedance and stable grasping impedance, which are detailed in the following sections.

OBJECT MANIPULATION IMPEDANCE

Given a hand-object system, the dynamics of the object, as shown in Fig. A.1, is governed by the equation:

$$\mathbf{f}_{f,o} + \mathbf{f}_{ext} = M_0 \ddot{\mathbf{x}} \quad (\text{A.1})$$

where $\mathbf{f}_{f,o} \in \mathbb{R}^6$ is the summation of manipulating forces $\mathbf{f}_{f,oi}$ exerted on the object from each fingertip, \mathbf{f}_{ext} is the external perturbation force. All the forces are expressed in the inertial frame. $M_0 \in \mathbb{R}^{6 \times 6}$ is the actual inertia matrix of the object and $\mathbf{x} \in \mathbb{R}^6$ is a vector represented the position and orientation of the object. To control the motion of

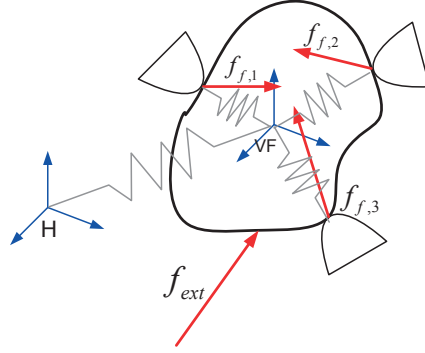


Figure A.1: An object is grasped by 3 fingers. The object impedance and grasp impedance are visualized as linear springs. The $\mathbf{f}_{f,i}, i = 1, 2, 3$ are the contact forces on each fingertips. \mathbf{f}_{ext} is the external perturbation force. The frame H and VF are the inertial frame and the virtual frame, respectively.

the object, instead of giving the control force directly, a desired impedance behaviour is specified in the object frame directly as follows,

$$\mathbf{f}_{ext} = M_d \ddot{\mathbf{x}} + D_d(\dot{\mathbf{x}} - \dot{\mathbf{x}}_r) + K_d(\mathbf{x} - \mathbf{x}_r) \quad (\text{A.2})$$

where $\mathbf{x}_r, \dot{\mathbf{x}}_r \in \mathbb{R}^6$ is the reference(desired) trajectory and $M_d, D_d, K_d \in \mathbb{R}^{6 \times 6}$ are the desired inertia, damping and stiffness matrix, respectively. From Eq. (A.1) and Eq.

(A.2), the object-level impedance control law is derived as

$$\mathbf{f}_{f,o} = ED_d(\dot{\mathbf{x}}_r - \dot{\mathbf{x}}) + EK_d(\mathbf{x}_r - \mathbf{x}) + (E - I)\mathbf{f}_{ext} \quad (\text{A.3})$$

where $E = M_0 M_d^{-1}$ and I is the identity matrix. In practice, it is often sufficient to keep the inertia unchanged, i.e., $M_0 = M_d$ and only shape the stiffness and damping. Then Eq. (A.3) can be simplified as:

$$\mathbf{f}_{f,o} = D_d(\dot{\mathbf{x}}_r - \dot{\mathbf{x}}) + K_d(\mathbf{x}_r - \mathbf{x}) \quad (\text{A.4})$$

STABLE GRASPING IMPEDANCE

To keep the grasp stable during manipulation, a stable grasping impedance is required to regulate the grasping forces. The contact model between the object and the fingertips is typically assumed to be point contact with friction, which can only transmit contact forces. Therefore, we only use one translational spring connecting each fingertip and the origin of the VF to represent the stable grasping impedance (stiffness), as shown in Fig. A.1. The grasping forces can be expressed as:

$$\mathbf{f}_{f,gi} = K_{gi}(\|\Delta\mathbf{p}_i\| - L_i) \frac{\Delta\mathbf{p}_i}{\|\Delta\mathbf{p}_i\|} \quad (\text{A.5})$$

where $\mathbf{f}_{f,gi}$ and K_{gi} are the grasping force and stable grasping stiffness at i -th fingertip. $\Delta\mathbf{p}_i = \mathbf{p}_o - \mathbf{p}_i$ with \mathbf{p}_i as the position of contact point on i -th fingertip and \mathbf{p}_o as the position of the origin of VF. L_i is the desired distance from the i -th fingertip to the origin of VF.

Instead of using the actual object frame, which requires external sensory information to estimate the object configuration, here the concept of VF is adopted, which is a function of all the contact points between object and fingertips, see Eq. (3.16, 3.17). VF can be used to estimate the actual object configuration if we assume that relative contact points between object and fingertips do not change. This assumption ignores the rolling and slipping effects between the fingertip and the object. However, these effect can be compensated if we can visual information to track the actual object configuration.

With the defined VF, one can compute the translational and rotational difference between the VF and the desired or reference frame. Thus from Eq. (A.4) and Eq. (A.5), the desired contact forces on each fingertip is the sum of manipulating forces and grasping forces:

$$\mathbf{f}_{f,i} = \mathbf{f}_{f,oi} + \mathbf{f}_{f,gi} \quad (\text{A.6})$$

Then the corresponding joint torques due to the contact forces at the fingertips can be obtained,

$$\boldsymbol{\tau}_i = J_{f,i}^T \mathbf{f}_{f,i} \quad (\text{A.7})$$

where $\boldsymbol{\tau}_i \in \mathbb{R}^{N_{f,i}}$ are the joint torques at i -th finger due to the manipulation task and $J_{f,i}$ is the Jacobian of the i -th finger.

STABILITY OF VARIABLE GRASP STIFFNESS CONTROL

In this part, we address the control stability of varying stiffness in robotic grasping using a two-fingered examples, as shown in Fig. B.1(a). To this end, a detailed formulation for the dynamics of the hand-object system is first derived, which takes rolling constraints and the soft fingertip into account, see (Arimoto, 2008) for more detailed formulation on the grasping dynamics. Then, a variable stiffness controller is presented to grasp the object stably and the upper bound for the change rate of the stiffness is derived from the proof of control stability.

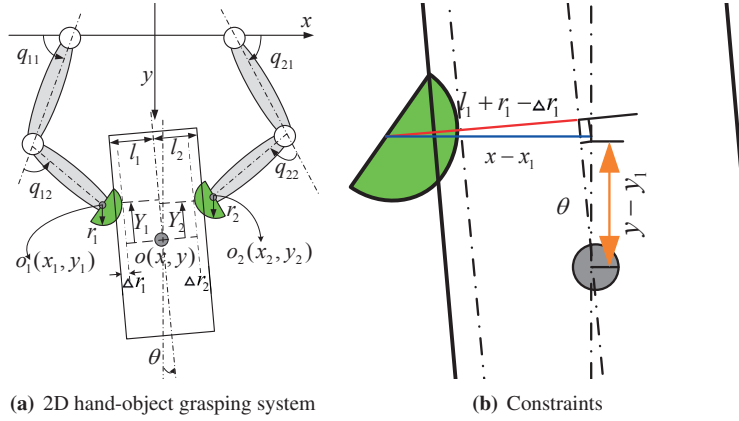


Figure B.1: (a) The hand-object system in 2D. Each finger has 2 DOFs with soft fingertips. (b) The constraint that the fingertip should keep contact with the object's surface.

B.1 DYNAMICS

In this part, we first consider the contact model of the fingertips and the constraints involved in the hand-object system. Then the overall dynamics of the system is formulated. The notations in this section are explained in Fig. B.1 to simplify the understanding of the grasping dynamics.

B.1.1 CONTACT MODEL OF SOFT FINGERTIP

$$f = c_1(\Delta r)^2 + c_2 \frac{d}{dt} \Delta r \quad (\text{B.1})$$

where c_1 and c_2 are positive constant parameters which depend on the material of the fingertip, and Δr is the deformation at the fingertip. The fingertip should keep contact with the object surface, as shown in Fig. B.1(b), which can be expressed as follows

$$l_1 + r_1 - \Delta r_1 = (x - x_1) \cos \theta - (y - y_1) \sin \theta \quad (\text{B.2})$$

$$l_2 + r_2 - \Delta r_2 = -(x - x_2) \cos \theta + (y - y_2) \sin \theta \quad (\text{B.3})$$

B.1.2 ROLLING CONSTRAINTS

The rolling constraints on each fingertip can be represented as

$$(r_i - \Delta r_i) \frac{d}{dt} \phi_i = - \frac{d}{dt} Y_i, \quad i = 1, 2 \quad (\text{B.4})$$

where Y_i and ϕ_i are given by:

$$Y_i = (x_i - x) \sin \theta + (y_i - y) \cos \theta \quad (\text{B.5})$$

$$q_{11} + q_{12} + \phi_1 = \pi + \theta \quad (\text{B.6})$$

$$q_{21} + q_{22} + \phi_2 = \pi - \theta \quad (\text{B.7})$$

B.1.3 OVERALL DYNAMICS

The total kinetic energy for the overall system can be described as follows

$$K = \sum_{i=1,2} \frac{1}{2} \dot{\mathbf{q}}_i^T H_i \dot{\mathbf{q}}_i + \frac{1}{2} M (\dot{x}^2 + \dot{y}^2) + \frac{1}{2} I \dot{\theta}^2 \quad (\text{B.8})$$

where $\mathbf{q}_i = [q_{i1}, q_{i2}]^T$ is the vector of finger joints and $H_i \in \mathbb{R}^{2 \times 2}$ is the inertia matrix for each finger, M and I are the mass and inertia matrix of the object respectively.

The total potential energy from deformation can be given as:

$$P = \sum_{i=1,2} \int_0^{\Delta r_i} c_1 \eta^2 d\eta \quad (\text{B.9})$$

Then from the Hamilton's principle, we have

$$\int_{t_0}^{t_1} \left[\delta(K - P) - \sum_{i=1,2} c_2 \Delta \dot{r}_i \frac{\partial \Delta r_i}{\partial X^T} \delta X + \sum_{i=1,2} \lambda_i [(r_i - \Delta r_i) \frac{\partial \phi_i}{\partial X^T} + \frac{\partial Y_i}{\partial X^T}] \delta X + \sum_{i=1,2} \mathbf{u}_i^T \delta \mathbf{q}_i \right] dt = 0 \quad (\text{B.10})$$

where $X = [\mathbf{q}_1^T, \mathbf{q}_2^T, x, y, \theta]^T$.

Then we have the overall dynamics for the object-hand system as follows

$$H_i(\mathbf{q}_i) \ddot{\mathbf{q}}_i + \left(\frac{1}{2} \dot{H}_i + S_i \right) \dot{\mathbf{q}}_i + f_i \frac{\partial \Delta r_i}{\partial \mathbf{q}_i^T} - \lambda_i [(r_i - \Delta r_i) \frac{\partial \phi_i}{\partial \mathbf{q}_i^T} + \frac{\partial Y_i}{\partial \mathbf{q}_i^T}] = \mathbf{u}_i \quad (\text{B.11})$$

$$M\ddot{x} + \sum_{i=1,2} [f_i \frac{\partial \Delta r_i}{\partial x} - \lambda_i \frac{\partial Y_i}{\partial x}] = 0 \quad (\text{B.12})$$

$$M\ddot{y} + \sum_{i=1,2} [f_i \frac{\partial \Delta r_i}{\partial y} - \lambda_i \frac{\partial Y_i}{\partial y}] = 0 \quad (\text{B.13})$$

$$I\ddot{\theta} + \sum_{i=1,2} [f_i \frac{\partial \Delta r_i}{\partial \theta} - \lambda_i ((r_i - \Delta r_i) \frac{\partial \phi_i}{\partial \theta} + \frac{\partial Y_i}{\partial \theta})] = 0 \quad (\text{B.14})$$

With the identities in section B.5, the overall dynamics can be simplified as

$$\begin{aligned} H_i(\mathbf{q}_i)\ddot{\mathbf{q}}_i + (\frac{1}{2}\dot{H}_i + S_i)\dot{\mathbf{q}}_i - (-1)^i f_i J_i^T \begin{bmatrix} \cos \theta \\ -\sin \theta \end{bmatrix} \\ - \lambda_i [(r_i - \Delta r_i) \begin{bmatrix} -1 \\ -1 \end{bmatrix} + J_i^T \begin{bmatrix} \sin \theta \\ \cos \theta \end{bmatrix}] = \mathbf{u}_i \end{aligned} \quad (\text{B.15})$$

$$M\ddot{x} - (f_1 - f_2) \cos \theta + (\lambda_1 + \lambda_2) \sin \theta = 0 \quad (\text{B.16})$$

$$M\ddot{y} + (f_1 - f_2) \sin \theta + (\lambda_1 + \lambda_2) \cos \theta = 0 \quad (\text{B.17})$$

$$I\ddot{\theta} - f_1 Y_1 + f_2 Y_2 + l_1 \lambda_1 - l_2 \lambda_2 = 0 \quad (\text{B.18})$$

B.2 VARIABLE GRASP STIFFNESS CONTROL

Motivated by the analysis of the overall system dynamics, the following control law is adopted for each finger to achieve stable grasp

$$u_i = -D_i \dot{\mathbf{q}}_i + k J_i^T (\mathbf{x}_i - \mathbf{x}_d) \quad (\text{B.19})$$

$$\mathbf{x}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad \mathbf{x}_d = \frac{1}{2} \begin{bmatrix} x_1 + x_2 \\ y_1 + y_2 \end{bmatrix} \quad (\text{B.20})$$

where D_i is a diagonal positive definite matrix representing the damping gain. $k \in \mathbb{R}^+$ represents the *variable* stiffness for each fingertip (k is the same value for the two-finger grasp to ensure force balance).

B.3 STABILITY PROOF-1

Taking the sum of inner product of Eq. (B.15) with $\dot{\mathbf{q}}_i$, $i = 1, 2$, Eq. (B.16) with \dot{x} , Eq. (B.17) with \dot{y} , Eq. (B.18) with $\dot{\theta}$, we have

$$\frac{d}{dt} E = - \sum_{i=1,2} \{\dot{\mathbf{q}}_i^T D_i \dot{\mathbf{q}}_i + c_2 \Delta \dot{r}_i^2\} + \frac{\dot{k}}{4} (\mathbf{x}_1 - \mathbf{x}_2)^T (\mathbf{x}_1 - \mathbf{x}_2) \quad (\text{B.21})$$

$$E = K + V + P \quad (\text{B.22})$$

$$K = \sum_{i=1,2} \frac{1}{2} \dot{\mathbf{q}}_i^T H_i \dot{\mathbf{q}}_i + \frac{1}{2} M(\dot{x}^2 + \dot{y}^2) + \frac{1}{2} I \dot{\theta}^2 \quad (\text{B.23})$$

$$P = \sum_{i=1,2} \int_0^{\Delta r_i} c_1 \eta^2 d\eta \quad (\text{B.24})$$

$$V = \frac{k}{4} (\mathbf{x}_1 - \mathbf{x}_2)^T (\mathbf{x}_1 - \mathbf{x}_2) \quad (\text{B.25})$$

As proved in Arimoto's book (Arimoto, 2008), the closed-loop dynamics is asymptotically stable if

$$\frac{d}{dt} E < 0 \quad (\text{B.26})$$

which leads to

$$\dot{k} < \frac{\sum_{i=1,2} \{\dot{\mathbf{q}}_i^T D_i \dot{\mathbf{q}}_i + c_2 \Delta \dot{r}_i^2\}}{(\mathbf{x}_1 - \mathbf{x}_2)^T (\mathbf{x}_1 - \mathbf{x}_2)} \quad (\text{B.27})$$

From this, we can conclude

- If the object is softer, namely c_2 is bigger, we can change the stiffness faster.
- If the object is smaller, i.e., $(\mathbf{x}_1 - \mathbf{x}_2)^T (\mathbf{x}_1 - \mathbf{x}_2)$ is smaller, we can change the grasping stiffness faster.

As noticed that the bound of change rate of the stiffness is still depending on the state variables, and thus in practice we need to design a state observer and filter the stiffness k according to the state. In next section, a conservative bound of the change rate of the stiffness will be found, which is independent of the state variables.

B.4 STABILITY PROOF-2

In Chapter 4, the desired stiffness is a function of the tactile sensing S and the rest length L , which is encoded using a probabilistic model GMM as follows

$$k = g(S, L) = \sum_{i=1}^m h_i [\mu_{k,i} + \Sigma_{ky,i} \Sigma_{y,i}^{-1} (y - \mu_{y,i})] \quad (\text{B.28})$$

Denote $y = [L^T, S^T]^T$, to prove Eq. (B.21), we need to compute the change rate of stiffness, i.e. \dot{k} ,

$$\dot{k} = \frac{\partial k}{\partial y} \left(\frac{\partial y}{\partial L} \frac{\partial L}{\partial t} + \frac{\partial y}{\partial S} \frac{\partial S}{\partial t} \right) \quad (\text{B.29})$$

$$= \frac{\partial k}{\partial y} \begin{bmatrix} \mathbf{x}_1^T J_1 \dot{\mathbf{q}}_1 + \mathbf{x}_2^T J_2 \dot{\mathbf{q}}_2 \\ \mathbf{x}_1^T J_1 \dot{\mathbf{q}}_1 + \mathbf{x}_2^T J_2 \dot{\mathbf{q}}_2 \\ \frac{\partial S}{\partial t} 2 \|\mathbf{x}_1 - \mathbf{x}_2\| \end{bmatrix} \frac{1}{2 \|\mathbf{x}_1 - \mathbf{x}_2\|} \quad (\text{B.30})$$

From the definition of Eq. (B.28), we have

$$\frac{\partial k}{\partial y} = \sum_{i=1}^m \left\{ \frac{\partial h_i}{\partial y} [\mu_{k,i} + \Sigma_{ky,i} \Sigma_{y,i}^{-1} (y - \mu_{y,i})] + h_i (\Sigma_{ky,i} \Sigma_{y,i}^{-1}) \right\} \quad (\text{B.31})$$

Assumption: During the change of stiffness, we assume the importance of each gaussian components does not change with respect to y , namely $\frac{\partial h_i}{\partial y} = 0$. This assumption implies a local linear controller is used to regulate the grasping stiffness according to Eq. (B.28).

With this assumption, we have

$$\frac{\partial k}{\partial y} = \sum_{i=1}^m h_i(\Sigma_{ky,i} \Sigma_{y,i}^{-1}) \quad (\text{B.32})$$

$$:= [A_{11}, A_{12}, A_{13}] \quad (\text{B.33})$$

Note that $A_{11} \in R, A_{12} \in R, A_{13} \in R^{n_s \times 1}$ are constant values or matrix, n_s is the dimension of the tactile readings. To prove $\frac{d}{dt}E < 0$, we only need to prove

$$\begin{aligned} [A_{11}, A_{12}, A_{13}] * \begin{bmatrix} \mathbf{x}_1^T J_1 \dot{\mathbf{q}}_1 + \mathbf{x}_2^T J_2 \dot{\mathbf{q}}_2 \\ \mathbf{x}_1^T J_1 \dot{\mathbf{q}}_1 + \mathbf{x}_2^T J_2 \dot{\mathbf{q}}_2 \\ \frac{\partial s}{\partial t} * 2 \|\mathbf{x}_1 - \mathbf{x}_2\| \end{bmatrix} & \frac{1}{2 \|\mathbf{x}_1 - \mathbf{x}_2\|} * \frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{4} \\ & - \sum_{i=1,2} \{\dot{\mathbf{q}}_i^T D_i \dot{\mathbf{q}}_i + c_2 \Delta \dot{r}_i^2\} < 0 \end{aligned} \quad (\text{B.34})$$

$$\begin{aligned} & \Leftarrow (A_{11} + A_{12}) \mathbf{x}_1^T J_1 \dot{\mathbf{q}}_1 + (A_{11} + A_{12}) \mathbf{x}_2^T J_2 \dot{\mathbf{q}}_2 + A_{13} \frac{\partial s}{\partial t} 2 \|\mathbf{x}_1 - \mathbf{x}_2\| < \\ & \dot{\mathbf{q}}_1^T \frac{2D_1}{\|\mathbf{x}_1 - \mathbf{x}_2\|} \dot{\mathbf{q}}_1 + \dot{\mathbf{q}}_2^T \frac{2D_2}{\|\mathbf{x}_1 - \mathbf{x}_2\|} \dot{\mathbf{q}}_2 + 2 \sum_{i=1,2} \{c_2 \Delta \dot{r}_i^2\} / \|\mathbf{x}_1 - \mathbf{x}_2\| \end{aligned} \quad (\text{B.35})$$

$$\begin{aligned} \Leftarrow A_{13} \frac{\partial s}{\partial t} & < \dot{\mathbf{q}}_1^T \frac{D_1}{\|\mathbf{x}_1 - \mathbf{x}_2\|^2} \dot{\mathbf{q}}_1 - \frac{A_{11} + A_{12}}{2 \|\mathbf{x}_1 - \mathbf{x}_2\|} \mathbf{x}_1^T J_1 \dot{\mathbf{q}}_1 + \\ & \dot{\mathbf{q}}_2^T \frac{D_1}{\|\mathbf{x}_1 - \mathbf{x}_2\|^2} \dot{\mathbf{q}}_2 - \frac{A_{11} + A_{12}}{2 \|\mathbf{x}_1 - \mathbf{x}_2\|} \mathbf{x}_2^T J_2 \dot{\mathbf{q}}_2 + \sum_{i=1,2} \{c_2 \Delta \dot{r}_i^2\} / \|\mathbf{x}_1 - \mathbf{x}_2\|^2 \end{aligned} \quad (\text{B.36})$$

$$\begin{aligned} \Leftarrow A_{13} \frac{\partial s}{\partial t} & + \left(\frac{A_{11} + A_{12}}{4} \right)^2 [\mathbf{x}_1^T J_1 D_1^{-1} J_1^T \mathbf{x}_1 + \mathbf{x}_2^T J_2 D_2^{-1} J_2^T \mathbf{x}_2] < \\ & \underbrace{\left\| \frac{D_1^{1/2}}{\|\mathbf{x}_1 - \mathbf{x}_2\|} \dot{\mathbf{q}}_1 - \mathbf{b}_1 \right\|^2 + \left\| \frac{D_2^{1/2}}{\|\mathbf{x}_1 - \mathbf{x}_2\|} \dot{\mathbf{q}}_2 - \mathbf{b}_2 \right\|^2 + \sum_{i=1,2} \{c_2 \Delta \dot{r}_i^2\} / \|\mathbf{x}_1 - \mathbf{x}_2\|^2}_{\text{denote this part as } \gamma \text{ and } \gamma > 0} \end{aligned} \quad (\text{B.37})$$

$$\Leftarrow A_{13} \frac{\partial s}{\partial t} + \left(\frac{A_{11} + A_{12}}{4} \right)^2 [\mathbf{x}_1^T J_1 D_1^{-1} J_1^T \mathbf{x}_1 + \mathbf{x}_2^T J_2 D_2^{-1} J_2^T \mathbf{x}_2] < \gamma \quad (\text{B.38})$$

Note that the first term in the left side of Eq. (B.38), i.e., $A_{13} \frac{\partial s}{\partial t}$ represents the change of stiffness due to the change of tactile sensing, if it is negative, Eq. (B.38) will keep holding and we don't need to change the damping to stabilize the system. When $A_{13} \frac{\partial s}{\partial t} > 0$, namely increasing the stiffness according to the tactile sensing, we need to set a proper

damping term to stabilize the system. From the bound of Jacobian (Ott, 2008), we have

$$\lambda_{max}(J_i J_i^T) < \sigma_{max}^2, i = 1, 2 \quad (\text{B.39})$$

Also, from the physical constraint of the finger, $\|\mathbf{x}_i\| < L_f, i = 1, 2$. Assume we use the same damping at each joint, $D_1 = D_2 = \text{diag}\{d\}$, then we have

$$d > \frac{2\sigma_{max}^2 L_f^2 (\frac{A_{11}+A_{12}}{4})^2}{\gamma - A_{13} \frac{\partial s}{\partial t}} \quad (\text{B.40})$$

Remark:

- γ is a parameter that is determined heuristically. If $\gamma \uparrow$, then $d \downarrow$. From the definition of γ , i.e., Eq. (B.37), it will increase if the softness of fingertip c_2 increases or the size of object decreases, i.e., $\|\mathbf{x}_1 - \mathbf{x}_2\| \downarrow$. This intuitively makes sense: if fingertip is softer, less damping is required.
- The value $\frac{\partial s}{\partial t}$ is determined by the sensitivity of the tactile sensor. In practice, most often a filter is used to preprocess the tactile data. The rate of change of tactile sensing $\frac{\partial s}{\partial t}$ can be directly obtained from the filter and the damping can be set accordingly. Another possibility is to assume a bound for $\|A_{13} \frac{\partial s}{\partial t}\| < \gamma_s$ and then damping d can be set using the maximal value γ_s .
- Note that the bound for d is still very conservative, due to the assumption made on γ and the neglect of relative high friction at each joint of the fingers. For all the experiments we have done so far in Chapter 4, we didn't observe any unstable behavior due to the change of the grasping stiffness.

B.5 IDENTITIES

From equations (A.2)-(A.7), we have the following identities, which are used in the derivation of the hand-object dynamics.

$$\frac{\partial \Delta r_1}{\partial \mathbf{q}_1^T} = J_1^T [\cos \theta, -\sin \theta]^T$$

$$\frac{\partial \phi_1}{\partial \mathbf{q}_1^T} = [-1, -1]^T$$

$$\frac{\partial Y_1}{\partial \mathbf{q}_1^T} = J_1^T [\sin \theta, \cos \theta]^T$$

$$\frac{\partial \Delta r_1}{\partial x} = -\cos \theta$$

$$\frac{\partial Y_1}{\partial x} = -\sin \theta$$

$$\frac{\partial \Delta r_1}{\partial y} = \sin \theta$$

$$\frac{\partial Y_1}{\partial y} = -\cos \theta$$

$$\frac{\partial \Delta r_1}{\partial \theta} = -Y_1$$

$$\frac{\partial \phi_1}{\partial \theta} = 1$$

$$\frac{\partial Y_1}{\partial \theta} = (x_1 - x) \cos \theta - (y_1 - y) \sin \theta$$

$$\frac{\partial Y_2}{\partial \theta} = (x_2 - x) \cos \theta - (y_2 - y) \sin \theta$$

$$\frac{\partial \Delta r_2}{\partial \mathbf{q}_2^T} = -J_2^T [\cos \theta, -\sin \theta]^T$$

$$\frac{\partial \phi_2}{\partial \mathbf{q}_2^T} = [-1, -1]^T$$

$$\frac{\partial Y_2}{\partial \mathbf{q}_2^T} = J_2^T [\sin \theta, \cos \theta]^T$$

$$\frac{\partial \Delta r_2}{\partial x} = \cos \theta$$

$$\frac{\partial Y_2}{\partial x} = -\sin \theta$$

$$\frac{\partial \Delta r_2}{\partial y} = -\sin \theta$$

$$\frac{\partial Y_2}{\partial y} = -\cos \theta$$

$$\frac{\partial \Delta r_2}{\partial \theta} = Y_2$$

$$\frac{\partial \phi_2}{\partial \theta} = -1$$

REFERENCES

- Alissandrakis, A., Nehaniv, C. L., and Dautenhahn, K. (2007). Correspondence mapping induced state and action metrics for robotic imitation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 37(2):299–307. [2.4.1](#)
- Arimoto, S. (2008). *Control Theory of Multi-fingered Hands: A Modelling and Analytical–Mechanics Approach for Dexterity and Intelligence*. Springer Science & Business Media. [B](#), [B.3](#)
- Bekiroglu, Y., Laaksonen, J., Jørgensen, J. A., Kyrki, V., and Kragic, D. (2011). Assessing grasp stability based on learning and haptic data. *IEEE Transactions on Robotics*, 27(3):616–629. [2.4.2](#), [4.3](#), [4.5.2](#)
- Bekiroglu, Y., Song, D., Wang, L., and Kragic, D. (2013). A probabilistic framework for task-oriented grasp stability assessment. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3040–3047. [2.4.2](#)
- Berenson, D., Abbeel, P., and Goldberg, K. (2012). A robot path planning framework that learns from experience. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3671–3678. [5.2.1](#)
- Berenson, D. and Srinivasa, S. S. (2008). Grasp synthesis in cluttered environments for dexterous hands. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 189–196. [2.2](#)
- Berenson, D., Srinivasa, S. S., Ferguson, D., Collet, A., and Kuffner, J. J. (2009a). Manipulation planning with workspace goal regions. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 618–624. [5.2.1](#)
- Berenson, D., Srinivasa, S. S., Ferguson, D., and Kuffner, J. J. (2009b). Manipulation planning on constraint manifolds. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 625–632. [5.2.1](#)
- Berenson, D., Srinivasa, S. S., and Kuffner, J. (2011). Task space regions: A framework for pose-constrained manipulation planning. *The International Journal of Robotics Research*. [5.2.1](#)
- Bicchi, A. (2000). Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity. *IEEE Transactions on Robotics and Automation*, 16(6):652–662. [1.1](#), [2](#), [3.2.1](#)
- Billard, A., Calinon, S., Dillmann, R., and Schaal, S. (2008). Robot programming by demonstration. In *Springer handbook of robotics*, pages 1371–1394. Springer. [2.4.1](#), [5.2.1](#)
- Bimbo, J., Kormushev, P., Althoefer, K., and Liu, H. (2015). Global estimation of an object’s pose using tactile sensing. *Advanced Robotics*, 29(5):363–374. [2.4.2](#)

- Bjorkman, M., Bekiroglu, Y., Hogman, V., and Kragic, D. (2013). Enhancing visual perception of shape through tactile glances. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 3180–3186. [2.2](#), [2.4.2](#), [3.2.2](#), [3.1](#)
- Bohg, J., Johnson-Roberson, M., Leon, B., Felip, J., Gratal, X., Bergstrom, N., Kragic, D., and Morales, A. (2011). Mind the gap - robotic grasping under incomplete observation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 686–693. [3.2.2](#), [3.6.1](#)
- Bookstein, F. L. (1989). Principal warps: Thin-Plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(6):567–585. [3.3.1](#)
- Borst, C., Fischer, M., and Hirzinger, G. (2002). Calculating hand configurations for precision and pinch grasps. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 1553–1559 vol.2. [2.2](#)
- Brost, R. C. (1985). Planning robot grasping motions in the presence of uncertainty. Technical Report CMU-RI-TR-85-12, Robotics Institute, Pittsburgh, PA. [2.2](#)
- Buchli, J., Stulp, F., Theodorou, E., and Schaal, S. (2011). Learning variable impedance control. *The International Journal of Robotics Research*, 30(7):820–833. [5.2.2](#)
- Buss, M., Hashimoto, H., and Moore, J. (1996). Dextrous hand grasping force optimization. *IEEE Transactions on Robotics and Automation*, 12(3):406–418. [2.3](#)
- Castiello, U. (2005). The neuroscience of grasping. *Nature Reviews Neuroscience*, 6(9):726–736. [1.1](#), [4.1](#)
- Chalon, M., Reinecke, J., and Pfanne, M. (2013). Online in-hand object localization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2977–2984. [2.4.2](#)
- Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58. [4.3](#), [4.5.2](#)
- Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>. [4.3](#)
- Chavan-Dafle, N. and Rodriguez, A. (2015). Prehensile pushing: In-hand manipulation with push-primitives. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [6.3](#)
- Cheng, M.-Y. and Wang, Y.-H. (2009). Velocity field construction for contour following tasks represented in nurbs form. *IEEE Transactions on Automatic Control*, 54(10):2405–2410. [5.2.2](#)
- Christopoulos, V. N. and Schrater, P. R. (2007). Handling shape and contact location uncertainty in grasping two-dimensional planar objects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1557–1563. [2.2](#), [3.1](#)
- Ciocarlie, M., Hicks, F. M., Holmberg, R., Hawke, J., Schlicht, M., Gee, J., Stanford, S., and Bahadur, R. (2014). The velo gripper: A versatile single-actuator design for enveloping, parallel and fingertip grasps. *The International Journal of Robotics Research*, 33(5):753–767. [4.7](#), [6.2](#)
- Ciocarlie, M., Hsiao, K., Jones, E. G., Chitta, S., Rusu, R. B., and Sucan, I. A. (2010). Towards reliable grasping and manipulation in household environments. In *International Symposium on Experimental Robotics (ISER)*, New Delhi, India. [3.2.2](#)

- Ciocarlie, M. T. and Allen, P. K. (2009). Hand posture subspaces for dexterous robotic grasping. *The International Journal of Robotics Research*, 28(7):851–867. [2.2](#)
- Cohn, D. A., Ghahramani, Z., and Jordan, M. I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145. [4.4.1](#)
- Cutkosky, M. R. and Howe, R. D. (1990). Human grasp choice and robotic grasp analysis. *Dextrous robot hands*, pages 5–31. [2.4.1](#)
- Dafle, N. C., Rodriguez, A., Paolini, R., Tang, B., Srinivasa, S. S., Erdmann, M., Mason, M. T., Lundberg, I., Staab, H., and Fuhlbrigge, T. (2014). Extrinsic dexterity: In-hand manipulation with external forces. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1578–1585. [6.3](#)
- Dang, H. and Allen, P. (2013). Stable grasping under pose uncertainty using tactile feedback. *Autonomous Robots*, pages 1–22. [2.3](#), [2.4.2](#), [3.1](#), [4.3](#)
- De Souza, R. L., El-Khoury, S., Santos-Victor, J., and Billard, A. (2015). Recognizing the grasp intention from human demonstration. *Robotics and Autonomous Systems*. [2.4.1](#), [3.2.1](#)
- Deimel, R. and Brock, O. (2015). A novel type of compliant and underactuated robotic hand for dexterous grasping. *The International Journal of Robotics Research*. [4.7](#), [6.2](#)
- Diankov, R. (2010). *Automated Construction of Robotic Manipulation Programs*. PhD thesis, Carnegie Mellon University, Robotics Institute. [3.4.2](#), [3.6.3](#)
- Ding, D. and Wang, S. (2001). Computation of 3-d form-closure grasps. *IEEE Transactions on Robotics and Automation*, 17(4):515–522. [3.2.1](#)
- Dogar, M. and Srinivasa, S. (2011). A framework for push-grasping in clutter. *Robotics: Science and Systems VII*. [1.2](#)
- Dollar, P., Rabaud, V., and Belongie, S. J. (2007). Non-isometric manifold learning: analysis and an algorithm. In Ghahramani, Z., editor, *ICML*, volume 227 of *ACM International Conference Proceeding Series*, pages 241–248. ACM. [5.3](#), [5.3](#)
- Dragiev, S., Toussaint, M., and Gienger, M. (2011). Gaussian process implicit surfaces for shape estimation and grasping. In *IEEE International Conference on Robotics and Automation (ICRA)*. [3.2.1](#), [3.3.1](#)
- Dragiev, S., Toussaint, M., and Gienger, M. (2013). Uncertainty aware grasping and tactile exploration. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 113–119. [2.2](#), [2.4.2](#), [3.1](#), [3.3.1](#)
- Ekvall, S. and Kragic, D. (2004). Interactive grasp learning based on human demonstration. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 3519–3524. [2.4.1](#)
- Ekvall, S. and Kragic, D. (2007). Learning and evaluation of the approach vector for automatic grasp generation and planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4715–4720. [2.4.1](#)
- El-Khasawneh, B. S. and Ferreira, P. M. (1999). Computation of stiffness and stiffness bounds for parallel link manipulators¹. *International Journal of Machine Tools and Manufacture*, 39(2):321 – 342. [5.5.2](#)
- El-Khoury, S., De Souza, R., and Billard, A. (2015). On computing task-oriented grasps. *Robotics and Autonomous Systems*, 66:145–158. [2.4.1](#), [3.2.1](#)

- El Khoury, S., Li, M., and Billard, A. (2012). Bridging the gap: One shot grasp synthesis approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2027–2034. [1.3](#), [2.2](#), [3.1](#), [3.2.1](#)
- El Khoury, S., Li, M., and Billard, A. (2013). On the generation of a variety of grasps. *Robotics and Autonomous Systems*, 61(12):1335–1349. [1.3](#), [2.2](#), [3.1](#), [3.2.1](#), [3.2.2](#), [3.3.2](#)
- Eppner, C., Deimel, R., Alvarez-Ruiz, J., Maertens, M., and Brock, O. (2015). Exploitation of environmental constraints in human and robotic grasping. *The International Journal of Robotics Research*, 34(7):1021–1038. <http://ijr.sagepub.com/content/34/7/1021>. [6.3](#)
- Fearing, R. (1986). Simplified grasping and manipulation with dextrous robot hands. *IEEE Journal of Robotics and Automation*, 2(4):188–195. [4.2.1](#)
- Felip, J. and Morales, A. (2009). Robust sensor-based grasp primitive for a three-finger robot hand. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*. [2.3](#), [3.1](#)
- Ferrari, C. and Canny, J. (1992). Planning optimal grasps. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2290–2295 vol.3. [2.2](#), [3.2.1](#)
- Goldfeder, C. and Allen, P. K. (2011). Data-driven grasping. *Autonomous Robots*, 31(1):1–20. [2.2](#), [3.2.1](#), [3.2.2](#)
- Goodwine, B. and Burdick, J. (1998). Gait controllability for legged robots. In *International Conference on Robotics and Automation (ICRA)*, volume 1, pages 484–489 vol.1. [4.7](#)
- Gori, I., Pattacini, U., Tikhanoff, V., and Metta, G. (2014). Three-finger precision grasp on incomplete 3d point clouds. In *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China. [3.2.2](#), [3.6.1](#)
- Grochow, K., Martin, S. L., Hertzmann, A., and Popović, Z. (2004). Style-based inverse kinematics. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 522–531. ACM. [6.2](#)
- Haidacher, S. and Hirzinger, G. (2003). Estimating finger contact location and object pose from contact measurements in 3d grasping. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1805–1810. [2.4.2](#)
- Hang, K., Haustein, J., Li, M., Billard, A., Smith, C., and Kragic, D. (2016). On the evolution of fingertip grasping manifolds. In *IEEE International Conference on Robotics and Automation (ICRA)*. (accepted). [3.4.2](#), [6.2](#)
- Hang, K., Li, M., Stork, J. A., Bekiroglu, Y., Billard, A., and Kragic, D. (2014a). Hierarchical fingertip space for synthesizing adaptable fingertip grasps. In *ICRA 2014 Workshop: Autonomous Grasping and Manipulation: An Open Challenge*. [1.3](#), [4.6](#), [4.18](#)
- Hang, K., Li, M., Stork, J. A., Bekiroglu, Y., Pokorny, F. T., Billard, A., and Kragic, D. (2015). Hierarchical fingertip space: A unified framework for grasp planning and in-hand grasp adaptation. *IEEE Transactions on Robotics*. (Conditional accepted). [1.3](#), [4.1](#), [4.2.3](#), [4.6](#), [4.6.1](#)
- Hang, K., Stork, J., Kragic, D., et al. (2014b). Hierarchical fingertip space for multi-fingered precision grasping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, pages 1641–1648. [1.3](#), [3.2.1](#), [4.1](#)

- Havoutis, I. and Ramamoorthy, S. (2013). Motion planning and reactive control on learnt skill manifolds. *The International Journal of Robotics Research*. 5.2.1, 5.3
- Hsiao, K., Chitta, S., Ciocarlie, M., and Jones, E. (2010). Contact-reactive grasping of objects with partial shape information. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*. 2.3, 3.1
- Huang, B., El-Khoury, S., Li, M., Bryson, J. J., and Billard, A. (2013). Learning a real time grasping strategy. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 593–600. 2.4.1, 4.3
- Huang, B., Li, M., De Souza, R., Bryson, J., and Billard, A. (2015). A modular approach to learning manipulation strategies from human demonstration. *Autonomous Robots*, pages 1–25. 5.2.2
- Huebner, K. and Kragic, D. (2008). Selection of robot pre-grasps using box-based shape approximation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1765–1770. 2.4.1
- J. Bohg, A. Morales, T. A. and Kragic, D. (2014). Data-driven grasp synthesis - a survey. *IEEE Transactions on Robotics*, 30(2):289–309. 1.1, 2.4.1, 3.2.1
- Jen, F., Shoham, M., and Longman, R. W. (1996). Liapunov stability of force-controlled grasps with a multi-fingered hand. *The International journal of robotics research*, 15(2):137–154. 4.2.1
- Johansson, R. S. (1998). Sensory input and control of grip. In *Novartis Foundation 218: Sensory Guidance of Movement*, pages 45–63. 4.4.1
- Johansson, R. S. and Flanagan, J. R. (2009). Coding and use of tactile signals from the fingertips in object manipulation tasks. *Nature Reviews Neuroscience*, 10(5):345–359. 1.1, 4.1
- Kang, S. B. (1994). *Robot instruction by human demonstration*. PhD thesis, Citeseer. 2.4.1
- Kanno, H., Nakamoto, H., Kobayashi, F., Kojima, F., and Fukui, W. (2013). Slip detection using robot fingertip with 6-axis force/torque sensor. In *IEEE Workshop on Robotic Intelligence In Informationally Structured Space (RiiSS)*, pages 1–6. 2.4.2, 4.2.1
- Kao, I. and Cutkosky, M. R. (1992). Quasistatic manipulation with compliance and sliding. *The International journal of robotics research*, 11(1):20–40. 2.1
- Kavraki, L. E., Švestka, P., Latombe, J.-C., and Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580. 5.2.1
- Kim, B.-H., Yi, B.-J., Oh, S.-R., and Suh, I. H. Fundamentals and analysis of compliance characteristics for multifingered hands. In *International Conference on Robotics and Automation (ICRA)*. 5.2.2
- Kim, B.-H., Yi, B.-J., Oh, S.-R., and Suh, I. H. Task-based compliance planning for multifingered hands. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2001. 5.2.2
- Kim, J., Iwamoto, K., Kuffner, J., Ota, Y., and Pollard, N. (2013). Physically based grasp quality evaluation under pose uncertainty. *IEEE Transactions on Robotics*, 29(6):1424–1439. 1.1, 2.2, 3.2.2, 4.2.1

- Kishi, Y., Luo, Z., Asano, F., and Hosoe, S. (2003). Passive impedance control with time-varying impedance center. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, volume 3, pages 1207–1212. [5.2.2](#)
- Ko, I., Kim, B., and Park, F. C. (2014). Randomized path planning on vector fields. *The International Journal of Robotics Research*, 33(13):1664–1682. [5.2.1](#), [5.3](#)
- Kopicki, M., Detry, R., Schmidt, F., Borst, C., Stolkin, R., and Wyatt, J. L. (2014). Learning dexterous grasps that generalise to novel objects by combining hand and contact models. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5358–5365. [2.4.1](#)
- Krainin, M., Henry, P., Ren, X., and Fox, D. (2011). Manipulator and object tracking for in-hand 3d object modeling. *The International Journal of Robotics Research*, 30(11):1311–1327. [2.4.2](#)
- Krakauer, J. W. and Mazzoni, P. (2011). Human sensorimotor learning: adaptation, skill, and beyond. *Current opinion in neurobiology*, 21(4):636–644. [1.1](#)
- Kronander, K. and Billard, A. (2012). Online learning of varying stiffness through physical human-robot interaction. In *International Conference on Robotics and Automation (ICRA)*. [5.2.2](#)
- Kronander, K. and Billard, A. (2016). Passive interaction control with dynamical systems. *IEEE Robotics and Automation Letters*, vol. 1, iss. 1, Jan. 2016, pp. 106–113. [4.7](#)
- Kuo, P.-H., DeBacker, J., and Deshpande, A. (2015). Design of robotic fingers with human-like passive parallel compliance. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2562–2567. [5.6.2](#)
- Laaksonen, J., Nikandrova, E., and Kyrki, V. (2012). Probabilistic sensor-based grasping. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*. [2.2](#)
- Li, M., Bekiroglu, Y., Kragic, D., and Billard, A. (2014a). Learning of grasp adaptation through experience and tactile sensing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3339–3346. [1.3](#), [2.4.2](#), [3.1](#), [3.2.3](#), [3.5](#), [4.1](#)
- Li, M., El Khoury, S., and Billard, A. (2013a). Synergy-level grasp synthesis learning. In *ICRA 2013 Workshop: Hand synergy-how to tame the complexity of grasping*. [3.1](#)
- Li, M., Hang, K., Kragic, D., and Billard, A. (2015a). Dexterous grasping under shape uncertainty. *Robotics and Autonomous Systems*, 75:352–364. [1.3](#), [3.1](#), [3.2.1](#)
- Li, M., Tahara, K., and Billard, A. (2015b). Learning task manifold for constrained object manipulation. (In submission). [5.1](#)
- Li, M., Yang, W., and Zhang, X. (2013b). Robust actuator force analysis of a heavy-duty manipulator using gmm/gmr. *International Journal of Robotics and Automation*, 28(4):349–356. [3.2.1](#)
- Li, M., Yin, H., Tahara, K., and Billard, A. (2014b). Learning object-level impedance control for robust grasping and dexterous manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6784–6791. [1.3](#), [2.3](#), [2.4.2](#), [3.4.1](#), [3.6.5](#), [4.3](#), [5.1](#), [5.2.2](#), [5.6.2](#), [A](#)

- Li, P. Y. and Horowitz, R. (1999). Passive velocity field control of mechanical manipulators. *IEEE Transactions on Robotics and Automation*, 15(4):751–763. [5.2.2](#)
- Li, P. Y. and Horowitz, R. (2001a). Passive velocity field control (pvfc). part i. geometry and robustness. *IEEE Transactions on Automatic Control*, 46(9):1346–1359. [5.2.2](#)
- Li, P. Y. and Horowitz, R. (2001b). Passive velocity field control (pvfc). part ii. application to contour following. *IEEE Transactions on Automatic Control*, 46(9):1360–1371. [5.2.2](#)
- Li, Y., Saut, J.-P., Pettre, J., Sahbani, A., Bidaud, P., and Multon, F. (2013c). Fast grasp planning by using cord geometry to find grasping points. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3265–3270. IEEE. [2.2](#)
- Li, Z., Hsu, P., and Sastry, S. (1989). Grasping and coordinated manipulation by a multifingered robot hand. *The International Journal of Robotics Research*, 8(4):33–50. [2.3](#)
- Lin, Y. and Sun, Y. (2014). Robot grasp planning based on demonstrated grasp strategies. *The International Journal of Robotics Research*, page 0278364914555544. [2.4.1](#)
- Lin, Y. and Sun, Y. (2015). Grasp planning to maximize task coverage. *The International Journal of Robotics Research*. [2.4.1](#)
- Liu, H., Nguyen, K. C., Perdereau, V., Bimbo, J., Back, J., Godden, M., Seneviratne, L. D., and Althoefer, K. (2015). Finger contact sensing and the application in dexterous hand manipulation. *Autonomous Robots*, 39(1):25–41. [2.4.2](#)
- Liu, H., Song, X., Nanayakkara, T., Althoefer, K., and Seneviratne, L. (2011). Friction estimation based object surface classification for intelligent manipulation. In *IEEE ICRA 2011 workshop on autonomous grasping, Shanghai*. [2.4.2](#)
- Melachlan, G. and Peel, D. (2000). *Finite Mixture Models*. Wiley Series in Probability and Statistics. Wiley-Interscience, 1 edition. [3.4.2](#)
- Miller, A. T. and Allen, P. K. (1999). Examples of 3d grasp quality computations. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1240–1246. [2.2](#)
- Miller, A. T., Knoop, S., Christensen, H. I., and Allen, P. K. (2003). Automatic grasp planning using shape primitives. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1824–1829. [2.2](#), [2.4.1](#)
- Mirtich, B. and Canny, J. (1994). Easily computable optimum grasps in 2-d and 3-d. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 739–747 vol.1. [2.2](#)
- Montana, D. J. (1988). The kinematics of contact and grasp. *The International Journal of Robotics Research*, 7(3):17–32. [2.1](#)
- Montana, D. J. (1991). The condition for contact grasp stability. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 412–417. [2.4.2](#), [4.2.1](#)
- Murray, R. M., Li, Z., Sastry, S. S., and Sastry, S. S. (1994). *A mathematical introduction to robotic manipulation*. CRC press. [2.1](#), [2.1](#), [3.3.2](#)
- Mussa-Ivaldi, F. A. and Giszter, S. F. (1992). Vector field approximation: a computational paradigm for motor control and learning. *Biological cybernetics*, 67(6):491–500. [5.3](#), [6.2](#)

- Nakamura, Y., Nagai, K., and Yoshikawa, T. (1989). Dynamics and stability in coordination of multiple robotic mechanisms. *The International Journal of Robotics Research*, 8(2):44–61. [2.4.2](#), [4.2.1](#), [4.2.1](#)
- Napier, J. R. (1956). The prehensile movements of the human hand. *Journal of bone and joint surgery*, 38(4):902–913. [2.2](#)
- Nguyen, K.-C. and Perdereau, V. (2013). Fingertip force control based on max torque adjustment for dexterous manipulation of an anthropomorphic hand. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [2.3](#)
- Nguyen, V. (1987). Constructing stable grasps in 3D. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 234–239. [2.2](#)
- Nguyen-Tuong, D., Peters, J. R., and Seeger, M. (2009). Local gaussian process regression for real time online model learning. In *Advances in Neural Information Processing Systems*, pages 1193–1200. [2.4.2](#)
- Odhner, L. U., Jentoft, L. P., Claffee, M. R., Corson, N., Tenzer, Y., Ma, R. R., Buehler, M., Kohout, R., Howe, R. D., and Dollar, A. M. (2014). A compliant, underactuated hand for robust manipulation. *The International Journal of Robotics Research*, 33(5):736–752. [4.7](#), [6.2](#)
- Okamura, A. M., Smaby, N., and Cutkosky, M. R. (2000). An overview of dexterous manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 255–262. [5.6](#)
- Oriolo, G., Ottavi, M., and Vendittelli, M. (2002). Probabilistic motion planning for redundant robots along given end-effector paths. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 1657–1662. [5.2.1](#)
- Ott, C. (2008). *Cartesian impedance control of redundant and flexible-joint robots*. Springer. [5.2.2](#), [B.4](#)
- Paolini, R., Rodriguez, A., Srinivasa, S. S., and Mason, M. T. (2014). A data-driven statistical framework for post-grasp manipulation. *The International Journal of Robotics Research*, 33(4):600–615. [1.2](#), [2.4.2](#)
- Pelosofof, R., Miller, A., Allen, P., and Jebara, T. (2004). An svm learning approach to robotic grasping. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 3512–3518 Vol.4. [2.4.1](#)
- Phillips, M., Cohen, B. J., Chitta, S., and Likhachev, M. (2012). E-graphs: Bootstrap-planning with experience graphs. In *Robotics: Science and Systems*. [5.2.1](#)
- Phillips, M., Hwang, V., Chitta, S., and Likhachev, M. (2013). Learning to plan for constrained manipulation from demonstrations. In *Robotics: Science and Systems*. [5.2.1](#)
- Ponce, J., Sullivan, S., Sudsang, A., Boissonnat, J.-D., and Merlet, J.-P. (1997). On computing four-finger equilibrium and force-closure grasps of polyhedral objects. *The International Journal of Robotics Research*, 16(1):11–35. [3.2.1](#)
- Porta, J. M., Jaillet, L., and Bohigas, O. (2011). Randomized path planning on manifolds based on higher-dimensional continuation. *The International Journal of Robotics Research*, page 0278364911432324. [5.2.1](#)

- Powell, M. (1994). A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in optimization and numerical analysis*, eds. S. Gomez and J-P. Hennart, pages 51–67. Kluwer Academic (Dordrecht). [3.6.3](#)
- Prattichizzo, D. (2014). Robot grasp control. In Baillieul, J. and Samad, T., editors, *Encyclopedia of Systems and Control*, pages 1–9. Springer London. [2.3](#)
- Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press. [3.3.1](#), [3.3.1](#)
- Ritter, H. and Haschke, R. (2015). Hands, dexterity, and the brain. In *Humanoid Robotics and Neuroscience: Science, Engineering and Society*, Frontiers in Neuro-engineering. CRC Press, Boca Raton (FL). [2.4.2](#)
- Roa, M. and Suarez, R. (2009). Computation of independent contact regions for grasping 3-d objects. *IEEE Transactions on Robotics*, 25(4):839–850. [2.2](#), [3.2.1](#), [3.1](#)
- Roa, M. A. and Suárez, R. (2014). Grasp quality measures: review and performance. *Autonomous Robots*, pages 1–24. [3.2.1](#), [3.3.2](#)
- Rodriguez, A., Basaez, L., and Celaya, E. (2008). A relational positioning methodology for robot task specification and execution. *IEEE Transactions on Robotics*, 24(3):600–611. [5.2.1](#)
- Romano, J., Hsiao, K., Niemeyer, G., Chitta, S., and Kuchenbecker, K. (2011). Human-inspired robotic grasp control with tactile sensing. *IEEE Transactions on Robotics*, 27(6):1067–1079. [2.3](#), [2.4.2](#)
- Rosales, C., Ajoudani, A., Gabiccini, M., and Bicchi, A. (2014). Active gathering of frictional properties from objects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3982–3987. [2.4.2](#)
- Rosales, C., Ros, L., Porta, J. M., and Suarez, R. (2011). Synthesizing grasp configurations with specified contact regions. *The International Journal of Robotics Research*, 30(4):431–443. [2.2](#), [2.4.1](#), [3.1](#)
- Roy, N., Newman, P., and Srinivasa, S. (2013). *Robust object grasping using force compliant motion primitives*, page 504. MIT Press. [6.3](#)
- Sahbani, A., El-Khoury, S., and Bidaud, P. (2012). An overview of 3d object grasp synthesis algorithms. *Robotics and Autonomous Systems*, 60(3):326–336. [3.2.1](#)
- Saitoh, Y., Luo, Z., and Watanabe, K. (2003). Adaptive modular vector field control for robot contact tasks in uncertain environment. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 3645–3650. [5.2.2](#)
- Salviati, G., Malvezzi, M., Gioioso, G., and Prattichizzo, D. (2014). On the use of homogeneous transformations to map human hand movements onto robotic hands. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5352–5357. [2.4.1](#)
- Santello, M., Flanders, M., and Soechting, J. F. (1998). Postural hand synergies for tool use. *The Journal of Neuroscience*. [2.2](#)
- Sausser, E. L., Argall, B., Metta, G., and Billard, A. (2012). Iterative learning of grasp adaptation through human corrections. *Robotics and Autonomous Systems*, 60(1):55–71. [2.4.2](#), [4.3](#)

- Saut, J.-P., Ivaldi, S., Sahbani, A., and Bidaud, P. (2014). Grasping objects localized from uncertain point cloud data. *Robotics and Autonomous Systems*, 62(12):1742 – 1754. [2.2](#), [3.1](#)
- Schneider, S. A. and Cannon, R. H. (1992). Object impedance control for cooperative manipulation: theory and experimental results. *IEEE Transactions on Robotics and Automation*, 8(3):383–394. [2.3](#), [5.2.2](#)
- Schölkopf, B., Platt, J. C., Shawe-Taylor, J. C., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13(7):1443–1471. [4.3](#)
- Shimoga, K. and Goldenberg, A. (1991). Grasp admittance center: Choosing admittance center parameters. In *American Control Conference*, pages 2527–2532. [5.2.2](#)
- Shukla, A. and Billard, A. (2012). Augmented-svm: Automatic space partitioning for combining multiple non-linear dynamics. In Bartlett, P., Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 25*, pages 1025–1033. [6.2](#)
- Siciliano, B. and Khatib, O. (2008). *Springer handbook of robotics*. Springer Science & Business Media. [2.1](#), [3.3.2](#), [3.3.2](#), [5.2.2](#)
- Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2008). *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 1st edition. [2.3](#), [5.2.2](#)
- Sikka, P. and McCarragher, B. J. (1997). Stiffness-based understanding and modeling of contact tasks by human demonstration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [5.2.2](#)
- Solak, E., Murray-Smith, R., Leithead, W. E., Leith, D., and Rasmussen, C. E. (2003). Derivative observations in gaussian process models of dynamic systems. In Thrun, S., Becker, S., and Obermayer, K., editors, *Advances in Neural Information Processing Systems 15*, pages 1033–1040, Cambridge, MA. MIT Press. [3.3.1](#)
- Sommer, N., Li, M., and Billard, A. (2014). Bimanual compliant tactile exploration for grasping unknown objects. In *International Conference on Robotics and Automation (ICRA)*. [2.2](#), [2.4.2](#), [3.2.2](#), [3.6.1](#), [4.19](#), [5.4.3](#)
- Song, D., Ek, C. H., Huebner, K., and Kragic, D. (2015). Task-based robot grasp planning using probabilistic inference. *IEEE Transactions on Robotics*. [2.4.1](#), [3.2.1](#)
- Stilman, M. (2010). Global manipulation planning in robot joint space with task constraints. *IEEE Transactions on Robotics*, 26(3):576–584. [5.2.1](#)
- Strub, C., Worgotter, F., Ritter, H., and Sandamirskaya, Y. (2014). Using haptics to extract object shape from rotational manipulations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2179–2186. [2.4.2](#)
- Şucan, I., Chitta, S., et al. (2012). Motion planning with constraints using configuration space approximations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1904–1910. [5.2.1](#)
- Suh, C., Um, T., Kim, B., Noh, H., Kim, M., and Park, F. (2011). Tangent space rrt: A randomized planning algorithm on constraint manifolds. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4968–4973. [5.2.1](#)

- Tahara, K., Arimoto, S., and Yoshida, M. (2010). Dynamic object manipulation using a virtual frame by a triple soft-fingered robotic hand. In *IEEE International Conference on Robotics and Automation (ICRA)*. [2.4.1](#), [3.4.1](#), [4.3](#), [5.2.2](#)
- Tahara, K., Maruta, K., Kawamura, A., and Yamamoto, M. (2012). Externally sensorless dynamic regrasping and manipulation by a triple-fingered robotic hand with torsional fingertip joints. In *IEEE International Conference on Robotics and Automation (ICRA)*. [2.3](#), [2.4.2](#), [A](#)
- Takahashi, T., Tsuboi, T., Kishida, T., Kawanami, Y., Shimizu, S., Iribe, M., Fukushima, T., and Fujita, M. (2008). Adaptive grasping by multi fingered hand with tactile sensor based on robust force and position control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 264–271. [2.3](#), [2.4.2](#), [3.1](#), [4.2.1](#)
- Trinkle, J. C. (1989). A quasi-static analysis of dextrous manipulation with sliding and rolling contacts. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 788–793. [2.1](#)
- Veiga, F., van Hoof, H., Peters, J., and Hermans, T. (2015). Stabilizing novel objects by learning to predict tactile slip. In *IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*. [4.2.1](#)
- Wachter, A. and Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57. [3.3.2](#)
- Wenyu, Y., Miao, L., and Xiaoping, Z. (2010). Robust robotic grasping force optimization with uncertainty. In *International Conference on Intelligent Robotics and Applications (ICIRA)*, pages 264–275. Springer. [3.2.1](#)
- Wettels, N., Santos, V. J., Johansson, R. S., and Loeb, G. E. (2008). Biomimetic tactile sensor array. *Advanced Robotics*, 22(8):829–849. [2.3](#)
- Wimböck, T., Ott, C., Albu-Schäffer, A., and Hirzinger, G. (2012). Comparison of object-level grasp controllers for dynamic dexterous manipulation. *The International Journal of Robotics Research*, 31(1):3–23. [2.3](#), [4.2.1](#), [5.2.2](#), [5.6.2](#)
- Wimbock, T., Ott, C., and Hirzinger, G. (2008). Analysis and experimental evaluation of the intrinsically passive controller (IPC) for multifingered hands. In *IEEE International Conference on Robotics and Automation (ICRA)*. [2.3](#), [5.2.2](#)
- Wolpert, D. M., Diedrichsen, J., and Flanagan, J. R. (2011). Principles of sensorimotor learning. *Nature Reviews Neuroscience*, 12(12):739–751. [1.1](#)
- Xiong, C.-H., Li, Y.-F., Ding, H., and Xiong, Y.-L. (1999). On the dynamic stability of grasping. *The International Journal of Robotics Research*, 18(9):951–958. [2.4.2](#)
- Yang, B.-H. and Asada, H. (1996). Progressive learning and its application to robot impedance learning. *IEEE Transactions on Neural Networks*, 7(4):941–952. [5.2.2](#)
- Yao, Z. and Gupta, K. (2005). Path planning with general end-effector constraints: Using task space to guide configuration space search. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1875–1880. [5.2.1](#)
- Yoshikawa, T. (2010). Multifingered robot hands: Control for grasping and manipulation. *Annual Reviews in Control*, 34(2):199 – 208. [2.3](#), [2.4.2](#)

- Yoshikawa, T. and Zheng, X.-Z. (1993). Coordinated dynamic hybrid position/force control for multiple robot manipulators handling one constrained object. *The International Journal of Robotics Research*, 12(3):219–230. [2.3](#)
- Yousef, H., Boukallel, M., and Althoefer, K. (2011). Tactile sensing for dexterous in-hand manipulation in robotics—A review. *Sensors and Actuators A: physical*, 167(2):171–187. [2.4.2](#)
- Yun, Y. and Deshpande, A. D. (2014). Control in the reliable region of a statistical model with gaussian process regression. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 654–660. [2.4.2](#)
- Zheng, Y., Lin, M., and Manocha, D. (2012). On computing reliable optimal grasping forces. *IEEE Transactions on Robotics*, 28(3):619–633. [2.3](#)
- Zheng, Y. and Qian, W.-H. (2005). Coping with the grasping uncertainties in force-closure analysis. *The International Journal of Robotic Research*, 24(4):311–327. [2.2](#), [4.2.1](#)
- Zhu, X. and Ding, H. (2004). Planning force-closure grasps on 3-d objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1258–1263 Vol.2. [2.2](#)

CURRICULUM VITAE

Miao Li

LASA, EPFL

ME A3 31

CH 1015, Lausanne, Switzerland

Phone: (041) 078-670-9643

Email: miao.li@epfl.ch

Homepage: <http://lasa.epfl.ch/>

Research Interests

My research interests are in robotics, machine learning and applied nonlinear control. They encompass **robot learning, robotic grasping and manipulation, human robot interaction, robotic hand and tactile sensing, and neuroscience**. I am particularly interested in finding the deep connections between dynamics of mechanical systems and learning algorithms (learning from humans or from optimizations), which enables adaptive, efficient and robust control design for complex mechanical systems. The goal of my research is to enable robots to perform skills with the level of dexterity and flexibility that human demonstrates in similar tasks.

Education

2011-2015.12 Ph.D Student, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland.

Thesis: Dynamic grasp adaptation – from humans to robots. Supervisor: Prof. Aude Billard

Jury member: Alberto Rodriguez (MIT), Kenji Tahara (Kyushu University),

Jose del R. Millan (EPFL), Jamie Paik (EPFL).

2008-2011 M.S. ME, Huazhong University of Science and Technology, China

Thesis: Grasping force distribution and uncertainty analysis, Supervisor: Prof. Wenyu Yang

2004-2008 B.S. ME, Huazhong University of Science and Technology (HUST), China

Thesis: A compliant end-effector for robot polishing, Supervisor: Prof. Wenyu Yang

Professional Experience

2013-2015 Teaching Assistant for [Robotic Practicals](#) (Master Course), EPFL

Topic: Teaching robots to accomplish a manipulation task

2011.9-2015 Research Assistant at LASA, EPFL, working on EU project RoboHow

Topic: Planning, control and learning on grasping and manipulation

- * Developed an optimization-based grasp synthesis method

- * Development of a learning-based adaptive control strategy for robust grasping

2008.9-20011.2 Researcher on National Basic Research Program of China (973 Program) –

Large-scale heavy component stable clamping principle and its driven strategy

Topic: Actuator force analysis for a heavy-duty manipulator in forging

- * Fixture layout design for industrial workpiece

- * Grasping force optimization

2008.6-2008.9 Research Intern, [HUAHENG WELDING](#), China

Topic: Stiffness identification and calibration for an industrial manipulator

- * Developed a joint stiffness identification method for FANUC painting robot.

Publications

Journal Articles

- 1 **M. Li**, K. Tahara, A. Billard, Learning task manifold for constrained object manipulation, *Robotics and Autonomous Systems*, 2016, in submission.
- 2 K. Hang **M. Li**, J. A. Stork, Y. Bekiroglu, F. T. Pokorny, A. Billard and D. Kragic. Hierarchical fingertip space: a unified framework for grasp planning and in-hand grasp adaptation. *IEEE Transactions on Robotics* (Conditionally accepted, 2015)
- 3 **M. Li**, K. Hang, D. Kragic, A. Billard. Dexterous grasping under shape uncertainty. *Robotics and Autonomous Systems*, 2015. [doi:10.1016/j.robot.2015.09.008](https://doi.org/10.1016/j.robot.2015.09.008)
- 4 B. Huang, **M. Li**, R. D. Souza, J. J. Bryson and A. Billard. A modular approach to learning manipulation strategies from human demonstration. *Autonomous Robots*, 2015. [doi:10.1007/s10514-015-9501-9](https://doi.org/10.1007/s10514-015-9501-9)
- 5 S. El-Khoury, **M. Li**, A. Billard. On the generation of a variety of grasps. *Robotics and Autonomous Systems*, 61(12), 1335-1349, 2013.
- 6 **M. Li**, W. Yang, X. Zhang. Robust actuator force Analysis of a heavy-duty manipulator using GMM/GMR. *International Journal of Robotics and Automation*, 28(4), 349-356, 2013.

Conference Proceedings

- 1 K. Hang, J.A. Haustein, **M. Li**, A. Billard, C. Smith and D. Kragic, On the evolution of fingertip grasping manifolds, ICRA 2016, accepted.
- 2 **M. Li**, Y. Bekiroglu, D. Kragic, A. Billard. Learning of grasp adaptation through experience and tactile sensing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, 2014.
- 3 **M. Li**, H. Yin, K. Tahara, A. Billard. Learning object-level impedance control for robust grasping and dexterous manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
- 4 N. Sommer, **M. Li**, A. Billard. Bimanual compliant tactile exploration for grasping unknown objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
- 5 B. Huang, S. El-Khoury, **M. Li**, J. J. Bryson, A. Billard. Learning a real time grasping strategy. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.
- 6 S. El Khoury, **M. Li**, A. Billard. Bridging the gap: one shot grasp synthesis approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, 2012.
- 7 **M. Li**, W. Yang, X. Zhang. Analysis of indeterminate force in multi-contact friction grasp based on potential energy minimization. In *International Conference on Modelling, Identification and Control (ICMIC)*, 2010.
- 8 W. Yang, **M. Li**, X. Zhang. Robust robotic grasping force optimization with uncertainty. In *International conference on Intelligent Robotics and Applications(ICIRA)*, 2010.
- 9 **M. Li**, W. Yang, X. Zhang. Projection on convex set and its application in testing force closure properties of robotic grasping. In *International conference on Intelligent Robotics and Applications(ICIRA)*, 2010.

- 10 X. Zhang, W. Yang, **M. Li**. An uncertainty approach for fixture layout optimization using Monte Carlo method. In *International conference on Intelligent Robotics and Applications(ICIRA)*, 2010.
- 11 X. Zhang, W. Yang, **M. Li**. Fixture layout and clamping force optimization for large-scale workpiece using augmented Lagrangian method. *Applied Mechanics and Materials* page 560–565, 2010.

Workshop

- 1 K. Hang, **M. Li**, J. A Stork, Y. Bekiroglu, A. Billard, D. Kragic. Hierarchical fingertip space for synthesizing adaptable fingertip grasps. *Autonomous Grasping and Manipulation: An Open Challenge*. ICRA, 2014.
- 2 **M. Li**, S. El Khoury, A. Billard. Synergy-level grasp synthesis learning. *Hand synergies - how to tame the complexity of grasping*. ICRA, 2013.

Invited Talks

- 1 Vision and Haptic-Guided Fine Manipulation of Objects, Bath University, UK, July, 2014.
- 2 Vision and Haptic-Guided Fine Manipulation of Objects, workshop of the British Machine Vision Association (BMVA) on Vision for Language and Manipulation, UK, July, 2014.
- 3 Robotic Grasping and Manipulation in Human-centered Environments, Huazhong University of Science and Technology, China, December, 2015.

Award and Honors

- 2010** Excellent Master Paper Award, HUST.
- 2008** Scholarship for Distinguished Graduate Student, HUST.
- 2008** Outstanding Graduation of HUST.
- 2005, 2006** Scholarship for Distinguished Students.

Professional Service

Student supervised (master semester project):

Zijin Yu, Object shape estimation with active touch sensing;
 Vermot Bertrand Claude Daniel, Learning tactile signals in object manipulation task

Reviewer

IEEE Transactions on Robotics, ICRA, IROS, Humanoids, IEEE Sensors Journal, International Journal of Advanced Robotic Systems

Skills

Robotics: robotic manipulation and control, robotic hands, tactile sensing, human robot interaction;
 Machine Learning;
 programming: C++, MATLAB, python

Language: Chinese(native), English(fluent)

Robots used: KUKA LWR, iiwa, iCub, Katana, Barrett hand, Allegro hand;

Other robotics related experience: ROS, OptiTrack, OpenRAVE, AMPL, ATI force sensing, BioTac;

Courses: machine learning, applied machine learning, optimal control, statistical sequence processing, model predictive control(attend);

Referees

Aude Billard, professor at EPFL, Switzerland, Email: aude.billard@epfl.ch
— Supervisor of my PhD thesis

Danica Kragic Jensfelt, professor at KTH, Sweden, Email: dani@kth.se
— Collaboration on several joint papers under EU project RoboHow

Kenji Tahara, professor at Kyushu University, Japan, Email: tahara@mech.kyushu-u.ac.jp
— Long-term collaboration on impedance control of robotic hand

Personal

Born on Dec 17, 1986, Hubei, China. Married with Hong Zhang and have a lovely son Noah.

Last updated: January 23, 2016
<http://lasa.epfl.ch/people/member.php?SCIPER=207200>

